# Commodore

AYLIK BİLGİSAYAR DERGİSİ Mart 1987

RADYOLAR VE BİLGİSAYAR

PASCAL'E BAŞLIYORUZ

SAYI 13

1500 TL (KDV dahil)

Program Dokum Dergisi ekiyle birlikte

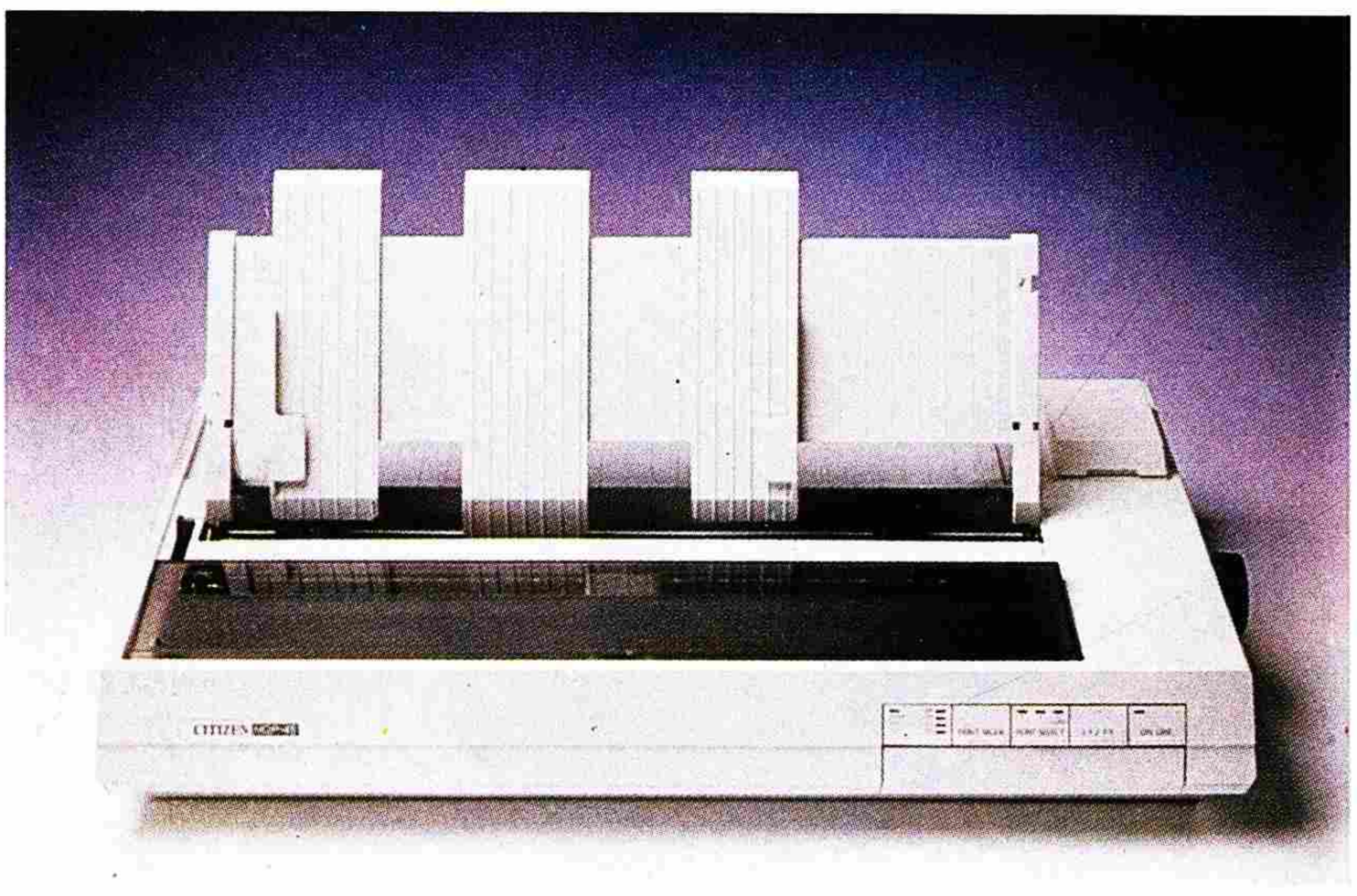
CITIZEN
BİLGİSAYAR YAZICILARI

# "İŞTE CITIZEN FARKI..."

- SAĞLAM YAPI, İKİ YIL GARANTİ
- MASANIZA YAKIŞAN ZERAFET
- 8 DEĞİŞİK MODEL 132-240 CPS SÜRAT
- UYUMLU (HEM IBM, HEM EPSON MUADİLİ)
- MEKTUP KALİTESİNDE YAZI
- EL YAZISI OPSİYONU
- GRAFİK ÇİZİM ÖZELLİĞİ
- KAĞIT HARCAMAYAN TRAKTÖR ÜNİTESİ
- KAĞIT BESLEME ÜNİTESİ OPSİYONU
- SESSİZ ÇALIŞMA
- TÜRKÇE HARFLER
- TÜRKÇE KULLANIM KILAVUZU
- GÜVENLİ TEKNİK SERVİS

VE

- EKONOMİK FİYAT



**SİZ DE BİLGİSAYARINIZA BİR CITIZEN BAĞLAYIN,**
**KALİTE İLE TANIŞIN...**

TÜRKİYE DİSTRİBÜTÖRÜ:

**aydın**
**PAZARLAMA A.Ş.**

Buyukdere Cad. Saadet Apt.
No.: 65/8 Mecidiyekoy - İstanbul
Tel.: 172 28 17 - 18 Telex: 27024 isip tr

**CITIZEN YAZICILARINI SEÇKİN**
**BİLGİSAYAR FİRMALARINDA**
**BULABİLİRSİNİZ.**

# İÇİNDEKİLER

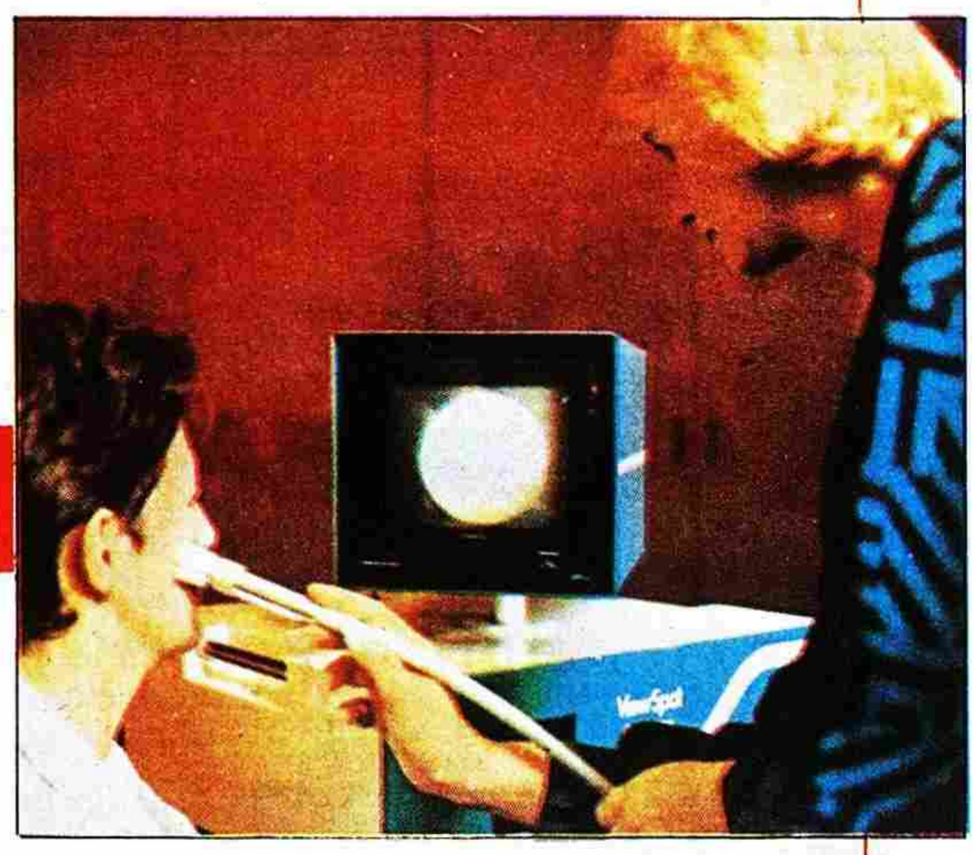

**5** Yayıncının Notu

**6 Haberler**


## ÖZEL BÖLÜM: Kadınlar ve Bilgisayar

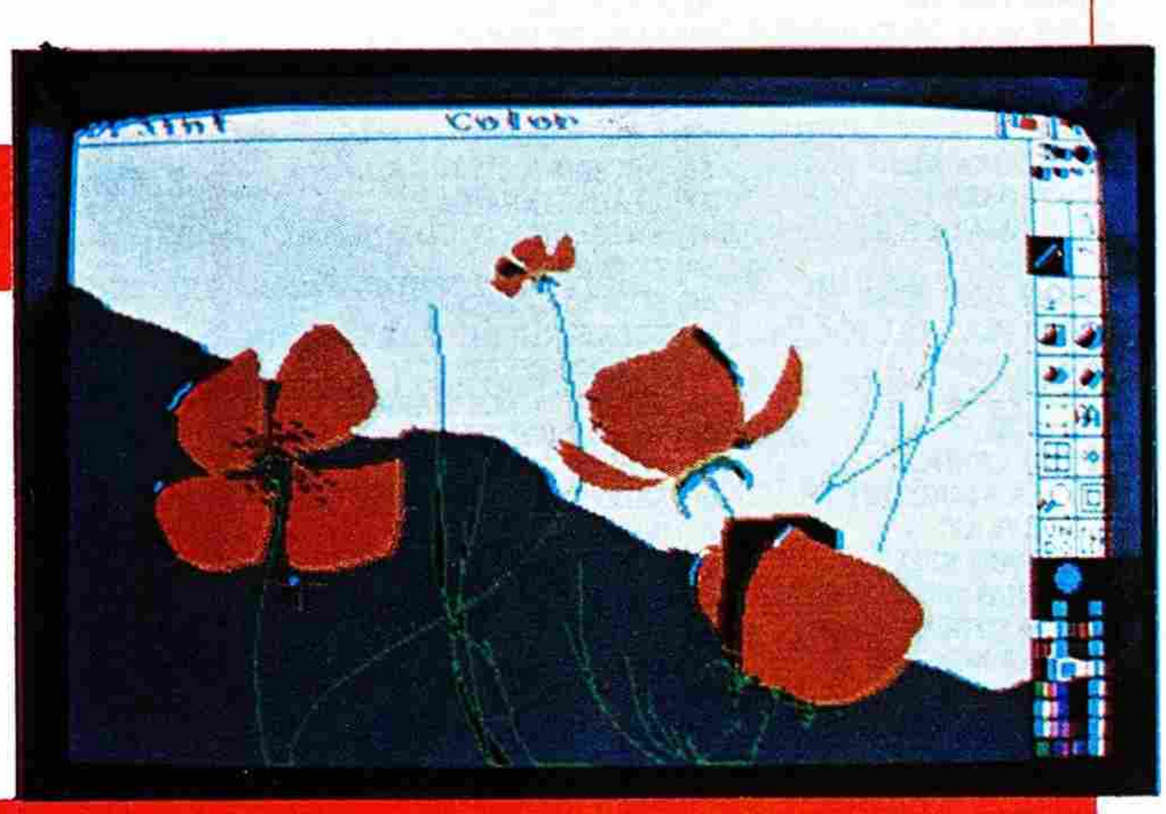

**10 Estetik Cerrahi ve Kozmetik**
**Estetik Müdahalenin Sonucunu**
**Bilgisayar Önceden Gösteriyor**

**17 Kadınlar Bilgisayarın Başına Geçmek İstiyorlar**

## DONANIM


**25 Stereo-SID**
**27 Harika CHIP'i Yaratmak**
**43 Yapısal Programlama**
**48 Modüler Programlamanın Anahtarı PASCAL (1)**
**51 C-64 ile Grafik (5)**

## YAZILIM


**55 Her İşin Başı Çekirdek (1)**
**57 Rastlantısal Sayıların Önemi**
Bilgisayar Laf Olsun Diye Zar Atmaz


## TANITIM

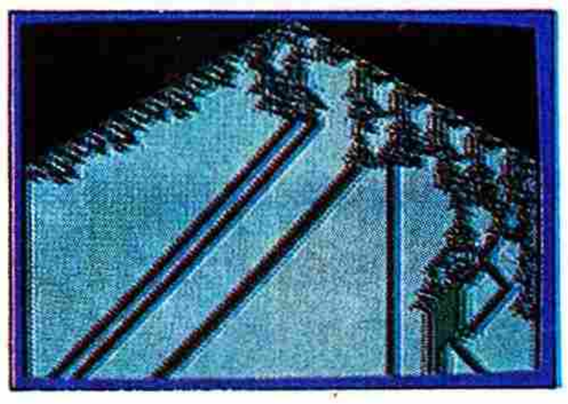

**30 Paperboy**
**32 Flight Simülatör II**
**34 Çizgiroman**
**36 Renkli Yaşam**
**40 Trap Door**
**42 Compu Hobi**

**45 Teleteknik**
**Matematik Dersinde**

**46 Küçük İlanlar**









SADECE

commodore

SATARIZ

Garanti Bankası
Bakırköy Şubesi
Hesap No: 1.609.576-2



# SORBİM

Bilgi İşlem Merkezi

Teleteknik

commodore

Yetkili Satıcısı

Istanbul Cad. No: 18 D.32
Bakırköy-ISTANBUL
Tel: 583 85 33-583 34 60

# Bir kerede 5000 TL'lık alışveriş yapan herkese bedava çekiliş kuponu

## MART AYI TEK OYUNLUK KASET

1) WORLD GAMES
2) TARZAN
3) SILENT SERVICE
4) TWO ON TWO
5) FIST 2
6) ASTERIX
7) INFILTRATOR
8) THE WAY OF THE TIGER
9) THAI BOXING
10) CAULDRON 2
11) LEADER BOARD 3
12) BAZOOKA BILL
13) DAN DARE
14) UCHI MATA
15) 1942
16) STAR PING PONG
17) WAR PLAY
18) PAPER BOY
19) PARALLAX
20) SINBAD
21) HIGH LANDER
22) IT'S A KNOCKOUT
23) ALLEYKAT
24) CAPTAIN KELLY
25) MISSION A.D.
26) STRIKE FORCE COBRA
27) YIE AR KUNG FU 2
28) ACE
29) HYPRABALL
30) INFODROID
31) THE SENTINEL
32) SUMMER GAMES 1
33) SUMMER GAMES 2
34) WINTER' GAMES
35) MOVIE MONSTER
36) SUPER CYCLE
37) JOHNNY REB 2
38) LAW OF THE WEST
39) TOUCHDOWN FOOTBALL
40) KNIGHT GAMES
41) BREAK THRU
42) VERA CRUZ
43) STALLONE COBRA
44) BOBBY BEARING
45) ACE OF ACES
46) REPTON 3
47) GLIDER RIDER
48) DROID
49) MISSION ELEWATOR
50) TRAP
51) S.F.POKER
52) SHORT CIRCUIT
53) W.A.R
54) GALIVAN
55) REBEL PLANET
56) ARCANA
57) STACE HARRIER
58) ANTIRIAD
59) TIME TRAX
60) SCOOBY DOO
61) EQUINOX
62) GREEN BERRET
63) STRIKE FORCE HARRIER
64) WORLD CUP CAN.
65) POWER PLAY
66) SABOTEUR
67) IRIDIS ALPHA
68) SKY RUNNER
69) HEAD COACH
70) KETTLE
71) 10TH. FRAME
72) TERRA KRESTA
73) DANTE'S INFERNO
74) FUTURE NIGHT
75) FOOTBALL OF THE YEAR
76) JUDGE DREDL
77) LEGEND OF KAGE
78) BOMB JACK II
79) AVENGER
80) THE GRAPHIC AD.GRE
81) HEARTLAND
82) IC.V.P.S.
83) COMMANDO LİBYA
84) DRACULA
85) JUMP JET
86) NINJA MASTER
87) MIKIE
88) GUNSHIP

## PROGRAM FİYATLARI

Kaset/Disket Tek Oyun.....600
Disket Tek Yüz.....1000
Disket Çift Yüz.....1500
Tek oyunlu kaset.....1200

## FLAŞ OLAY!..

### BİR YILLIK KURS

Aylık Fasiküller Halinde
1. Commodore Basic
2. Commodore Machine Language

VE

3. Oyun Açıklamaları (her ay 10 ad.)
4. Programlar (her ay 5 ad.)
5. Commodore Dünyasından Haberler

AYRICA
SORBİM'DE
BİR YIL SÜREYLE
%10 İNDİRİM

**YILLIK 16.000 TL.**

POSTA ÜCRETİ ALINMAZ

## ÇEKİLİŞ HEDİYELERİ

1 Ad. Quick Shot IX Joystick
2 Ad. Quick Shot V Joystick
2 Ad. Sylbert Joystick
2 Ad. Quick Shot VII Joystick
3 Ad. Elektronik Hesap Mak.
4 Ad. Bilgisayar Masa Saati
10 Ad. Bayan Kol Saati Takımı
20 Ad. Tek Oyunluk Kaset

Çekiliş 4.4.1987 Günü
Saat 14.00'te SORBİM'de

## DERGİCİLERİMİZ

Commodore.....1500
Commodore User.....2800
Your Commodore.....2800
Computer Gamer.....2500
Computer and Video Games.....3000
Your Computer.....4400
Compute .....5200
Byte.....8250

## JOYSTİCKLERİMİZ

Sys-Cont-1 .....8700
Quick Shot II.....12500
Quick Shot V (ithal).....17500
Quick Shot VII (ithal).....12500
Quick Shot IX (ithal).....22500

## DİĞER ÜRÜNLERİMİZ

PRECESION DİSKET.....1950
BASF 2D DİSKET.....1950
DISKEXPRESS DİSKET.....1950
NEMOREX DİSKET.....1950
DİSK KAFA TEMİZLEYİCİ.....2800
FREEZE FRAME.....15000
ÖZEL KUTULU BOŞ KASET.....700
ICE MACHINE.....32500

## İSTEK YAPACAKSINIZ

1. İsteklerinizi belirleyiniz
2. Fiyatlarını tespit ediniz
3. İndiriminizi düşünüz (aboneler için)
4. Tutarını banka hesabına yatırınız
5. Banka dekontunun fotokopisini ve isteklerinizi bize postalayın
6. İstek mektubunuzda ad, soyad, adresiniz, varsa telefon numaranızı belirtiniz.
7. Size aynı gün yurt içi kargo veya posta ile gönderelim (gönderme ücreti alınmaz).

**İsteyenlere ücretsiz yeni oyunların listesi gönderilir**

## Abonelere ve Bayilere % 10 indirim

COMMODORE
AYLIK BİLGİSAYAR DERGİSİ

**Teleteknik A.Ş. adına**
**Sahibi ve Sorumlu Yazışleri Müdürü**
Osman BABAOĞLU

**Yayın Yönetmeni**
İskender SAVAŞIR

**Yayın Koordinatörü**
Turhan TANDOĞMUŞ

**Danışma Kurulu**
Güray AKDOĞAN
Bilal Sadi OĞUZ
Yusuf BİTON
Bülent ÖZÜKAN
Emre SENAN

**Yayın Kurulu**
Kenan AKKERMAN
Seyfettin ÇANKAYA
Tuna ERTEMALP
Mustafa GÜVEN
Ayhan KALAYCIOĞLU
Samet KUMAŞ
İskender SAVAŞIR
Aziz ŞENLEN
Turhan TANDOĞMUŞ
Ayşegül TARIM
Tan ORAL

**Fotoğraflar**
Mehmet ÖZCAN

**Baskı Öncesi Hizmetler ve Yönetim**
BOYUT Reklam/Tanıtım Hizmetleri
Yayıncılık ve Ticaret A.Ş.

**Teknik Yönetmen**
Sevinç YENAL

**Grafik**
Saadettin OKTAY

**Dizgi**
Hüsnü ABBAS
Hamit BALABAN

**S/B filmler**
Mustafa ÖZSOY

**Reklam ve Halkla İlişkiler**
Aysın AYDEMİR

**Kapak Düzeni**
Emre SENAN

**Katkıda Bulunanlar**
Sema AKÇAY
Birgül ASENA
Silva AVEDİKOĞLU
Cafer BULUT
Türkan ÇEBİOĞLU
Öjeni GÜLLAPYAN
Derya KÖROĞLU
Mehmet SARIYAR
Ankin SİNANYAN
Esra SİPAHİOĞLU
Mahmure ÜNYELİ
Can KOZANOĞLU
Fethi SERİM
Bilge AYAN

**Reklam Rezervasyonları**
Rıhtım Cad. 207/4
Nesli Han Karaköy/İstanbul
Tel: 145 13 30

**Film ve Renk Ayrımı:** Laçin Repro
**Baskı:** Alaş Matbaası

**Kapak Renk Ayrımı:** Güven Reprodüksiyon

**Dağıtım:** GAMEDA

*KKTC Satış Fiyatı: 1850 TL.*

# YAYINCININ NOTU

## *Bilgisayar ve Kadınlar*

Bugüne kadar dergimizde daha çok Commodore'un ve genel olarak bilgisayarın sunduğu imkânların çeşitliliğinden söz ettik. Ama bu çeşitlilik önemli ölçüde bilgisayar kullanan insanların da çeşitliliğinden kaynaklanıyor. Günümüz piyasasında artık bilgisayarların ve bilgisayar dillerinin dostane (user friendly) olması tam anlamıyla vazgeçilmez bir gereklilik haline gelmiş durumda. Bu ise kullanıcıların kendilerine özgü bilgisayarcılık üslupları geliştirmelerinin önünde engin ufuklar açıyor.

Kişisel üslup dediğinizde bireysel farkların yanısıra çeşitli gruplar arasındaki farklardan da söz etmek mümkün. Bunlar arasında bugüne kadar üzerinde en çok durulmuş olan kadınlarla erkekler arasındaki fark. Bu yüzden biz 8 Mart Dünya Kadınlar Günü'nü fırsat bilip Kadınlar ve Bilgisayar hakkında bir dosya hazırlamaya karar verdik. Daha da doğrusu dosyanın kapağını araladık. Araladık diyoruz çünkü bu konunun hakettiği biçimde işlenmesi asıl kadın okurlarımız sayesinde olacak. Elimizde yabancı dergilerden bulduğumuz yazıların çoğu daha çok bilgisayarın iş dünyasına girmesinin kadınlar üzerindeki etkileri üzerinde duruyor. Biz de "Estetik Cerrahi" yazımızda örneklendiği gibi ağırlıklı olarak kadınlar için tasarlanmış uygulamalar var. Ama kadınların kendileri bilgisayardan nasıl yararlanıyor? Kadınlara özgü bilgisayar kullanma biçimleri var mı? Bu sorulara biz cevap veremeyiz. Ama önümüzdeki aylarda kadın okurlarımızın bilgisayar dünyasında sık sık sözü edilen eşitsizliği kıracaklarına inanıyoruz.

Bu arada 16-22 Şubat tarihleri arasında gerçekleştirdiğimiz Commodore Show da üzerimizde tatlı bir yorgunluk bırakarak sona erdi. Fuar boyunca süren ilgi ve desteğiniz, hem yorgunluğumuzun tatlı olmasına, hem de Show'un gelecek yıl tekrarlanması düşüncesinin doğmasına yolaçtı. Fuar konusunda daha detaylı bilgileri bu ve bundan sonraki sayılarımızda bulabileceksiniz.

Dergimizin Yayın Koordinatörlüğü'nü bu sayıdan itibaren daha önce Yayın Kurulu'nda bulunan Turhan Tandoğmuş yapacak. Tandoğmuş'a yeni görevinde başarılar diliyoruz. □

*Osman Babaoğlu*





## Dört Dörtlük Bir Şölen

# Commodore Show Büyük İlgi Gördü



Teleteknik tarafından düzenlenen ve Commodore Bilgisayarları ile donanım gereçlerini pazarlayan yirmi beş yetkili satıcının katıldığı Commodore Show bilgisayar meraklıları tarafından büyük ilgi gördü. 16 Şubat 1987 Pazartesi günü Genel Müdür Osman Babaoğlu tarafından açılışı yapılan Commodore Show, yalnızca bilgisayarların sergilendiği bir alan olmaktan öte, sürdüğü bir hafta boyunca bilgisayar dünyasının gündeminde olmayı başaran, çeşitli seminerler, gösteriler ve çekilişlerle zenginleştirilmiş tam bir "bilgisayar şenliği" idi.

Commodore'un ilk ürünlerinden olan C-64'ten başlayıp, üstün ses ve görüntü özelliklerine sahip Amiga'ya uzanan tüm ürünlerin yer aldığı fuarın yıldızı, Amiga idi. Teleteknik standında çeşitli gösterilerle tanıtılan Amiga, izleyiciler tarafından büyük ilgi gördü. Fuar boyunca Amiga'nın becerilerinin sergilendiği Teleteknik standı, fuarın en kalabalık bölümü idi.

Bu arada fuarda yer alan yeni ürünler de Commodore'cular tarafından ilgi ile izlendi. Bu yeni ürünlerden biri de Light Pen (Işıklı Kalem) idi. Işıklı Kalem için ayrılmış bilgisayarların başında toplanan bilgisayar meraklıları, cazibesine kapıldıkları Işıklı Kalem'i bilgisayar gereçlerinin arasına katacakları izlenimini bıraktılar.

Fuara katılan firmalar tarafından gerçekleştirilen ve çeşitli program tanıtımları ile bilgisayar kullanım olanaklarını konu alan seminerler farklı ilgi alanlarına





seslenmekle birlikte, bilgisayar kullanımı noktasındaki ortaklıkları nedeniyle yine kalabalık bir seyirci kitlesi buldular.

Commodore Dergisi'nin de yer aldığı yayın köşesi de fuarın renkli köşelerinden biri idi. Bilgisayar konusunda çeşitli yayınlara ve diğer bilgisayar dergilerine yer verilen köşede en fazla ilgiyi Commodore Dergisi gördü. Fuarı ziyaret eden pek çok kişinin dergiye abone oldukları izlendi.

Bu arada fuar süresince, yapılan çekilişlerle hergün 15 kişiye çeşitli hediyeler verildi.

Son derece hareketli geçen Commodore Show 22 Şubat Pazar günü saat 19.00'da fuara katılan herkesin tekrarlanması dilekleri ile son buldu. □

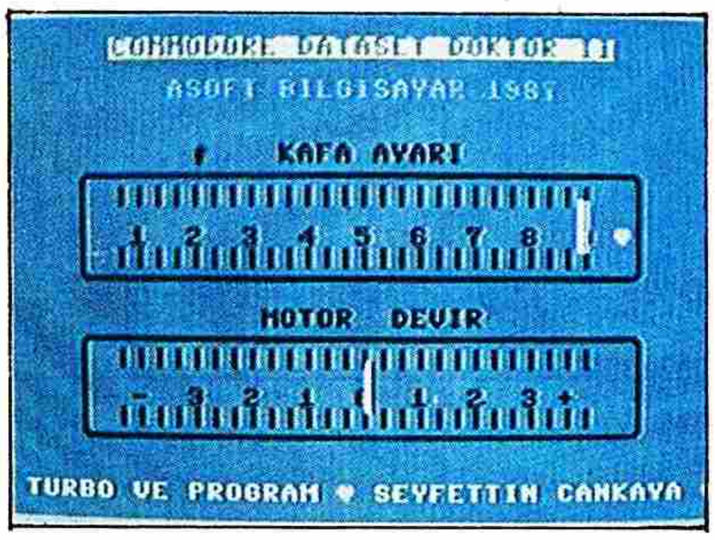
# Dataset Doktor 2 Çıktı

İlki 1985 Yılında TES bilgisayar tarafından hazırlanan Dataset Doktor program gelişmiş haliyle 1987 başında ASOFT bilgisayar tarafından yeniden üretildi.

İlkinden farklı olarak motor devir ayarını da yapabilen program böylece iyi bir kafa ayarına karşılık motor devri yüzünden çıkan sorunları da önlemekte ayar olanağı sağlamaktadır.

Bilindiği gibi DD1 programında




kafa ayarı bir ibre yardımı ile en iyi konum olan 9 ölçeğine getirilmekteydi. DD2'de ise bu ölçeğin hemen altında motor devir ölçeği de yer almakta kafa ayarı ve motor devri aynı anda izlenebilmektedir. Motor devri düşük aldığında ibre (-) yine hızlı alındığında ise (+) yine kaymaktadır. En iyi konum olan "0" noktasının motor ayar vidası ayarlanarak ibreyle çakışması sağlanmalıdır.

DD2 kullanıcılara ASOFT bilgisayar tarafından üretilecektir. Kopyaları tam bir ayarlama yapmayacağı için kasetlerin ASOFT etiketli olması önemlidir.

Daha ayrıntılı bilgiyi 167 57 43 no.lu telefondan elde edebiliriz.□

Sorbim "Sömestr Kampanya"sına katılan öğrenci ve ailelerine teşekkür eder.

24 Ocak'ta başlattığımız kampanya, büyük ilgi gördü. Katılan 140 kişiden 70'i çeşitli hediyeler kazandı.



# SATRANÇ / Begüm ERCE



## 5. Dünya Satranç Bilgisayarları Şampiyonası

11-16 Haziran 1986 tarihleri arasında Hegener-Glaser satranç bilgisayar kuruluşu ile Köln "Messe" tarafından düzenlenen ve Batı Almanya Satranç Federasyonu'nun ev sahipliğini yaptığı 5. Dünya Satranç Bilgisayarları Şampiyonası Köln'de yapıldı. Şampiyonanın en önemli yanlarından birisi dünyada ilk kez ulusal satranç federasyonunun ev sahipliğini üstlendiği bir şampiyona olmasıydı. Ayrıca turnuvaya 23 bilgisayarın katılması da bir rekordu.

1974 yılından bu yana yapılan turnuvaya, küçük ve büyük her türden bilgisayar katılabiliyor. Başlangıcından itibaren turnuvalara kısaca bir göz atalım: Stockholm'deki 1. Dünya Şampiyonası'nda, Moskova Kontrol Bilimi Enstitüsü tarafından yapılan KAISSA programı turnuvanın birincisi oldu. 1977'ye kadar (1977 turnuvası Toronto'da yapıldı) David Slate ve Larry Atkin tarafından yazılan Northwestern Üniversitesi programı Chess 4.6 dünyanın en iyisi olarak ün yaptı. Ancak Avusturya'nın Linz kentinde 1980'de Ken Thompson'un Belle'i salt satranca dayanarak yapılan bilgisayarlarla neler yapılabileceğini gösterdi. New York'da yapılan 1983 Dünya Şampiyonası'ndan önce Belle ilk ulusal usta ünvanını alan bilgisayar olduğu için Fredkin kuruluşundan 5.000 Dolar ödül aldı. İnsanlar arasında yapılan yarışmalarda elde ettiği 2203 rating; ona otomatik olarak ABD'nin Ulusal Ustası ünvanını kazandırdı. Çoğu otorite Belle'in 1983 Dünya Şampiyonu ünvanını kazanmasını bekliyordu. Ancak Cray Blitz adlı program, dünyanın en güçlü ticari programı üzerinde oynayarak 4. Dünya Şampiyonu oldu.

1983'te şampiyon olan Cray Blitz, 1985 yazına kadar üstünlüğünü korudu. Ancak o sırada Pittsburgh'da Carnegie-Mellon Üniversitesi'nde yazılan Hetich adlı bir program ortaya çıktı. Programlayıcısı bilgisayar bilimi profesörü ve Correspondence Satrancı'nda eski dünya şampiyonu olan Hans Berliner'di.

Hitech'in yapılış tarzı da, çok ilginçtir, 64 kare için ayrı ayrı 64 mikroişlemci kullanıyor, saniyede 200 bin analiz yapabiliyor, birincisi hesap yapmak, diğeri karar vermek gibi iki ayrı sistemden oluşuyordu.

Hitech ilk büyük başarısını insanlarla yaptığı maçlarda kazandı. 1985 sonbaharına kadar bir bilgisayar tarafından ulaşılabilmiş en yüksek ratinge sahip oldu. Pittsburgh'daki ustalar turnuvasında ratingi 2530'a ulaştı. 1985 Kuzey Amerika Şampiyonası Hitech tarafından inandırıcı bir şekilde kazanıldı. Bütün bunlardan anlaşılacağı gibi genel olarak Cray Blitz'in, ünvanını Hitech'e kaptırması bekleniyordu. Önceleri bu tahmin doğru çıkacakmış gibi görünse de, ikinci turda Cray Blitz Alman programı Bobby'ye karşı kaybetti. oysa bu programın, üç favori programdan birisine (Hitecn, Cray Blitz ve Be-Be) zorluk çıkaracağı sanılmıyordu. Onüçüncü hamlede programlar aşağı yukarı oyun sonuna ulaştılar. Vezirler değiştirildi. İki tarafın da 2 kalesi, 2 fili ve 5 piyonu bulunuyordu. Pozisyon tamamen simetrikti. Hemen hemen her oyuncu (insan), burada beraberliği kabul edip el sıkışırdı. Ancak Bobby yöntemsel olarak vezir kanadı piyonlarını sürdü. Sonra da o kanatta belirleyici bir üstünlük kazandı.

Üçüncü turdan sonra 23 programdan yalnızca ikisinin skoru % 100'dü: Hitech ve Rebel, Hollanda programı Rebel, turnuvanın ilk büyük sürpriziydi.

Özel olarak düzenlenmiş bir yapı üzerinde 6502 mikroişlemcisini kullanarak oynuyordu. Başlangıçta etkili olmasına rağmen dördüncü turda Hitech'e karşı başı derde girdi.

Turnuvanın beşinci ve son turu başladığı sırada Hitech bütün yarışmacılardan 1 puan ilerideydi. Ve herkes yeni dünya şampiyonunun taç giyişini seyretmeyi bekliyordu. Ancak saatler ilerledikçe Cray Blitz'in vezir kanadı piyonları, gitgide daha tehlikeli görünmeye başladı. Sonuçta Hitech rakibinin duble piyonları olmasına rağmen önemli bir geçer piyonunun bulunduğu oyun sonuna girmeye zorlandı. Cray Blitz oyunun sonraki kısmını önemli hamleleri en iyi şekilde oynayarak devam ettirdi.. 6 saatlik oyundan sonra Berliner, programı adına yarışmadan çekildi.

1986-1989 Dünya Şampiyonluğu'nu "rakiplerin puan toplama yöntemi" belirledi.



### Hitech'in Turnuvadaki En Mükemmel Oyunu

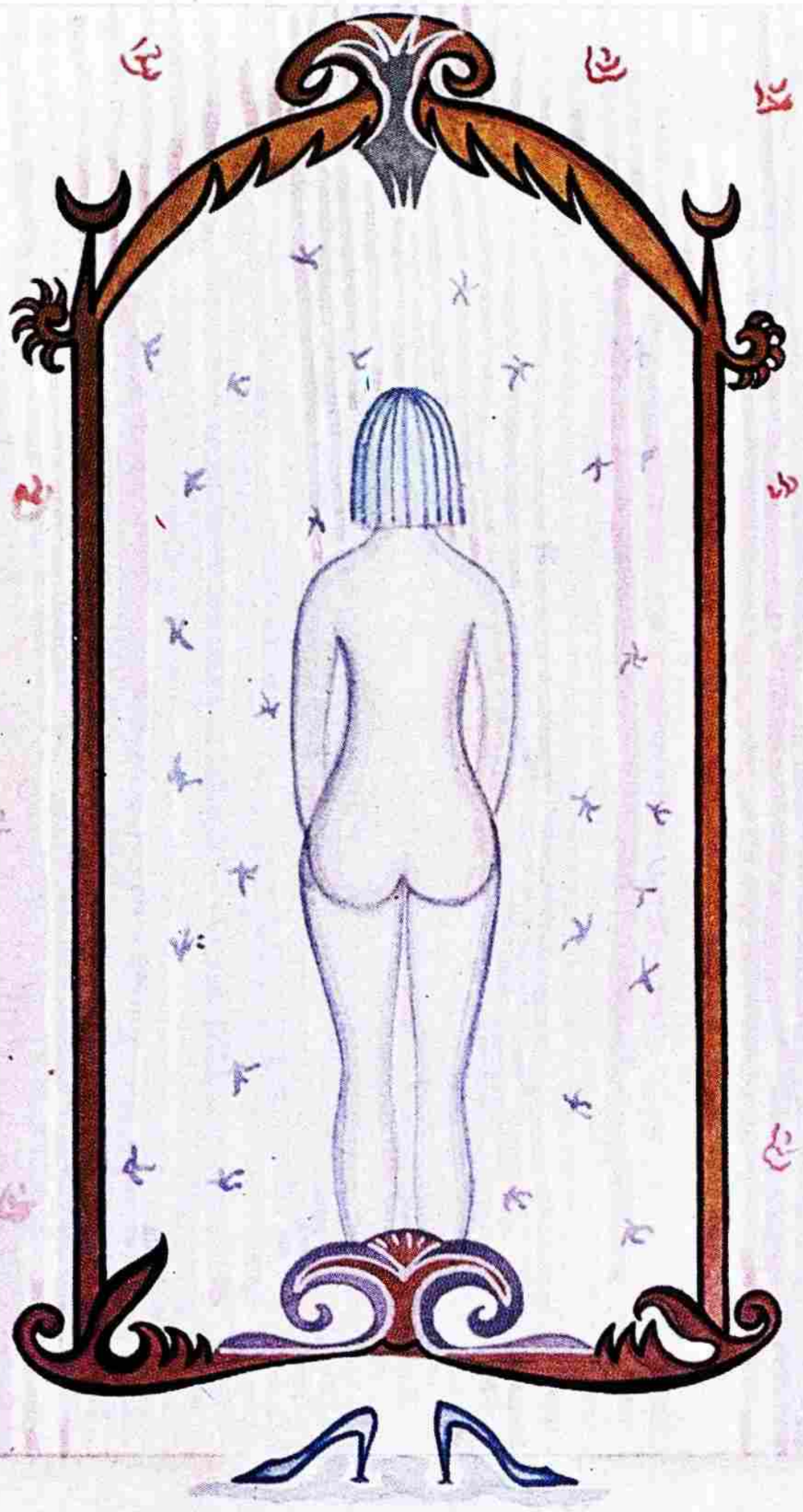
Hitech (ABD) - Schach (F.Almanya)

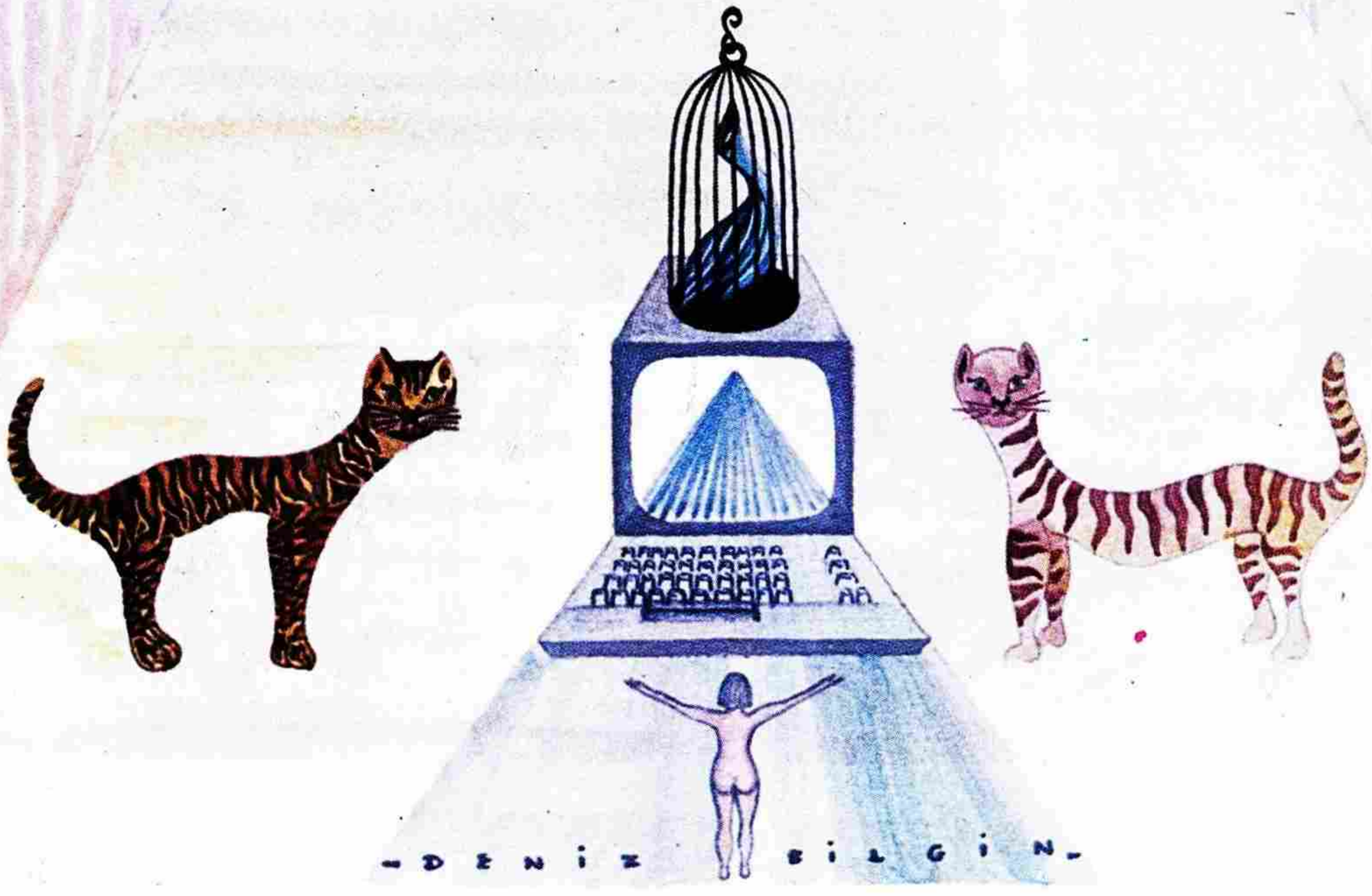
Sicilya savunması
1. e4 c5 2. Af3 d6 3. Fc4 e6 4. d4 cxd4 5. Axd4 Af6 6. Ac3 Fe7 7. Fe3 Abd7 8. Vd2 Ae5 9.Fe2 0-0 10. h3 Fd7 11. Af3 Axf3 12. gxf3 Va5 13. 0-0-0 Kac8 14. Khg1 Kfe8 15. Fh6 g6 16. Fg5 Vc5 17. Vf4 Ah5 18. Vh4 f6 19. Fe3 Va5 20. Fb5 Fxb5 21. Vxh5 g5 22. Fxg5 fxg5 23. Kxg Şh8 24. Kdg1 siyah terk eder.

### Turnuvaya Katılan Bilgisayarların Puanlarına Göre Ayarlanması

1. Cray Blitz (ABD) 2. Hitech (ABD) 3. Be-Be (ABD) 4. Sun Phoenix (Kanada) 5. -10. Rebel (Hollanda), Bobby (F.Almanya), Plymate (İsveç), Mephisto Cologne (İngiltere), Dutch and Nona (Hollanda), 10-13. Advance (İngiltere), Lachex (ABD), Ostrich (Kanada), 14-17. Schach (F.Almanya), Chrus 68 K (ingiltere), Vaxchess (İngiltere), Chat (F.Almanya), 18-20. BCP (İngiltere), Advanced Starchess (Danimarka), Awit (Kanada), 21-22. Rex (ABD), Shess (Hollanda), 23. Kempelen Atari (Macaristan).

# KADINLAR
ve
# BİLGİSAYAR





-DENİZ BİLGİN-




*PC, video-kamera ve grafik çizelgesinin yardımıyla, bir güzellik ameliyatı karşısında duyulan korku kaybolmaktadır. Dr. Barry Weintraub bir hastasına, bilgisayar monitöründe burnuna nasıl başka bir görünüm kazandırabileceğini gösteriyor (alttaki resim). Sağdaki resimde düzeltimden önceki video resmi, yanında ise (ekranda) düzeltilmiş burun gözüküyor.*



*Manon BAUKHAGE*

# *Estetik Cerrahi ve Kozmetik*




*Şimdiye kadar "bilgisayar-Simulation"unun baştan çıkarıcılığı karşısında yenik düşenlerin çoğu erkekler olmuştu. Şimdiyse kadınlar bilgisayarı kendilerine kusursuz güzelliğin yolunu açacak sihirli bir değnek olarak keşfetmekteler. Aşağıdaki yazıda plastik cerrahinin ve kozmetik salonlarının bilgisayarlardan nasıl yararlandığını okuyacaksınız.*

# *Estetik Müdahalenin Sonucunu Bilgisayar Önceden Gösteriyor*

Yeni bir buruna kavuşmak ya da yüzünün derisini gerdirmek gibi düşüncelerle bir kez flört etmiş olan herkes sihirli bir küreye de sahip olmak istemiştir mutlaka. Bu küreye bakıp "Ameliyattan sonra görünüşüm nasıl olacak?" sorusuna şimdiden bir yanıt verilebilseydi, bu konuda karar almak da kolay olacaktı.

Bilgisayar işte bu rüyayı kısa bir süre sonra gerçekleştirmiş olacak. Hastalar cerrahın bıçağına kendilerini teslim etmeden önce, yeni yüz hatlarının alacağı şekli gerçekten görebileceklerdir. Amerika'da sansasyonel bir tarzda kutlanan bu yöntemi ilk kez ünlü plastik cerrahi uzmanı Dr. Barry Weintraub uyguladı (Los Angeles'da). İşin şaşırtıcı yanı şu idi: Sisteminin temel taşını tamamen normal bir IBM XT oluşturuyordu. Buna ek olarak özel grafik yazılımı ve bir video-kamera gerekmekteydi.

Hastanın gelecekteki kendi yüzünü görmesi ise şöyle mümkün oluyordu: Monitör üzerinde hastalar kendi yüz hatlarının nasıl değişikliğe uğradığını izleyebiliyorlardı ve dahası yüzlerinin düzeltilmesine bizzat yardımcı olabiliyorlardı. Güzellik bilgisayarına el atma cesaretini gösteren ilk kadınlardan biri 23 yaşındaki Amerikalı Holly Floren idi. Floren'in yüzü video-kameraya alındıktan birkaç saniye sonra bilgisayarın monitöründe belirmişti. O, uygun bir program ve bir ışık kalemi aracılığıyla grafik çizelgesi üzerindeki video resmi değiştirmeye başlayabilirdi şimdi. Okul döneminden bu yana kendisinde hoşnutsuzluk uyandırmış olan burnunun şekli kendi gözleri önünde değişikliğe uğruyordu yavaş yavaş. Burnu daha yumuşak ve düzgün hatlara sahip oluyordu. Bilgisayar "öncesi/sonrası" resimlerini yanyana gösterdiği zaman Floren adeta büyülenmişti. Yaşamı boyunca ilk kez olarak ameliyattan artık hiç korkmuyordu. Neler hissettiğini şöyle dile getirmişti: "Birden kendimi bütünüyle güvence içinde duydum; çünkü müdahaleden sonra görünümümün nasıl olacağını tam olarak biliyordum. Tabii bunun yapılabilmesi Dr. Weintraub'un cerrah ellerinin, bilgisayarın önerisini tamı tamına gerçekleştirebilecek kertede mahir olmasına bağlıydı.

"Star" gazetesinin tıp sayfası redaktörünün bu deneyi konu alan yazısına göre, Dr. Weintraub estetik cerrahide gayrı ciddi ve bilmecemsi tahminler yapıldığına dair çıkarılan söylentiye son vermeyi başarmıştır. Aynı yazı, gelecekte geçirecekleri güzellik ameliyatları sonuçlarından memnun olmayan hasta sayısının herhalde azalacağından söz etmektedir.

Anılan doktor bu arada yüzlerce hastasını ameliyata hazırlarken bilgisayardan yararlanmıştır. Doktor bu ameliyatların sonucuna ilişkin olarak şunu söylemektedir: "Ameliyat olanların hepsi daha sonra yaptığım işten memnun kaldıklarını bildirmişlerdir." Doktorun yaptığı işin yalnızca burunları düzeltmekle sınırlı olmadığını da belirtmek gerekiyor. 58 yaşındaki bir bayan hasta şunları söylüyor: "Ameliyattan müthiş bir korku duyuyor, ama yüzümün derisini de mutlaka gerdirmek istiyordum. Ne var ki, ameliyat sonrasında yabancı bir yüzün aynadan bana bakmasını da istemiyordum." Dr. Weintraub ise onun ekrana yansıyan yüzündeki birçok derin kırışıklığı birkaç saniye içinde yok ediverdi. Durumu gören hasta ameliyatı kabul etmişti ve şöyle diyordu: "Gerçi on yıl daha gençleşmiştim, ama gene de hâlâ kendim olarak kalmıştım."

*Bilgisayarda depolanmış çeşitli "ideal" cilt tiplerinin karşılaştırılması yoluyla, müşteriye uygun düşen bakım programı hazırlanıyor ve müşteriye takdim ediliyor.*

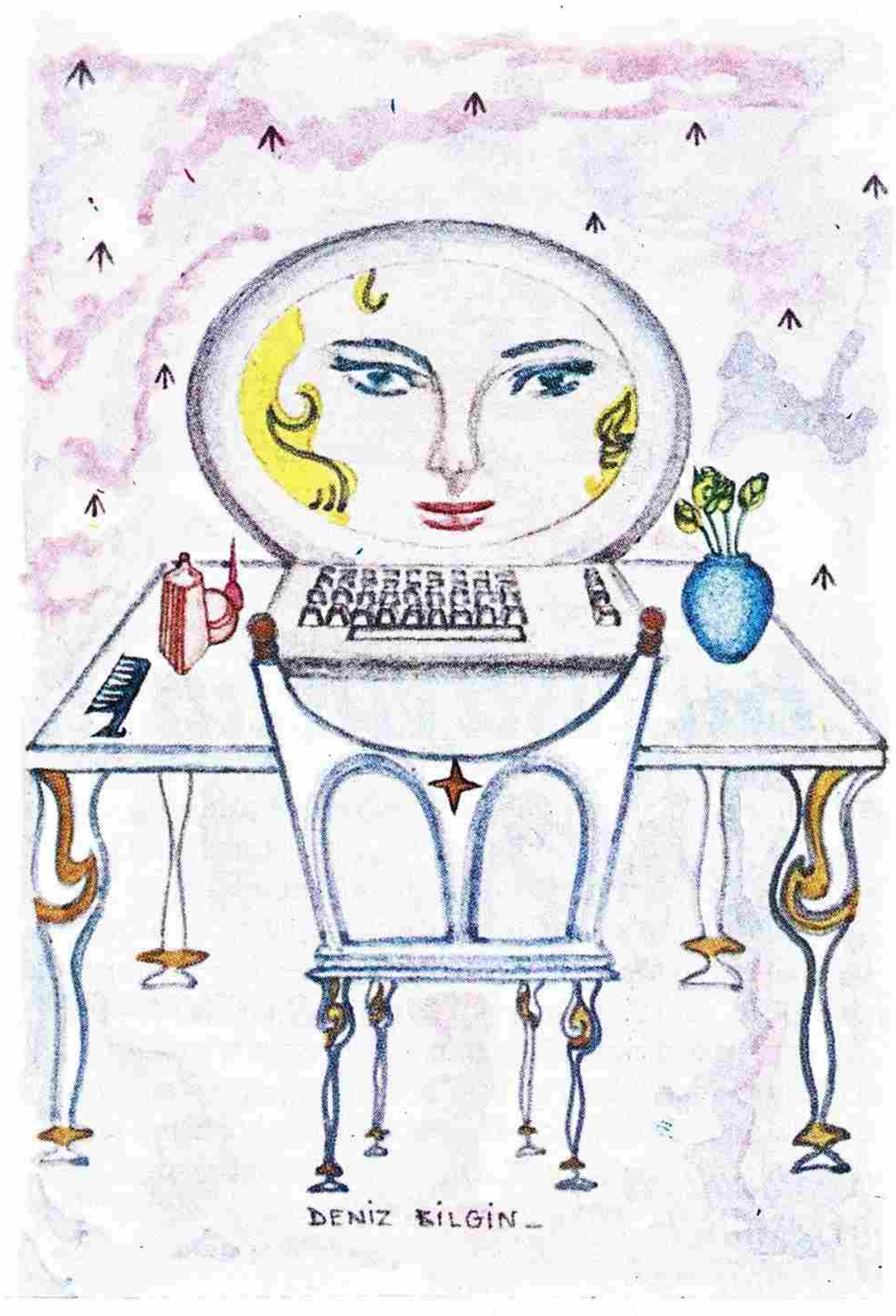


## *Kayıp Çocuklardan Güzel Kadınlara*

Dr. Barry Weintraub kendi güzellik bilgisayarını New York'taki Cornell Tıp Merkezi'nde Federal Polis (FBI) ile birlikte kaybolan çocukların araştırılması programında çalıştığı sıralarda geliştirdi. Kaybolan çocukları araştıranlar doktordan çocuk yüzlerinin zamanla nasıl bir değişiklik gösterdiklerine ilişkin bilgi istiyorlardı. Bu yola başvurmalarının nedeni ise şuydu: Kaybolan çocukların ana-babalarının ellerindeki resimler çoğunlukla yıllar öncesine aitti ve çocukların bulunmasında artık işe yaramıyordu. Keşke bilgisayar beş yaşındaki bir çocuk yüzünü on yaşındaki durumunda gösterebilseydi...

Bilgisayar gerçekten de başardı bu işi. O zaman Dr. Weintraub'un aklında şöyle bir düşünce doğdu: Aynı yöntem estetik cerrahi alanında da herhalde uygulanabilmeliydi - tabiî tersine bir yol izleyerek-.

Weintraub'un keskin görüşü sayesinde (ameliyat edileceklere ait) fotoğraflar üstünde rötuş yapma ve kurşun kalemle yetersiz taslaklar çizme işleri artık gereksizleşiyordu. Bilgisayar daha hızlı, çok daha kesin çalışabildiği gibi, hastaların da kendi değişiklikleri konusunda bizzat etkin olabilmelerinden kaynaklanan önemli bir psikolojik etkiye sahipti. Elbette bu elektronik imkânlar kişileri dürüst olmayan davranışlara da itebilir ve yeni bir tehlikeyi beraberle-



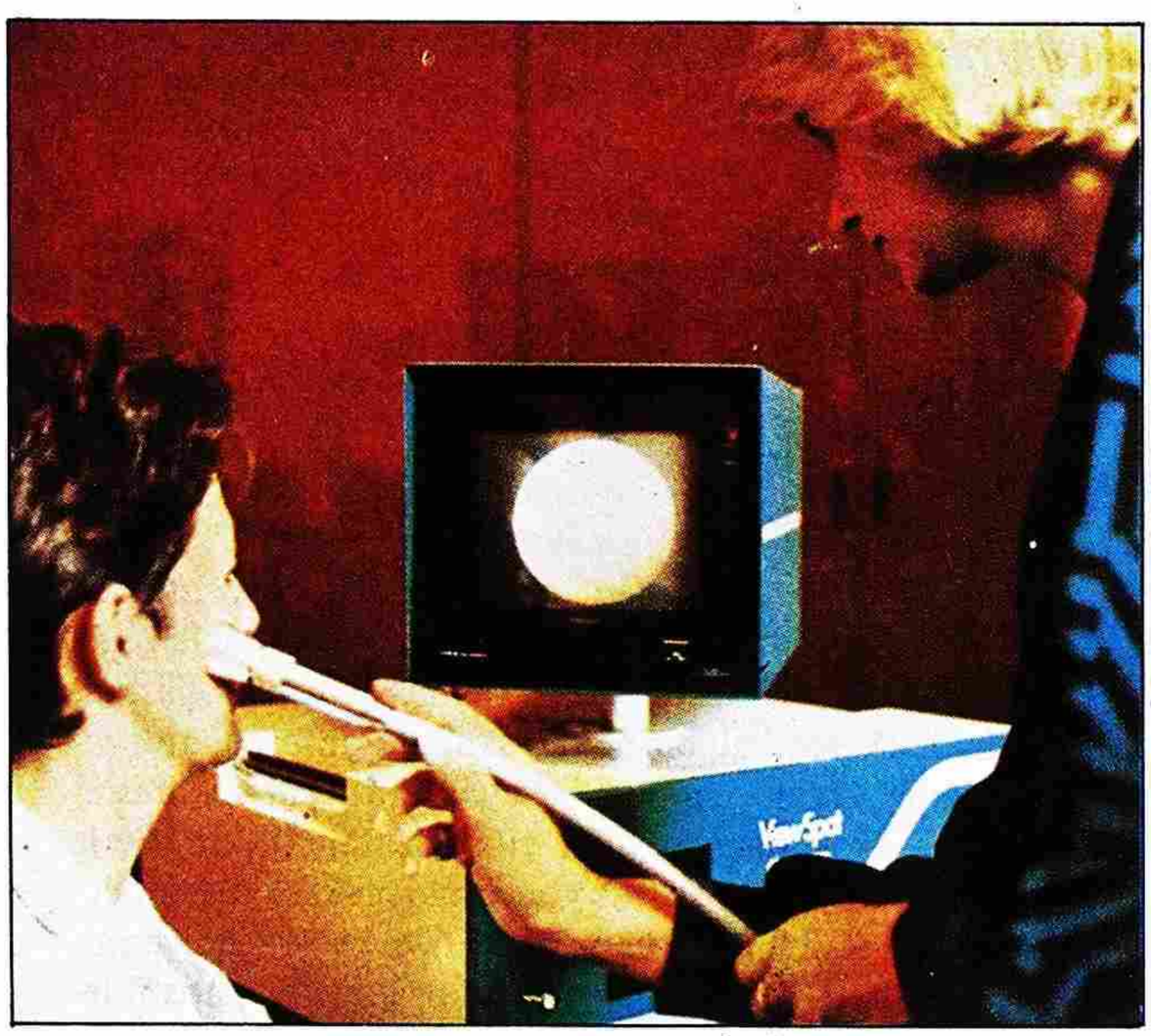

rında taşırlar: Doktor ve hastası ekran üzerinde öylesine kendi gururlarını okşayıcı resimler yaratabilirler ki, cerrah bunları ameliyat salonunda gerçekleştirmeye kalkıştığında gücünü aşan bir durumla yüzyüze gelebilir. Sonuçta doğa da kendi hakkını talep edecektir: Cerrahın yeteneklerinin yanısıra kemiklerin ve liflerin yapısı da her zaman önemli bir rol oynar. Bir şeyi ortadan kaldırmak ise bir şey eklemekten daima daha kolaydır. Dr. Weinbtraub şöyle diyor: "O, bilgisayarının başlıca görevinin hastalarının korkularını gidermek olduğunu düşünüyor."

Ama güzellik konusunda bilgisayarın tek kullanıldığı alan Weintraub'un ameliyata ilişkin hazırlık çalışmaları değildir. Japonya'nın en büyük kozmetik merkezi "Shissido"-'nun güzellik salonlarında, bilgisayar yüz bakımını kolaylaştırmak ve bunun daha iyi yapılmasını sağlamak amacıyla kullanılmaktadır. Burada ilkin yüzün çeşitli saç tuvaletleriyle alacağı biçim ekrana yansıtılmakta ve bu da kadınlar tarafından hayranlıkla karşılanmaktadır:

Müşterinin video resmi monitöre geçirildikten sonra yaklaşık 100 değişik saç biçimiyle donanmakta ve bunların içinden müşterinin yüzüne en uygun düşeni seçilebilmektedir. Bunun hemen arkasından makyaj denemesi gelmektedir: Müşterinin sayısal portresi özel bir yazılım aracılığıyla ve ideal çözüm bulununcaya kadar ekran üzerinde farklı şekillerde makyajlanmaktadır. Işık kalemleriyle çalışan kozmetik elemanı uygun renk tonlarını bulup çıkarmaktadır. Kumanda düğmesine basmak suretiyle renklerin yeğinlikleri değişebilmektedir. En sonunda uygun aksesuarlar da kullanılarak, işlem göz kamaştırıcı bir biçimde tamamlanabilir.

PC aracılığıyla cilt bakımı, "simulation-kozmetiği"nin en son buluşudur. Optik sensorlar yardımıyla gerek kırışıklıkların derinlikleri gerekse cildin nüansları, nemlilik ve yağlılık oranı açığa çıkarılabilmektedir. Depolanmış 36 adet cilt çeşidinin karşılaştırılması yoluyla (ve ekranda) kişisel makyaj programı hazırlanabilmektedir. Müşteri bu programı plastik bir kartın mıknatıslı şeritlerinde depolanmış olarak evine götürebilmektedir. Bir sonraki gelişinde de bakımın etkisini tam olarak denetleyebilmektedir.

Bütün o muazzam hesaplama başarıları bir yana, yeni bir "simulation" kültürünün doğuşu bilgisayarın yol açmış olduğu belki de en önemli olgudur. Bu kültür insanların gelecekte kendi kendileri ve gerçeklik üstüne ne düşüneceklerinin ölçütlerini sunmaktadır. Güzelliğin tam bir "simulation"u her zaman için güzelliğin kendisinden farklı bir şey olsa gerek... □



*Japon kozmetik merkezi "Shiseido" bilgisayarla cilt bakımı hizmeti sunuyor. Optik "sensor"lar aracılığıyla cildin yapısı açığa çıkarılıyor.*

*Çev. Güven Savaş KIZILTAN*




*Kariyerleri*
*Daktilo Odalarında*
*Son Bulmasın Diye*

# Kadınlar Bilgisayarların Başına Geçmek İstiyorlar

**Manon BAUKHAGE**

*Gerçi elektronik ve işleme salt erkekler dünyasına özgü bir iş olmadı hiçbir zaman, ama görevlerin dağılımı her zaman eşitsizdi ve kadınlar özellikle "zeki olmayan" terminalleri kullanmak zorunda kaldılar. Şimdi bu durumun değişmesi isteniyor...*

"Bilgisayarlar erkekler içindir. Kadınların bu işle hiçbir ilgileri yoktur." Böyle konuşan sadece erkekler değil, pek çok kadın da aynı şeyleri söylüyor. Kadınlar acaba bilgisayar dünyasını kendilerine açıp bu alanda uğraş versinler mi, yoksa "erkeksi bir teknik"ten çıkan bu sevimsiz ürüne hiç ilişmesinler mi?

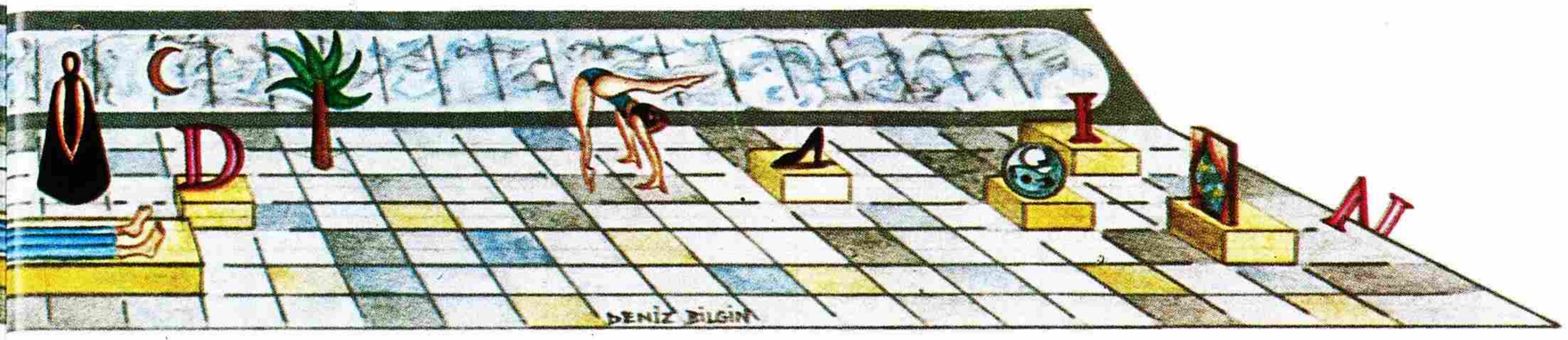
Özellikle angaje kadınların görüşlerini şu slogan iyi temsil etmektedir: "Biz Yokuz!-Kadınlar Bilgisayar Egemenliğine Karşı-" Ama bu işten kaçınmanın herhangi bir şeye yararı yoktur. Yeni teknikler bütün toplumsal kurumları değiştirirler. Bu tekniklerin getirdiği imkânlar ve taşıdığı tehlikeler kadınları, çocukları ve erkekleri hep birlikte ilgilendirir.

Mikro-bilgisayar üreticileri bugüne değin bu çetin konuya dokunmadan geçtiler. Sonunda büyük firmalardan biri bu işe el attı ve şu savı ileri sürdü: "Bilgisayar kadınların meslek yaşamlarına ilişkin durumlarını etkin bir tarzda düzeltebilir." Bunun ön koşulu olarak da kadınların bu işe gönül vermeleri ve tamamlayıcı mesleki eğitim konusunda istekli davranmaları gerektiği söyleniyordu. Bu savlar Commodore firmasının Hannover'de düzenlediği "Kadınlar ve Bilgisayar-Üçüncü Görüşte Başlayan Aşk mı?" temalı forumun odak noktasını oluşturuyordu.

Bütün katılanlar, kadınların bu işe el atmak, hem de çabucak el atmak zorunda olduklarında birlik olmuşlardır; çünkü meslek sahibi kadınların çoğunluğunun birkaç yıl gibi kısa bir süre içinde işyerinde çalışırken ekran ve bilgisayar kullanacağını düşünmektedirler.

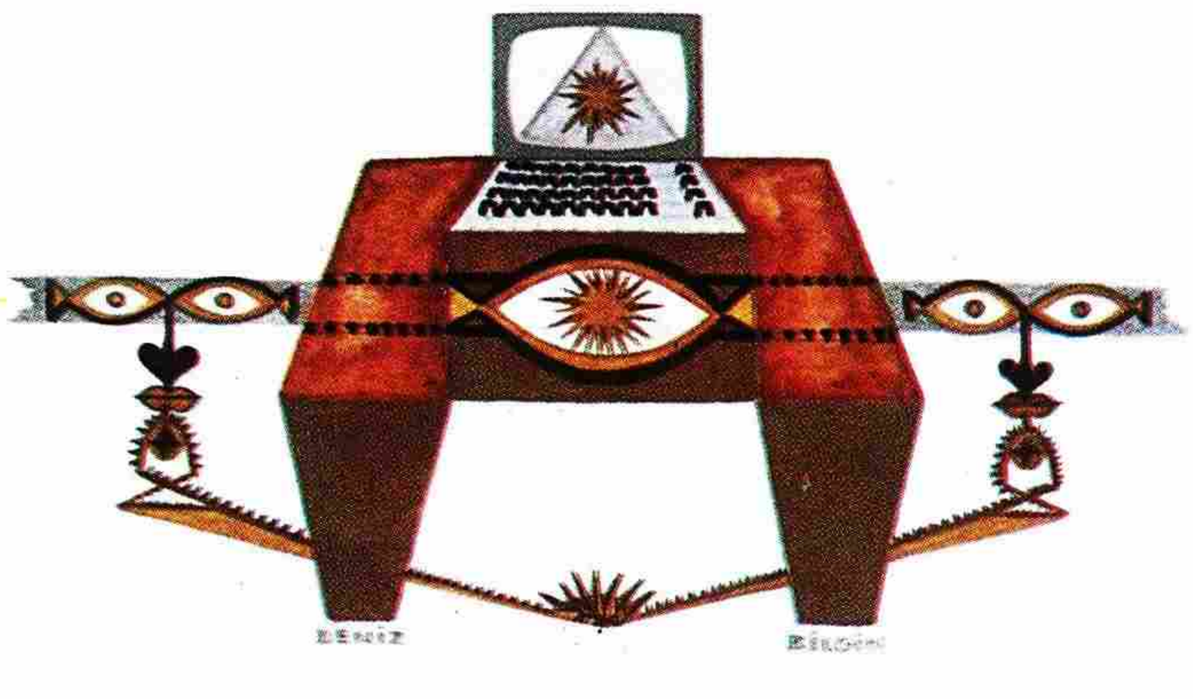
Hannover'da düzenlenen bu forumda değişik meslek gruplarından gelen kadınlar bilgisayarla olan ilişkilerinde edinmiş oldukları deneyimleri naklettiler. Nürnberg'li Emmi Schalk 25 elemanlı bir berber salonunun şefi olarak çalışmaktadır. Onun gündüzleri bayan müşterilerine hasredilmişken, akşamları da yıllar yılı yönetime ilişkin işlerle dolmuştur. Etkin bir işletme yönetimi için kasa tutma, ücretlerin hesaplanması türünden işlerin ve personelin verimliliği gibi işletme ekonomisi açısından gerekli istatistiklerin olabildiğince rasyonel bir tarzda organize edilmeleri zorunludur. Bayan Schalk bugün artık bilgisayarsız personel işleri ve iş verimliliği konularında sağlıklı bir görüşe nasıl varabileceğini tasarlayamamaktadır bile.

Claudia Bade ve Sigrid Waitkus'a gelince, bunlar Hannover'deki bir okulun kız öğrencileridir. Bu iki öğrenci ev idaresi yöneticiliği dalındaki eğitimleri sırasında ve bir model denemesi çerçevesinde gelecekteki işleri için önemli sayılan alanlarda bilgisayarla çalışmasını nasıl öğrendiklerini açıkladılar. Onların orta düzeyde yönetici konumunda çalışmak istedikleri büyük mutfakların, hastanelerin ve




yurtların ambar idaresinde, maliyet hesaplarında, menü ve iş planlamalarında bugün bilgisayarlar kullanılmaktadır. Kendi eğitim düzenlerinde ise bu modern elektronik aygıtlardan yararlanmak öngörülmemiştir bile. Bilgisayara egemen olmaları, bu iki öğrencinin bir işyeri bulma şanslarını şüphesiz artıracaktır.

Augsburglu otelcilik uzmanı Bayan Irene Strasser, bilgisayarla karşılaşmasını bir rastlantıya borçlu olduğunu vurguladı: Lokantasını bilgisayarla yöneten şimdiki şefinin tekniğe karşı duyduğu hayranlığa. Bayan Strasser'in söylediğine göre, meslek okulunda bilgisayar hiç sözkonusu olmamaktadır. Orada olsa olsa mekanik bir yazı makinesiyle çalışmaktadır. Bugün gastronomi alanında da, örneğin ambar idaresi, kasa tutma, işlerin denetimi, menünün planlanması ve maliyet hesabı türünden görevlerin bilgisayar olmaksızın ekonomik bir tarzda üstesinden gelinemeyeceği kesin olarak görüldüğü halde, otelcilik uzmanlığı dalında yapılan bitirme sınavında bile veri işleme konusunda adayların sorumlu tutulmaları için yetkililerce yeterli bir neden bulunmamaktadır.



Mesleki eğitimleri çerçevesinde elektronik araçları öğrenmek, şimdilik kadınların kendi insiyatiflerine kalmıştır. Özellikle metinlerle ve yazılı belgelerle çalışılan alanlarda bilgisayar gitgide vazgeçilemez bir unsur olmaktadır. Bayan sekreterler, çevirmenler ve gazeteciler etkinliklerinin bazı bölümleri için gelecekte en azından bilgisayara ilişkin temel bilgilerin öğrenilmesinin şart olacağını ifade etmişlerdir.

Bu örnekler neyi göstermektedir?

Somut bir hedefe ulaşmak istediği için bilgisayarla ilgilenen pek çok kadın olduğu açıktır. Genç kızlar ve kadınlar delikanlılara ve yetişkin erkeklere oranla bilgisayarların pratik yararları ve kullanım imkânları üzerine çok daha ısrarlı sorular sormaktadırlar. Ama bir başka ve gözardı edilmemesi gereken olgu da, delikanlıların ve yetişkin erkeklerin bugün için oyunda ve programlamada kadınlardan daha baskın olduklarıdır. Programların uygulanması sözkonusu olduğunda ise, cinslere ilişkin özgül (spesifik) bir ayrım açığa çıkmamaktadır.

Kadın-erkek rollerine ilişkin davranış kalıplarının (teknik ve doğa bilimleri her şeyden önce erkek işidir) genç kızların ve kadınların bilgisayar dünyasına girmelerini zorlaştırdığı açıktır. Kendisiyle oynamak ve kendisine egemen olmak istenilen bir makine olarak bilgisayarın büyüsünü yaşantılarına katanlar daha çok erkekler olmuştur. Kadınların motivasyonu ise bu denli kolay olmamaktadır; onların bu işe çok daha pragmatik bir tarzda el attıkları bellidir.

Ama tam da kadınların bilgisayara bu farklı yaklaşımları, gelecekteki gelişmeleri etkileyecek ve değiştirecek imkânlar sunmaktadır. İşin bu yönü kadınlar forumuna katılanların pek çoğu tarafından defalarca vurgulanmıştır. Gerçi şimdiye değin kadınlar çok ender olarak bir teknik geliştirmişlerdir, ama teknik var olalı beri kadınlar her zaman teknik aygıtların kullanımı ve uygulanmasıyla uğraşmışlardır. Daha az vasıflı ve daha az ücretli etkinliklerde bulunmuş olsalar da böyledir bu.

Federal Almanya Araştırma ve Teknoloji Bakanlığı'ndan Bayan Irene Rüde, şöyle bir uyarıda bulunmuştur: "Edilgin kadın kullanıcılardan oluşan bir ordu kurulmamalıdır." Bayan Rüde, şimdiye değin kadın rolüne atfedilmiş olan kendiliğindenlik, duyarlılık, sosyal yetenekler ve hattâ tensel duyarlılık gibi özelliklerin de yeni tekniklerle ilişkide ve bu tekniklerin geliştirilmesinde payları bulunmasının iyi olacağını söylemiştir. O zaman kadınların da, bilgi işlem yapmanın budala bir ekran çalışması ya da tekdüze bir veri işleme anlamına gelmediğini keşfedebileceklerini vurgulamıştır.

Hiç şüphesiz, yeni tekniklerin kadınların meslekleri üzerine en güçlü etkilerinin duyulduğu yerler bürolar ve hizmet sektörleri olacaktır. Buralarda her şeyden önce yazma etkinlikleriyle sınırlı işyerlerinin köklü bir tarzda rasyonelleştirilmeleri gerekecektir. Buna karşılık da yeni görevler belirecektir, daha ilginç olan, ama artık böylesine çok olmayan görevler. İşyeri bilgisayarlarıyla ilişki, bir yazı makinesinin kullanımından daha başka yetenekler gerektirir. Artık önemli olan sadece hatasız, çabuk yazmak değildir; çünkü yazım hataları çabuk ve kendiliğinden düzeltilebilmektedir. Bundan böyle istenen şey yüksek bir soyutlama yetisidir: Metnin nasıl oluştuğunu tasarlayabilmek, bilgisayar sisteminin dilinden anlamak, görece çabukluk-

ta benzer sistemlerle de doğru bir ilişkiye girme yeteneğini geliştirmek zorunluğu belirmektedir.

Bugün bile büro, yönetim ve hizmet alanlarında kadınların etkin olabilecekleri büyük işler mevcuttur: Uzmanlık isteyen şube şeflikleri, kamu hizmetlerinde yöneticilik ve akademik kariyere, gazeteciliğe, sanata ilişkin etkinlikler. Nitelikli konumlarda çalışmaya devam edebilmek ya da bu konumlara ulaşabilmek için, kadınların bu alanların istediği bilgilerle donatılmış olmaları gerekmektedir. Gelecekte bütün bu alanlar için, metin işleme sistemleriyle, muhtemelen de grafik programlarıyla çalışmak ve veri iletişimine daha önce telefona nasıl egemen olunduysa öyle egemen olacak bir formasyon edinmek, mesleğin gereklerinden sayılacaktır.

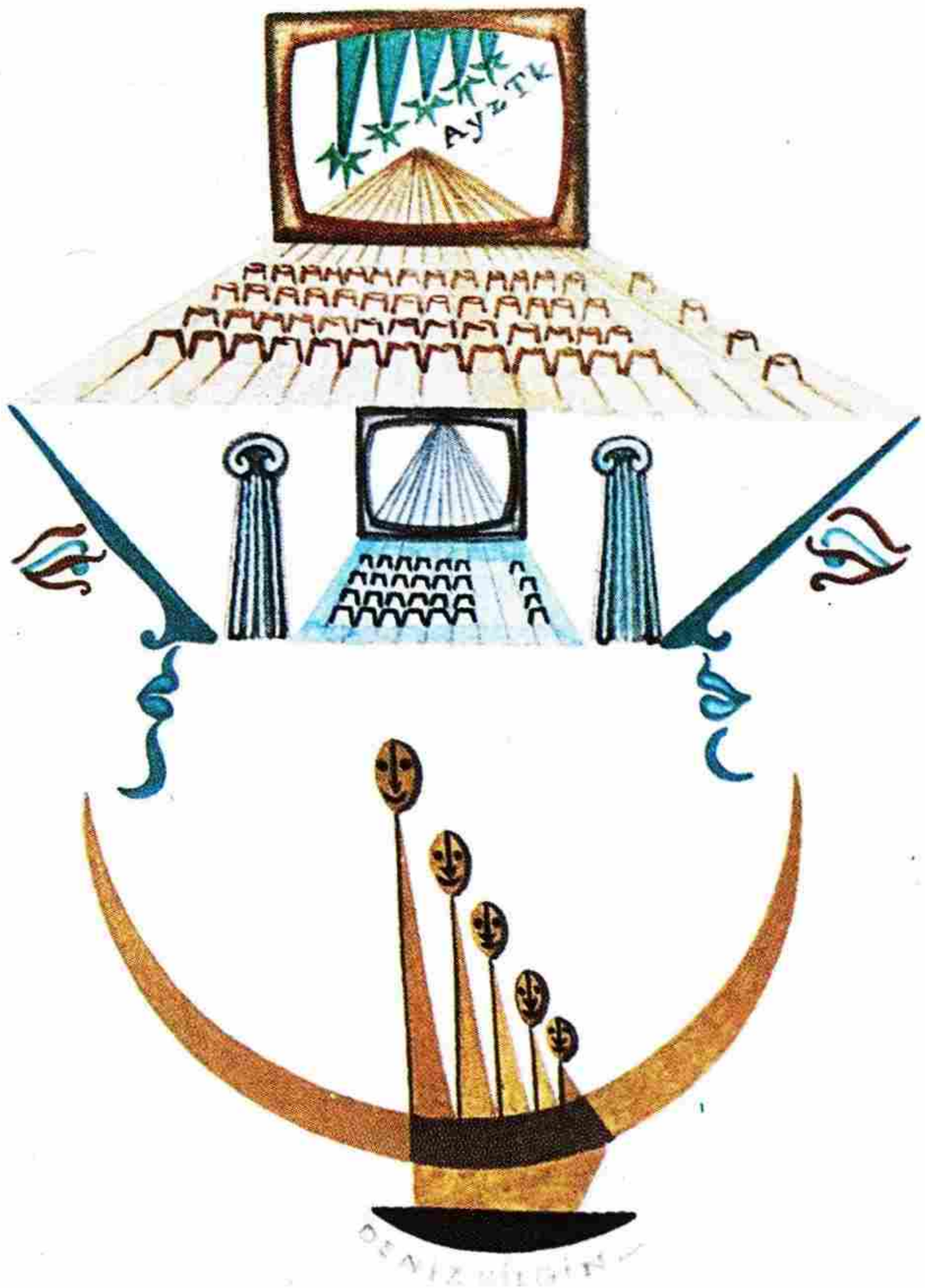
Kadınlar, almış oldukları eğitim nedeniyle, teknik ilerleme konusunda erkeklerden daha kuşkucu ve eleştiricidirler. Bunda haklıdırlar da. Ama bu yeni tekniğin sonuçlarından sadece kaygı duymak yetmez. Kadınlar, eğer bilgisayarların kötü ücret ödenen kullanıcıları olarak kalmak istemiyorlarsa, teknik yetenekler ve beceriler edinmek zorundadırlar.

Bunun için de, ailede ve özellikle okulda bu tür yetenekleri teşvik eden yöntemler geliştirilmelidir. Federal Alman Bayan sekreterlerinin yöneticisi Annelore Schliz ise, bu talepler sözkonusu olduğunda, devlet tarafından alınacak önlemlere bel bağlanmaması gerektiğini öne sürmüştür.

Federal Almanya'da pek çok eleştirel zihniyetli kişi ve hattâ sendikalar bu işin yalnızca tehlikelerini öne çıkarıyorlar: Sanayi Sendikası'nın (IG) Kimya Şubesi Yönetim Kurulu üyesi Eva Kaiser bir yıl önce şöyle demiştir: "Yeni teknolojilerle gitgide daha geniş alanlara yayılan ve daha hızla ilerleyen rasyonalizasyon ve otomasyon süreci, kadınların işyerlerini erkeklere oranla daha fazla yok etmektedir; çünkü kadınların büyük bir bölümü en önce otomasyona uğramaya elverişli tekdüze işlerde çalışmaktadırlar öteden beri." Buna ayrıca bir dehşet tablosuyla yaratılan tecrid edilmiş, işçi olarak haklarından yoksun bırakılmış, elektronikleşmiş ev zanaatkârı kadın imgesi eşlik edince, bilgisayar tekniğinin kadınlar için de yarattığı imkânlar çok az çözümlenebilmekte ve aceleyle elden çıkarılmaktadır.

Gerçekte ise, bu alanda ilerde de bir şeyler yapılmazsa, kadınların meslek yaşamlarındaki zaten mevcut olan mağduriyetleri yeni tekniklerle daha da artmış olacaktır.

Ancak, Batelle-Enstitüsü'nce hazırlanan "Kadınların Çalışması ve Yeni Teknoloji" konulu araştırmanın da ortaya koyduğu şu gerçeği dikkate almak gerekmektedir: Kadınların mağduriyeti tekniğin kendisine yüklenemez. Unutmayalım ki, "erkekliği" bunca baskın gözüken bilgisayar tekniğinin ilerlemesinde bu alanın "yumuşak cephesi"ni oluşturan ve bilgisayarı son derece esnek bir araç kılan yazılım (Software) kadınlar tarafından geliştirilmiştir. İngiliz şair Lord Byron'un kızı olan Lovelace Kontesi Ada, 19 yy'da ilk modern bilgisayarın taslağını yapan Charles Babbage'nin mesai arkadaşıydı. O, ilk kadın programcı olarak kabul edilmektedir. İlk Amerikan bilgisayarları MARK I ve ENIAC'ın programlanmalarını 40'lı yıllarda savaş hizmetiyle yükümlü kadınlar hazırlamışlardır. Bu kadınlar ilkin yoğun bir kursa tâbi tutulmuşlar ve "ENIAC-Girls" yarışmasına katılmışlardır. Şunun da gizlenmemesi gerekir: Bu kadınlardan birkaçı daha sonra "Manhattan-Projesi"nde de programlama yapmışlardır ki, bu proje ilk atom



bombasının geliştirilmesi olarak adlandırılmıştı.

Bilgisayarın ilk günlerinde aygıt geliştirilmesiyle uğraşan erkeklerin gözünde, bilgisayarların programlanması can sıkıcı bir büro çalışması idi. Daha yüksek düzeyde programlama dilleri henüz yoktu. Amerikan Kuvvetleri'nde komutanlık yapan ve daha sonra sanayi dalının en üst düzeyde programcılarından biri olan matematik Profesörü Grace Hopper, 50'li yılların başlarında derleyici (Compiler) üstüne ilk bilimsel incelemeleri yayımladı. Bilgisayar dili COBOL'un geliştirilmesinde Bayan Hopper'in önemli payı olmuştur. Gene de bu tür kariyerler istisna olarak kalmıştır. Erkekler savaş sonrası evlerine döndüklerinde "ENIAC-Girls"ün kadınlarına artık eskisi kadar talep yoktu. Modern bilgisayarcılığın başlangıç dönemlerinde uzman kişilerin aranmasına rağmen, okullardaki sistematik mesleki eğitim ihmal ediliyordu. İlk sayısal (digital) bilgisayarlardan biri olan UNIVAC I'i programlayan Bayan Ruth Davis gibi parlak bilgisayar-yıldızları ancak tek tük çıkıyordu ortaya. Bayan Davis, Data Processing Management Association'ın koyduğu ve çoğu kişinin göz dikmiş olduğu ödülü kazandı ve ironiye bakın ki "Yılın Adamı" seçildi (Man-of-the-Year).

Konumuza dönersek: Bilgisayarın kadınlar için ya işsiz kalma ya da tekdüze bir zorunlu iş anlamına geldiği düşüncesi aşılmalıdır. Öte yandan, "Kadınlar ve Bilgisayar-Üçüncü Görüşte Aşk mı?" temalı ve Commodore-Teknoloji Forumu türünden düzenlemelerin hiçbir reçete çözüm sunmadıkları bilinmelidir. Gene de bu tür toplantıların, kadınların bilgisayarın meydan okumasına nasıl karşılık vermek istediklerini kamu önünde düşünebilmeleri bakımından önemi vardır. Ama kadınlar bu yüzden bilgisayarı sevmek zorunda da değillerdir, hattâ üçüncü görüşte bile olsa... □

*Çev: Güven Savaş KIZILTAN*

# Makine Diline Merhaba (3)

Ayhan KALAYLIOĞLU

## 5. Mantıksal İşlem Komutları

Mantıksal işlemler de, aritmetik işlemler gibi iki değer arasında yürütülür. Aritmetik işlemlerle diğer bir benzerlik, operand'lardan birinin akümülatör'de bulunma zorunluğu. Diğer operand için tanıdık adresleme tiplerimiz mevcut. Mantıksal işlem yapıldıktan sonra sonuç burada da akümülatör'e alınmakta. 6510 ana işlem ünitemiz üç türlü mantıksal işlem yapabiliyor.

**VE (and) işlemi:**

Bu işlem akümülatör ve diğer operand arasında bit bazında yürütülür. Bazılarınızın daha evvelden biliyor olabileceği 've' işlemi, aslında iki adet birer bitlik veriler arasında yürütülür. Bu işlemin sonucu yalnızca her iki veri de '1' olduğunda '1'dir. Bunun ne anlama geldiğini ufak bir mantıksal tablo yaparak inceleyelim:

```
0 AND 0 = 0
0 AND 1 = 0
1 AND 0 = 0
1 AND 1 = 1
```

Bit bazında 've' işlemini gördükten sonra, şimdi de byte bazında bu olay nasıl gerçekleşiyor, gelin hep beraber bir örnek üzerinde görelim:

```
AND #$37
```

Diyelim ki akümülatör bu işlemden önce $5D içermekte, işlem şöyle gerçekleşir:

```
$5D 01011101
$37 00110111
-----------
$15 00010101
```

Sonuç olarak %00010101 yani $15 elde ettik. Bu işlemi BASIC'te yaptırmak için tabiî ki AND komutunu kullanacağız.

```
A = A AND 55
(37)
```

Burada A'nın değeri; A=$5D AND $37, yani A=93 AND 55'tir. Sonuç olarak 21, yani $15 elde ederiz. AND işlemi sırasında N- ve Z-flagları etkilenir. Sonuç olarak sıfır elde edersek, Z-flag'ı yanar. Sonucun 7'nci biti yanıksa N-flag yanar.

Aşağıda AND işlemine ait komut kodlarını ve adresleme tiplerini bulacaksınız.




| Adresleme tipi | AND |
|---|---|
| direkt | $29 |
| absolut | $2D |
| 0. sayfa | $25 |
| X-indeksli | $3D |
| Y-indeksli | $39 |
| 0. sayfa X-indeks | $35 |
| indirekt indeksli | $31 |
| indersli indirekt | $21 |

**VEYA (or) işlemi**

Bu işlem de akümülatör ve veri arasında bit bazında yürütülür. Şayet iki bitten bir tanesi bile '1'se sonuç '1' olur. Küçük bir tabloda inceleyelim:

```
0 ORA 0 = 0
0 ORA 1 = 1
1 ORA 0 = 1
1 ORA 1 = 1
```

Bu veya işlemine 1, 1 kombinasyonunu da içerdiği (sonucu olumlu olduğu) için aynı zamanda 'içeren veya' (inclusive or) denir. 'İçermeyen veya' (exclusive or) işlemini birazdan göreceğiz. VEYA işlemi sırasında da N- ve Z-flag'ları etkilenir. Şimdi işlemimiz byte bazında nasıl yürüyor, onu görelim:

```
ORA #$37
```

Diyelim ki akümülatör $5D içeriyor. İşlem şöyle gelişir:

```
$5D 01011101
$37 00110111
-----------
$7F 01111111
```

Sonuç olarak %01111111 yani $7F elde ettik. Aynı işlemi BASIC'te yaptırmak isteseydik OR komutunu kullanmamız gerekecekti:

```
A = A OR 55
(37)
```

Burada A'nın değeri A = $5D OR $37, yani A = 93 OR 55 işlemiyle belirlenir. Sonuç ise 127, yani $7F'dir.

Aşağıdaki tabloda ORA işleminin komut kodlarını ve adresleme tiplerini bulacaksınız.

| **Adresleme tipi** | **ORA** |
|---|---|
| direkt | $09 |
| absolut | $0D |
| 0. sayfa | $05 |
| X-indeksli | $1D |
| Y-indeksli | $19 |
| 0. sayfa X-indeks | $15 |
| indirekt indeksli | $11 |
| indeksli indirekt | $01 |

**İÇERMEYEN VEYA (exclusive or, EXOR) işlemi**

Exor işlemi sırasında da, bit bazında yürür tüm işlem. Sonuç, verilerden sadece biri '1' değeri taşıyorsa '1'dir. Değer tablomuzda bu olayı daha yakından görelim:

0 EOR 0 = 0
0 EOR 1 = 1
1 EOR 0 = 1
1 EOR 1 = 0

Görüldüğü gibi bu ilginç işlemde sonuç sadece iki değer birbirinden farklı ise olumludur (= 1). Bu işlem sırasında da N- ve Z-flag'ları etkilenir. BASIC'te bu işlemi yapan bir komutumuz olmadığı için ufak bir döngüde, bu işlem bit bit gerçekleştirilmeli. Böyle bir örnek burada sadece kalabalık yaratmaktan başka bir şey yapmayacağı için böyle bir örneği meraklı programcı arkadaşlarımızın özel zevkine bırakıyoruz. Şimdi byte bazında olayı, bir örnekle görelim:

EOR #$37

Burada da akümülatör $5D içersin. Bakalım neler oluyor:

$5D 01011101
$37 00110111
---
$6A 01101010

Sonuç olarak %01101010 yani $6A (106) elde ederiz. Bu komutun en çok kullanıldığı biçimini belirtelim. Bu komut genelde akümülatör'ün belli bazı bitlerinin içeriğini tersine (0 =>1 veya 1 =>0) çevirmekte kullanılıyor. Örnek olarak AND #$FF işlemi akümülatör'ün içeriği ne olursa olsun, tüm bitlerini tersine çevirir.

Aşağıdaki tabloda EOR işleminin adresleme tiplerini ve komut kodlarını bulacaksınız.

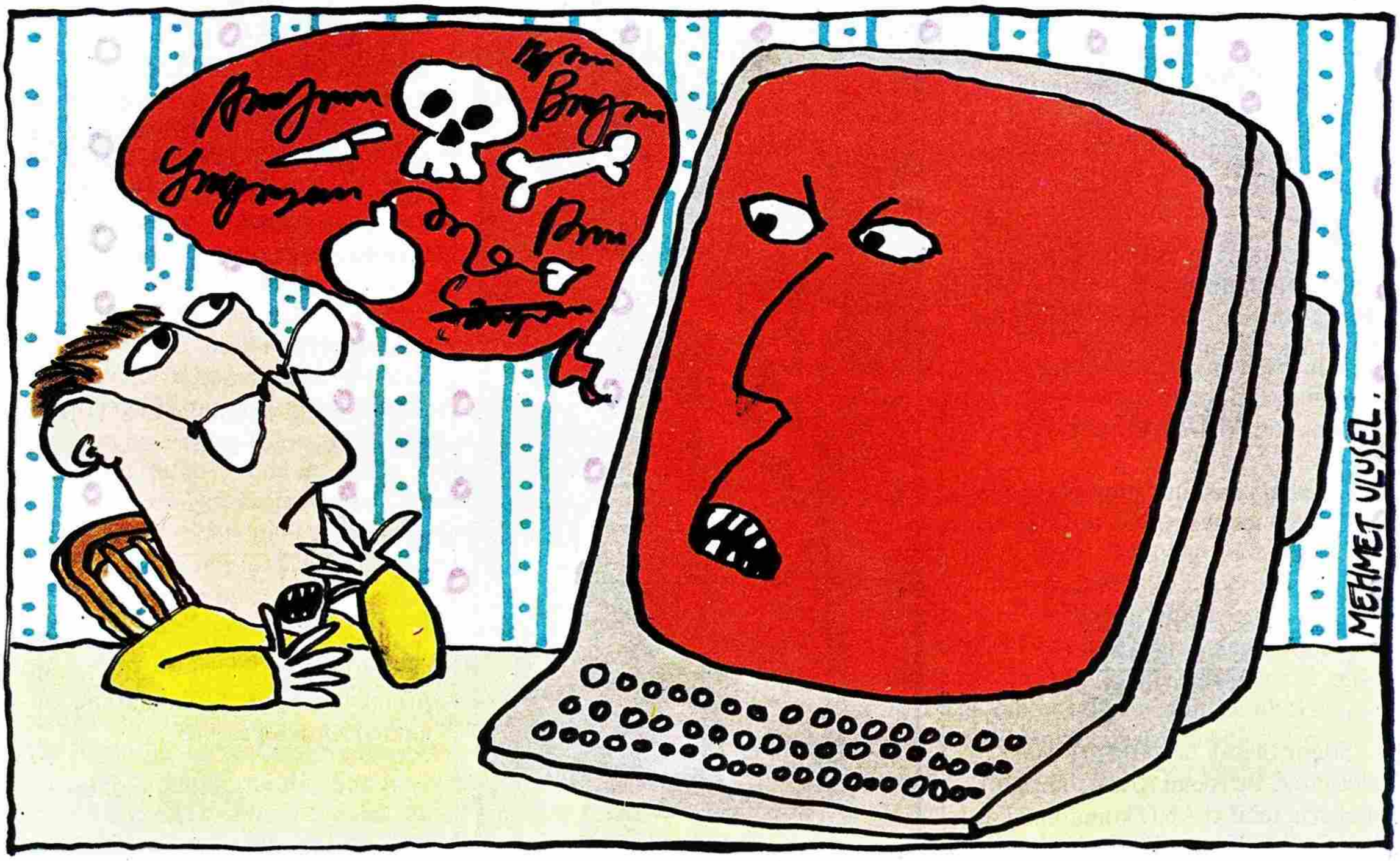
| **Adresleme tipi** | **EOR** |
|---|---|
| direkt | $49 |
| absolut | $4D |
| 0. sayfa | $45 |
| X-indeksli | $5D |
| Y-indeksli | $59 |
| 0. sayfa X-indeks | $55 |
| indirekt indeksli | $51 |
| indeksli indirekt | $41 |

**BIT komutu**

BIT komutu 65XX ailesinden olan ana işlem ünitelerinin değişik bir özelliğidir. Bu işlem hiçbir register'in içeriğini değiştirmez. Sadece flag'larını yürütülen işleme göre değiştirir. Etkilenen flag'lar Z-, N- ve V-flag'larıdır. Z-flag'ı akümülatör'le, veri arasında yürütülen hayali bir 've' (AND) işlemine göre şekillenir. N-flag'ı verinin 7'nci bitinin kopyasıdır. V-flagı da 8'inci bitle aynı içeriktedir. Bir kere daha hatırlatalım: Bu işlem sırasında ne register'lerin içeriği ne de hafızadaki verinin içeriği değişmez. Sadece bit'lere göre test yapılır ve sonuç flag'lara kaydedilir. Başka hiçbir şey değişmez. Örneğimiz bu olayı daha iyi açıklayacaktır:

BIT $1234

Diyelim ki akümülatör'ümüz $10



içeriyor. $1234 adresinin içeriği de $43 olsun. Yürütülen hayali 've' işlemi şu sonucu verir:

$10 00010000
$43 01000011

$00 00000000

AND işlemi, sonuçta sıfır verdiğine göre Z-flag'ı yanacaktır. V-flag'ı verinin 6'ncı bitine eşit olacak demiştik, yani '1'. Verinin 7'nci biti '0' olduğu için N-flag'ı sönük olacaktır. Sonuçta şunu elde ederiz:

Z = 1 ; V = 1 ; N = 0.

BIT komutu sadece iki adresleme tipine sahiptir:

| **Adresleme tipi** | **BIT** |
|---|---|
| 0. sayfa | $24 |
| absolut | $2C |

## 6. Karşılaştırma Komutları

Bu gruba giren komutların yaptığı işlem; register'lere zarar vermeden içeriklerini hafızadaki verilerle karşılaştırmamızı sağlar. Sonuç BIT komutundaki gibi flag'ları etkiler. Flag'ların durumuna göre programın akışını değiştiren komutlar (bu komutları bir sonraki bölümde göreceğiz) yardımıyla, bir bakıma IF...THEN...ELSE yapısına sahip program akışı oluşturabiliriz.

Karşılaştırma komutlarında verinin içeriği sözkonusu register'in içeriğinden çıkartılır (aritmetik olarak). Bu işlem sırasında register'lerin içeriği değişmez. Fakat sonuca C-, N- ve Z-flag'ları belirlenir. Ana işlem ünitemizin üç çalışma register'i için de karşılaştırma komutları mevcuttur.

**CMP komutu**

Bu komut (CMP: to CoMPare) akümülatör'ün içeriğini hafızadaki veri ile karşılaştırmaya yarar. Operand (veri), akümülatör'den çıkarılır (CARRY'ye dikkat etmeksizin). Şayet sonuçta 'alma' olayı gerçekleşirse CARRY biti söner, tersi durumda da yanar. Şayet sonuç sıfır ise, yani veri akümülatör'e eşitse, Z-flag'ı yanar, diğer durumlarda söner. Şayet sonuç $7F'ten büyükse (7'nci biti yanıksa), N-flag'ı yanar, yoksa söner. Gene bir örnek verelim:

CMP #$30

İlk örneğimizde akümülatör $50 içersin. Yani $50-$30, işlemi yürütülecektir. Sonuç pozitif olduğu için (yani 'alma' olmadığı için), CARRY biti yanacaktır. Sayılar birbirine eşit olmadığı için Z- (zero) flag'ı sönecektir. Sonuç olarak $20 elde ederiz, bu sayı $7F'ten büyük olmadığı için N-flag'ı sönük kalacaktır. Sonuç olarak; şu görünümü elde ederiz.

C = 1 ; Z = 0 ; N = 0.

Diyelim ki akümülatör'ün içeriği $30. Burada CARRY biti gene yanacaktır, çünkü 'alma' olayı yoktur. Fakat sonuç sıfır olduğu için, ZERO biti yanacaktır. N-flag'ı gene yanmayacaktır, çünkü sonuç $7F'ten büyük değildir. Sonuç olarak flag'lar bu durumu alacaktır:

C = 1 ; Z = 1 ; N = 0.

Üçüncü örneğimizde, akümülatör'ün içeriği $10 olsun. Yapılan işlem $10-$30=$E0 biçiminde gerçekleşecek ve görüldüğü gibi 'alma' olayı olacaktır. C-flag'ı bu yüzden sönecektir. Tabiî Z-flag'ı da. Sonucun 7'nci biti yanık olduğundan ($E0=%11100000), N-flag'ı yanacaktır. Sonuç şudur:

C = 0 ; Z = 0 ; N = 1.

Pratikte sonuçlar şöyle değerlendirilir:

C = 1 ; anlamı: büyük veya eşit.
Z = 1 ; anlamı: eşit.
C = 0 ; anlamı: küçüktür.

Şayet 'büyüktür' şartını kontrol etmek istiyorsak, iki flag'a birden bağımlıyız. Bu durum:

Z = 0 ; ve C = 1 ; anlamı büyüktür.

Bir kere daha hatırlatalım: Karşılaştırma komutları register'lerin veya operand'ların içeriğini değiştirmez. Sadece flag'lar değişir. Flag'ları değerlendiren komut grubunu bir sonraki bölümde inceleyeceğiz.

Aşağıdaki tabloda CMP komutunun adresleme tiplerini ve kodlarını bulacaksınız.

| **Adresleme tipi** | **CMP** |
|---|---|
| direkt | $C9 |
| absolut | $CD |
| 0. sayfa | $C5 |
| X-indeksli | $DD |
| Y-indeksli | $D9 |
| 0. sayfa N-indeks | $D5 |
| indirekt indeksli | $D1 |
| indeksli indirekt | $C1 |

**CPX komutu**

CPX komutu CMP komutundan farksız çalışır. Burada karşılaştırılan register, akümülatör değil X-register'dir. Burada da X-register'inin içeriği değişmez. Yukarıda tüm söylenenler, aynen CPX komutu için de geçerlidir. Yalnız CPX komutunun o kadar çok adresleme tipi yoktur. Aşağıdaki tabloda bu adresleme tiplerini ve karşılık gelen komut kodlarını bulacaksınız.

| **Adresleme tipi** | **CPX** |
|---|---|
| direkt | $E0 |
| absolut | $EC |
| 0. sayfa | $E4 |

**CPY komutu**

Burada, gene yukarıda söylenenler geçerliliğini koruyor. Fakat, bu sefer karşılaştırılan Y-register'i. Bu komut için de sadece üç adresleme tipi geçerli.

| **Adresleme tipi** | **CPY** |
|---|---|
| direkt | $C0 |
| absolut | $CC |
| 0.cı sayfa | $C4 |

## 7. Şartlı Programlama Komutları

Şimdi öğreneceğimiz komutlar programlarımızın şarta bağlı olarak yürümesini sağlıyor. Şartları belirleyen şey ise daha evvelden tanıdığımız flag'lar. Flag'lardan yalnızca dört tanesi için bir karar verme olayı sözkonusudur. Bunlar: Z-flag'ı (Zero), N-flag'ı (Negative), C-flag'ı (Carry) ve V-flag'ıdır (Verflow). Şartlı programlama; şarta göre programın belli bir adrese dallanması biçiminde oluşur. Her bir flag için ikişer adet şartlı dallanma komutumuz mevcut. Birinci komut grubu şayet flag yanıksa (içeriği 'bir' ise) dallanıyor; ikinci gruptakiler ise flag sönükse (içeriği 'sıfır' ise) dallanıyor.

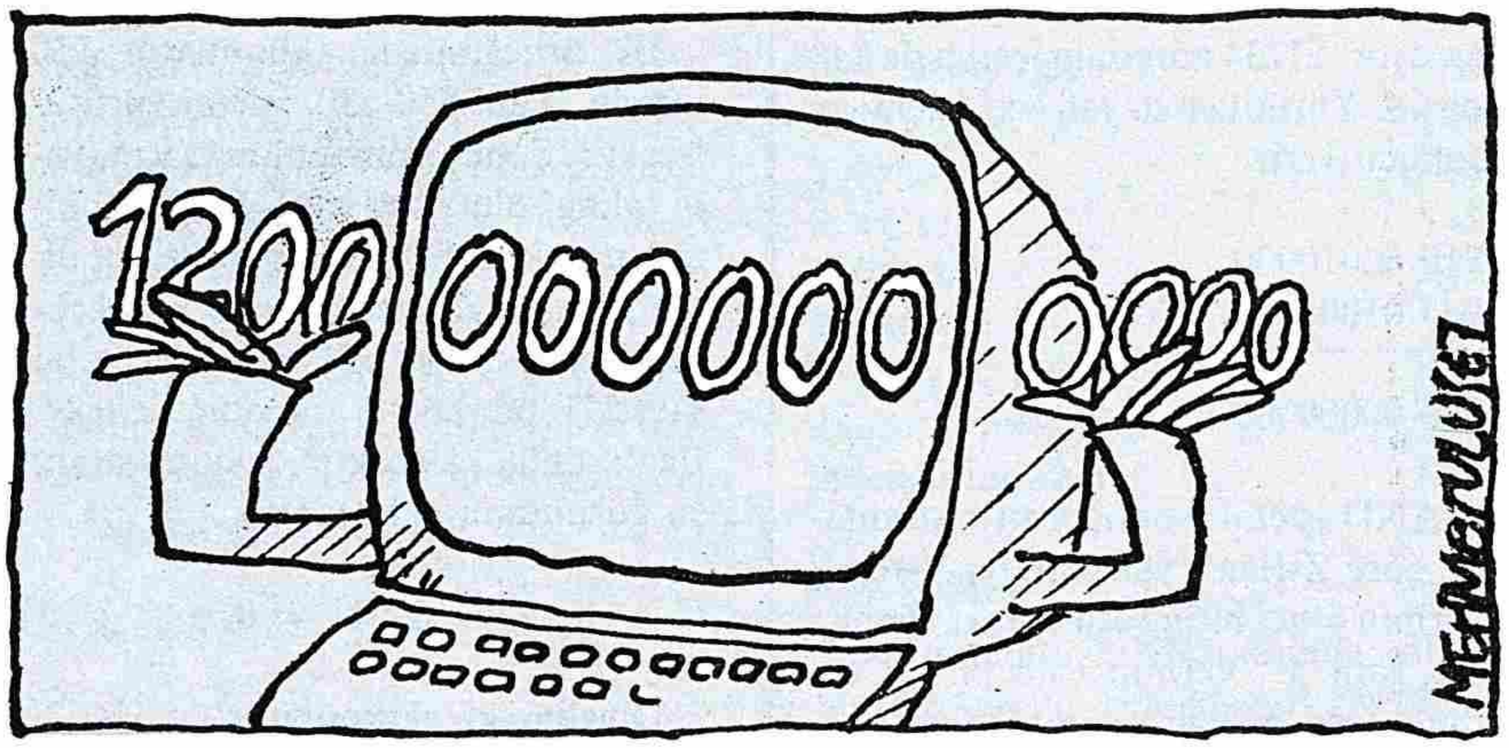
Şimdi bir olaya geri dönelim; şayet şart gerçekleşmezse hiçbir şey olmadan program devam edecek, fakat şart gerçekleştiğinde ise program belli bir adrese dallanacak. Peki bu belli adres neresi?.. Dikkatlice bakıldığında

bu adresin sözkonusu komutların hemen arkasından bir operand olarak belirtilmesi gerektiği ortaya çıkar. En basit çözüme örnek olarak, onaltı bitlik adresin hemen komutun arkasından gelmesi verilebilir. Böylece her bir şartlı dallanma komutu (branch) içeren satır üç byte sürecekti. Fakat ana işlem ünitemizin tasarımını yapan akıllı mühendisler daha pratik bir çözüm bulmuşlar. Bir programın dallanması genelde kısa bir mesafe içinde olduğu için yeni bir adresleme tipi geliştirilmiş: Relatif Adresleme. Bu adresleme tipinde yapılacak adres onaltı bit'lik absolut adresi ile değil, o adresin program sayacına olan uzaklığı ile belirtiliyor. Bu uzaklık ise yalnızca sekiz bit'lik bir değer ile temsil ediliyor.

Daha evvelden de bildiğimiz gibi sekiz bit'lik bir bilgi ile 256 değişik sayıyı temsil edebiliriz. En basit düşünce ile 256 byte ileri dallanabiliriz. Fakat çoğu zaman geriye dallanmak da gerekebilir. Örneğin: Bir döngünün başına. Bu yüzden sekiz bit'lik bir bilgiden negatif değerler de elde edebilmemiz lazım. Bunun için şöyle bir yöntem kullanılıyor: 7'nci bit'in değeri + 128 değil -128 olarak değerlendiriliyor. Yani bir bakıma yedinci bit sayımızın işareti ile ilgili bilgiyi de taşıyor. Aslında bu yöntem ile ilgili birçok örnek vermek mümkün. Fakat bütün bunların hiçbiri aşağıdaki gösterim kadar açıklayıcı olmayacaktır:

| | | |
|---|---|---|
| **%10000000** | **$80** | **-128** |
| **%10000001** | **$81** | **-127** |
| ... | . | .. |
| ... | . | .. |
| **%11111110** | **$FE** | **-2** |
| **%11111111** | **$FF** | **-1** |
| **%00000000** | **$00** | **0** |
| **%00000001** | **$01** | **1** |
| **%00000010** | **$02** | **2** |
| ... | . | .. |
| ... | . | .. |
| **%011111110** | **$7E** | **126** |
| **%011111111** | **$7F** | **127** |

Şimdi 7'nci bite baktığımızda görüyoruz ki, bu bit bir sayının negatif olup olmadığını belirliyor (Aynı zamanda N-flag'ın görevi). Şimdi bir örnekle relatif adreslemede adres farkını hesaplamayı görelim. Dallanma sırasında program sayacının içerdiği bir sonraki komutun adresi olur. Örneğimiz şu: Şartlı dallanma komutumuz




$C47A adresinde duruyor ve biz SC4BF adresine dallanmak istiyoruz.

$C47A Şartlı dallanma komutunun adresi
$C47C Bir sonraki komutun adresi
$C4BF Varış adresi

Şimdi basit bir çıkartma işlemi bize varış ile çıkış adresi arasındaki farkı verecektir:

$C4BF -$C47C = $43

Yani şartlı dallanma komutumuz için operand byte'ımızın değeri $43'-müş. Bir örnekle de geriye doğru sıçramayı inceleyelim. Diyelim ki $C440 adresine dallanacağız. Burada farkı belirlemek için değişik yöntemler sözkonusu. İlk yöntem bildiğimiz çıkartma işlemi, fakat sonuçta negatif bir sayı elde ediyoruz.

$C440 -$C47C = $FFC4 (alma var)

Elde ettiğimiz sonucun düşük değerli byte'ı yani $C4 işimizi görecektir, o'nu operand olarak kullanabiliriz. Diğer bir yöntem ise pozitif çıkartma işlemini yapmak ve ikili eşleniğini (komplementini) bulmak.

$C47C -$C440 = $3C

Bunun ikili eşleniği:

%00111100
%11000011

\+ 1
%11000100 = $C4

Gördüğünüz gibi gene $C4 değerini elde ettik.

Relatif adreslemenin bize sunduğu avantajlara gelelim: Öncelikle üç byte yerine iki byte kullanarak belli bir ölçüde hafızadan tasarruf ediyoruz. (Bazılarınız bu tasarrufu küçümseyebilir. Fakat şöyle bir açıklama kuşkuları giderecektir. Coğu compiler'ın ürettiği kod'da bu komutlar o kadar yoğundur ki bu komutlar üç byte'lık olsaydı program %5 ila %7 uzayacaktı.) Relatif adreslemenin en önemli avantajı absolut değerler içermemesidir. Adresler çıkış yerine bağıl (izafi) uzaklıklarına göre belirtilir. Şayet bu program parçasını hiç değiştirmeden başka bir bölgeye taşırsak çıkış ve varış komutları arasındaki fark değişmeyecektir (Fakat absolut adresler değişecektir).

Şayet absolut adres kullanmış olsaydık bu adreslerin de kaydırma işlemi sonunda yeni baştan elden geçirilmesi gerekirdi. Halbuki relatif adreslemede böyle bir sorun olmadığı için üretilen program 'transportable'-dır, yani başka bir bölgede de çalışabilir. (Tabii bu ölçü programın içinde kullanılan diğer programlama yöntemlerine de bağlıdır.)

Relatif adreslemenin dezavantajlarına bakarsak tek dezavantajın sınırlı adresleme kapasitesi olduğu görülür. Şartlı dallanma komutuna göre 128 byte öne veya 126 byte geriye sıçrayabiliyoruz. Pratikte bu durum pek sorun teşkil etmez. Nedeni ise daha uzun mesafelere sıçrama gereksinimimin pek az meydana çıkması.

Eğer relatif adreslemede operand hesaplamayı çok karışık bulduysanız, size sakin olmanızı tavsiye ederiz. Çünkü hemen hemen hiçbir zaman bu operand'ı siz hesaplamayacaksınız. Kullanacağınız tüm yardımcı programlar (gerek monitör programı, gerek assembler programı) bu işi üstlenirler. Siz bu komutları kullanırken sadece varış adresini verirsiniz, o program da o adresin izin verilen böl-

ge içinde olup olmadığını kontrol eder, şayet mümkün değilse sizi uyarır. Eğer mümkünse gerekli kodu üretir ve işleme devam eder.

Şimdi bu dört flag'a göre şartlı dallanma komutlarını görelim:

**Zero flag'ına göre dallanma**

Zero flag'ı yanık olduğu durumda dallanmak istiyorsak 'branch on equal' komutunu kullanmamız gerekir, kısaltılmışı BEQ. Zero flag'ı sönükken dallanmak istiyorsak 'branch on not equal' komutunu kullanacağız, kısaltılmışı BNE.

**Carry flag'ına göre dallanma**

Carry varsa dallanmak için 'branch on carry set' yani BCS komutunu, carry yokken dallanmak için ise 'branch on carry clear' yani BCC komutuni kullanmamız gerekmektedir.

**Negative flag'ına göre dallanma**

N-flag'ı yanıkken sıçramak için 'branch on minus' yani BMI komumutunu kullanmamız gerekmektedir. 'branch on plus' yani BPL komutunu kullanmamız gereklidir.

**Overflow (taşma) flag'ına göre dallanma**

Taşma varken dallanmak için 'branch on overflow set' yani BVS komutunu taşma yokken dallanmak için BVC komutunu kullanmamız gerekir. Bu gruptaki komutlar genelde diğerlerine göre daha seyrek kullanılır.

Aşağıdaki tabloda tüm şartlı dallanma komutlarını ve kod'larını bulacaksınız.

| Komut | Komut kodu |
|---|---|
| BEQ | $FO |
| BNE | $DO |
| BCS | $BO |
| BCC | $90 |
| BMI | $30 |
| BPL | $10 |
| BVS | $70 |
| BVC | $50 |

## 8. Sıçrama Komutları

Bir önceki bölümde gördüğümüz şartlı dallanma komutlarının tersine belli bir (absolut) adrese giden komutlara sıçrama komutları denir. Bu komutlar herhangi bir şarta bağlı değildir ve her seferinde yürütülürler. Sıçrama komutları absolut adresleme ile kullanıldığı gibi daha evvelden kısmen tanıdığımız başka bir adresleme tipine de sahiptir. İndirekt adreslemeli sıçrama. Bu komutun hemen arkasından gelen adres varış adresi değildir. Bu adres, varış adresini gösteren bir göstergecin adresidir. Burada varış adresi peşpeşe gelen iki Byte'ta daha evvelden tanıdık olduğumuz göstergeç formatında (Lo-Byte, Hi-Byte) bulunur. Bir örnekle görelim:

```
JMP ($0302)  $0302'deki göstergeç
             üzerinden indirekt
             sıçrama.
```

Bu emir yürütülürken varış adresi $0302 ve $0303 adreslerinin içeriğinden oluşturulur. Diyelim ki orada $40 ve $C8 bulunsun, böylece programımız $C840 adresine sıçrar ve oradan devam eder. Bu adresleme tipi yalnızca JMP komutuna aittir. Varış adresi olarak aletinizin mümkün olan tüm adreslerini belirtebilirsiniz. Aşağıdaki tabloda her iki komutun adresleme tiplerini ve emir kodlarını bulacaksınız:

| Adresleme tipi | JMP |
|---|---|
| absolut | $4C |
| indirekt | $6C |

COMMODORE 64'ün işletim sisteminde indirekt adresleme tipi oldukça yoğun şekilde kullanılmıştır. Hafızanın üçüncü sayfasında $0300 adresinden, $0330 adresine kadar sistem vektörleri yer almaktadır. Bu vektörlerin RAM'de bulunmasının oldukça fazla avantajları vardır. En basit örnek olarak bu vektörleri değiştirerek işletim sistemi ile ortak çalışan programlar yazabilme olanağı. Böylece bilgisayara yeni bir özellik kazandıracak programlar yazmak çocuk oyuncağı kadar basit bir olay haline geliyor.

## 9. Sayma Komutları

Döngüler ve sayaçlar içeren kuvvetli programlama tekniklerini kullanabilmek için 6510'un özel sayma komutları vardır. Bu komutlar sayesinde gerek register, gerek hafıza birimlerinin içeriğini bir artırabilir veya bir eksiltebiliriz. Artırma komutları fonksiyon bakımından BASIC'teki NEXT komutuna benzer. Eksiltme komutlarında ise 'STEP-1' komutu ile çalışıyormuş gibi düşünün.

**INX**

Bu komutla X-register'inin içeriği bir artırılır. Sonuca göre N- ve Z-flag'larının içeriği değişir. BASIC'te bir örnek verecek olursak:

X = X + 1

Diyelim ki artırmadan önce X-register'inin içeriği $FF olsun. Değer olarak 256'yı gösteremeyeceğimiz için gene sıfıra döneriz. Yani kısacası X-register'ine bir toplanıyor. Fakat bu toplama sırasında C-flag'ı gözönüne alınmıyor ve sonucunda 255 geçilince C-flag'ı da etkilenmiyor. Bu örnekte sonuç olarak sıfır elde ettiğimiz için Z-flag'ı yakılır.

**INY**

Bu komut Y-register'ini bir artırmak için kullanılır ve Y-register'inin kullanılması dışında geri kalan her şey yukarıdaki komutla aynıdır. Bu komutun BASIC'te benzeri:

Y = Y + 1

COMMODORE 64'ün ana işlem ünitesi 6510, akümülatör'ün içeriğini tek komutta bir artıran bir komuta sahip değildir.

**INC**

Bu komut belli bir adresteki byte'ın içeriğini bir artırmaya yarar. Sonuca göre N- ve Z-flag'ları etkilenir. Bu komutun şimdiye kadar gördüğümüz diğer komut gruplarından farkı, belli bir hafızanın içeriğinin önce ana işlem ünitesinin içine okunması orada bir artırılması ve tekrar yerine yazılmasıdır (Read-Modify-Write). Şimdiye kadar gördüğümüz komutlar ise bir hafızanın içeriğini ya okuyor, ya da yazıyordu; hiçbiri ikisini birden yapmıyordu. INC komutu yürütülürken akümülatör'ün, X-register'inin ve Y-register'inin içeriği değişmez. BASIC'te bu işi yapan satırı şöyle yazarız:

POKE M, PEEK(M) + 1

Burada M, artırılacak hafızanın adresini göstermektedir.

Şimdi hafıza ve register'leri bir eksilten komutları görelim.

**DEX**

Bu komut X-register'inin içeriğini bir eksiltir. Sonuca göre N- ve Z-flag'ları etkilenir. Diyelim ki X-register'inin içeriği $00'dı ve biz eksilttik. Bu durumda $FF elde ederiz ve hiçbir şekilde Carry flag'ı etkilenmez. BASIC'te aynı işi yapan satır şöyle olacaktır.

X = X-1

Tabii negatif sayı diye bir şey sözkonusu olmayacaktır.

**DEY**

Bu komut Y-register'inin içeriğini bir eksiltir. DEX komutu için söylenenler aynen geçerlidir. BASIC'te aynı işi yapan satır:

Y = Y-1

Burada da $00'dan $FF'e geçeriz.

**DEC**

Bu komut belli bir adresteki hafıza biriminin içeriğini bir eksiltmeye yarar. Bu işlem sırasında akümülatör'ün içeriği kesinlikle etkilenmez. INC komutu için söylenenler aynen geçerlidir.

Aşağıda komutları, adresleme tiplerini ve komut kodlarını bulacaksınız.

| Komut | Komut kodu |
|---|---|
| INX | $E8 |
| INY | $CA |
| DEX | $CA |
| DEY | $88 |

| Adresleme tipi | INC | DEC |
|---|---|---|
| absolut | $EE | $CE |
| absolut X-indeksli | $FE | $DE |
| 0. sayfa X-indeks | $F6 | $D6 |

## 10. Flag'ları Değiştirme Komutları

Flag'ların durumuna göre programın akışını değiştiren komutlardan başka, kullanıcının flag'ları istediği biçimde değiştirmesini sağlayan komutlar da mevcuttur. Daha evvel de örneklerini gördüğümüz gibi; toplama ve çıkarma işlemlerinden önce bu tip komutları kullanıp C-flag'ının durumu belirlenmelidir. Bu gruptaki komutlar operand'a ihtiyaç göstermezler ve bir Byte'a sığarlar.

**Carry flag'ı**

SEC ('set carry') komutu ile C-flag'ı yakılır. CLC ('clear carry') komutu ile de C-flag'ı söndürülür.

SEC komutu her çıkartma işleminden önce, CLC komutu ise her toplama işleminden önce kullanılmalıdır, yoksa yanlış sonuçlar elde ederiz. Tabii birkaç byte genişliğindeki bilgileri toplayıp, çıkarıyorsak ara işlemlerde C-flag'ı ile oynamayın, çünkü C-flag'ı, byte'ların birbirine birleşmesi sırasında gerekli bilgileri içerir. Carry flag'ı yalnızca ilk toplama veya çıkarma işleminden önce belirlenmelidir.

**Decimal flag'ı**

Bu flag toplama ve çıkarma işlemlerinin bildiğimiz ikili sistemde mi, yoksa ikili kodlanmış ondalık sistemde mi (binary Coded Decimals, BCD) yapılacağını belirler. BCD modunda işlem yapabilmek için C-flag'ının yanık olması gerekir. Konuya yeterince hakim olmadıkça bu modu kullanmanızı pek tavsiye etmeyiz. Daha ileride yayınlayacağımız 'PROGRAMLAMA TEKNİKLERİ' bölümünde, bu modun detaylarını açıklayıp örnek bir program geliştireceğiz. O zamana kadar sabır!.. D-flag'ını yakmak için SED ('set decimal') komutunu, bu flag'ı söndürmek için ise CLD ('clear decimal') komutunu kullanacağız.

**Interrupt flag'ı**

Interrupt denilen olay, programın yürürken, herhangi bir (hardware) donanımın gönderdiği sinyal yüzünden kesilmesi, özel bir program parçasının yürütülmesi, daha sonra da kesilmiş olan programa hiçbir şey yokmuş gibi olduğu gibi geri dönmesidir. Aslında yukarıdaki uzun cümle olayın çok basit bir özetidir. Daha ileride bu olaya çok daha detaylı olarak eğileceğiz. Interrupt flag'ı ınterrupt'a izin verilip verilmediğini belirler. I-flag'ını yakmak için SEI ('set interrupt disable') komutunu kullanırız ve durum da interrupt'a izin verilmez. Interrupt'a tekrar izin vermek için I-flag'ı söndürülmelidir. Bunun için CLI ('clear interrupt disable') komutu kullanılır.

**Overflow flag'ı**

Overflow flag'ı sadece söndürülebiliyor (Çünkü yakmanın pek bir anlamı yok). Bu işi yapan komut ise CLV ('clear overflow'). COMMODORE 64'ümüzün ana işlem ünitesi 6510'da bulunmayan, fakat 6502 ana işlem ünitesinde bulunan bir özellikten bahsedelim. Bu flag aynı zamanda ana işlem ünitesinin bir bacağına uygulanan bir sinyal tarafından da yakılabiliyor. Yani programdan bağımsız da etkilenebiliyor bu flag. Tekrar yakılabilmesi için söndürülmesi gerektiğinde CLV komutu kullanılıyor. Bilhassa 1541 Disk-Drive'de çok kullanılan bu özelliğe yazımızın ilerleyen bölümleriyle değineceğiz.

Aşağıdaki tabloda komutları ve komut kodlarını bulacaksınız.

| Komut | Komut kodu |
|---|---|
| CLC | $18 |
| SEC | $38 |
| CLD | $D8 |
| SED | $F8 |
| CLI | $58 |
| SEI | 578 |
| CLV | $B8 |

## 11. Bit Kaydırma Komutları

6510 ana işlem ünitemizin yürüttüğü başka bir komut grubu daha var ki, bu gruptaki komutlara karşılık gelen BASIC komutları yok, bu grup bit kaydırma komutlarını içeriyor. Sözkonusu işlem sırasında akümülatör'ün, ya da adreslenen hafıza biriminin bit yapısı bir bit sağa veya sola kaydırılıyor. Şayet kaydırılan şey akümülatör ise, akü adreslemesinden söz ediyoruz. Bu gruptaki komutlar adresleme tipine göre bir, iki, ya da üç Byte uzunluğunda olabiliyor. Şayet adresleme tipi bir hafıza elemanını gösteriyorsa, aynen INC ve DEC komutlarındaki oku-değiştir-yaz (Read-Modify-Write) işlemleriyle komut yürütülür. Bu işlem sırasında akümülatör'ün içeriği korunacaktır.

**ASL**

ASL anlam olarak 'arithmetic shift left'in baş harflerinden oluşmuş bir kı-

saltmadır. Yapılan işlem şudur: Adreslenen byte'ın içeriği bir bit'lik pozisyon olarak sola kaydırılır. Sıfırıncı bit'in yeri boş kalacağı için oraya '0' yazılır. Yedinci bit'teki bilgi ise dışarda kalır. O da Carry'ye alınır. Diyelim ki akü ile bu işi yaptık. Bir örnekle görelim:

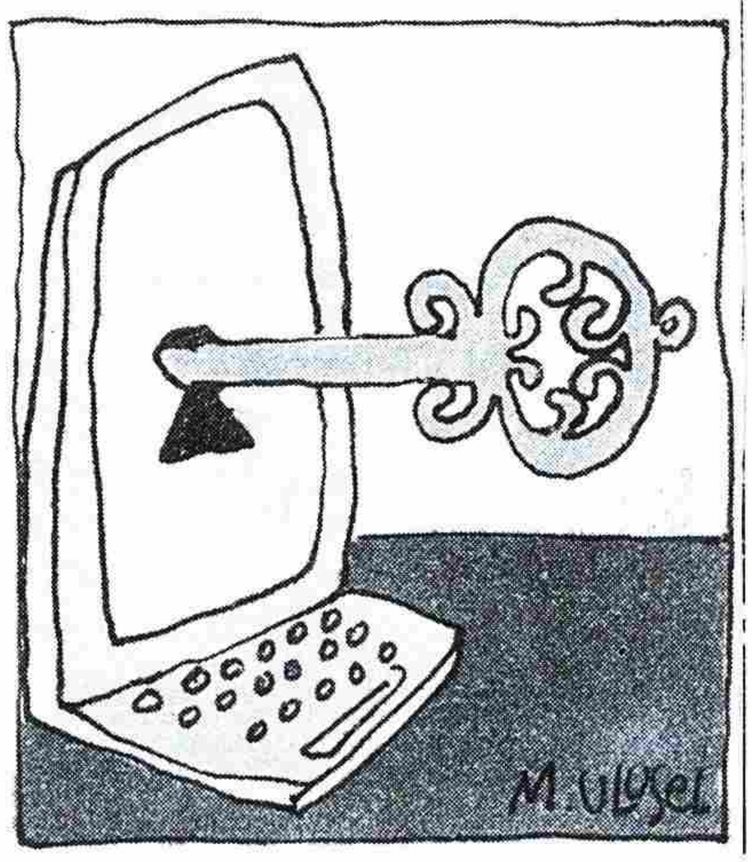
```
ASL A       Akü $47 içersin
$47   %01000111
      %10001110     $8E, C=0
```

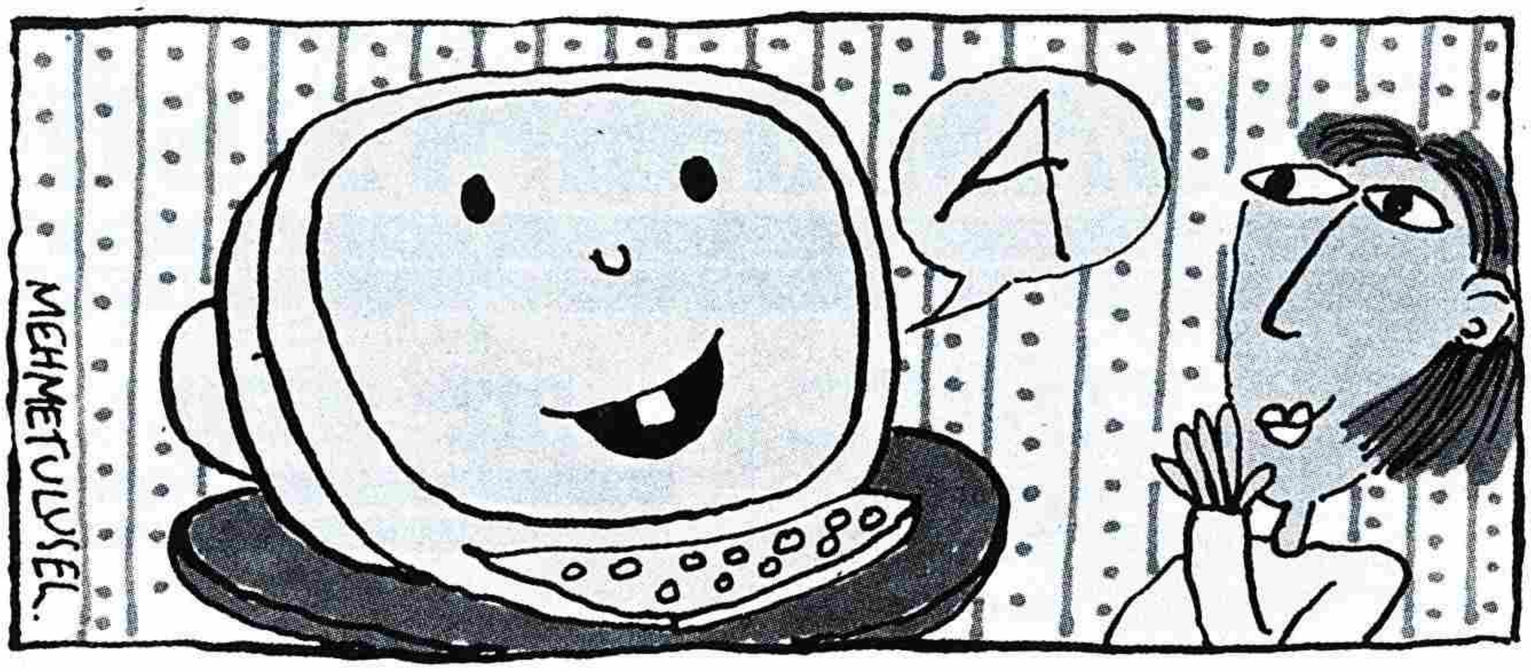
Bu örneğimizde sonuç olarak akümülatör'ün içeriği olarak $8E elde ettik ve Carry bit'i (flag'ı) söndürüldü, çünkü kaydırmadan önce akümülatör'ün 7'nci bit'i '0' içeriyordu. Akümülatör'ün kaydırmadan önce ve sonraki içeriklerini karşılaştırırsak görürüz ki akümülatör'ün içeriği iki katına çıkmış. Durumu daha iyi kavrayabilmek için ondalık sistemde düşünelim. Şayet ondalık bir sayıyı bir basamak sola kaydırırsak değeri on katına çıkar. Aynı olayın ikili sistem için de geçerli olduğunu gözönünde bulundurursak, demin elde ettiğimiz sonucun rastlantı olmadığını görürüz.

Yani ASL komutu ile bir sayıyı iki katına çıkarabiliriz. Bir örnek daha inceleyelim.

```
ASL A       Akü $CD içersin
$CD   %11001101
      %10011010     $9A, C=1
```

Carry flag'ını işlemin sonucunda sol taraftaki ekstra bit olarak görürsek, gene sayımızın değerinin iki katına çıktığı sonucuna varırız. $CD (205)'in iki katı $19A (410)'dur.




**LSR**

LSR komutu "logical shift right'-ın kısaltılmışı olup, ASL komutunun tersi işlemi yapar. Burada Byte'ın içeriği sağ tarafa kaydırılır. 7'nci bit'e '0' yüklenir. Dışarı çıkan 0'ncı bit ise Carry'e alınır. Bir örnekle görelim:

```
LSR A       Akü $CA içersin
$CA   %11001010
      %01100101     $65, C=0
```

Sonuç olarak akümülatör'de $65 ve sönük bir Carry bit'i elde ettik. Sizin de dikkatinizi çektiği gibi akümülatör'ün değeri yarıya indi. Carry bit'i kaydırmadan önceki sıfırıncı bit'in kopyasını içeriyor, başka bir deyişle Carry bit'i ikiye bölme sırasında artan olup olmadığını gösteriyor. Böylece bir sayının tek mi, çift mi olduğunu belirleyebiliriz. LSR en alttaki bit'i Carry'ye çıkarır, böylece BCC ve BCS komutları ile dolaylı olarak en alttaki bit'i test edebiliriz. Eğer LSR komutu bir hafıza elamanını adreslerse akümülatör'ün içeriği değişmez.

**ROL**

ROL komutu 'rotate left'in kısaltılmışından gelir ve bit'lerin içeriğini ise dizilmiş boncukları kaydırır gibi döndürür. Sola kaydırma sırasında boşalan sıfırıncı bit'in yerine Carry bit'i kopya edilir. Dışarı çıkan yedinci bit'in içeriği ise Carry bit'ine alınır. Daha genel bakarsak, yapılan işlem dokuz bit'lik bir bilginin kendi içinde döndürülmesidir (Byte'a ait 8 bit ve Carry bit'i). Bir örnek daha iyi kavramımızı sağlayacak:

```
ROL A       Akü $4B içersin
                Carry yanık olsun
$4B   %01001011     C=1
C=0   %10010111     $97
```

Bütün bit'ler bir basamak sola kaydı. Carry bit'i boşalan sıfırıncı bit'in yerine geldi. Dışarı çıkan yedinci bit Carry'ye alındı. Sonuç olarak $97 ve sönük bir Carry elde ettik. Burada da akünün değerinin iki katına çıktığını, fakat bundan başka mesela daha evvel oluşmuş bir eldenin veya daha evvelki bir kaydırma işleminden çıkmış Carry bit'inin de toplandığını görüyoruz. Bu son yorum oldukça ilgi çekici, çünkü Carry bit'i yalnızca toplama ve çıkarmada değil, aynı zamanda sağa ve sola kayırmalarda birkaç Byte'ı birbirine bağlamak için kullanılan bir ara değer taşıyıcısı durumunda bulunuyor.

**ROR**

ROR komutu ('rotate right') ROL komutunun tam tersi işlemi yapar. Bu sefer bit'lerin dönüş yönü sağa doğrudur. Bir örnekle görelim:

```
ROR A   FAkü $89 içersin
            Carry sönük olsun
$89   %10001001     C=0
$44   %01000100     C=1
```

Akümülatör'ün içeriği olarak $89'-dan $44 elde ettik. Carry'nin yanık olması bölme işleminde artan olduğunu gösteriyor. Kaydırma komutlarında Carry Flag'ından başka sonuca göre N- ve Z-flag'ları da etkilenir.

Aşağıdaki tabloda komutları ve adresleme tiplerini bulacaksınız.

| Adresleme tipi | ASL | LSR | ROL | ROR |
|---|---|---|---|---|
| akü | $0A | $4A | $2A | $6A |
| absolut | $0E | $4E | $2E | $6E |
| 0.cı sayfa | $06 | $46 | $26 | $66 |
| absolut X-indeksli | $1E | $5E | $3E | $7E |
| 0.cı sayfa X-indeks | $16 | $56 | $36 | $76 |

# OSAM'dan 87 SÜRPRİZİ

## BİZ SİZDEN YANAYIZ

## Hediye LIGHT PEN

**Commodore 64 setlerini OSAM'dan alanlara OSAM süper oyunlar paketinden 8 paketi birden alanlara LIGHT PEN'i hediye olarak veriyoruz.**
**Eski müşterilerimizi unutmadık. Onlarda LIGH PEN'lerini %20 indirimli alacaklar.**

## Süper Paket Programlar serisine devam

**Paket VII**

KAFA AYARI
EXPLODİNG FİST II
1942
ALLEYKAT
BOMBA
CYBORG
FISTFULOF BACK
HACKER II
MISSİON A.D.
DRAGONG LAIR
PAI BOXING
INFILTRATOR I
INFLITRATOR II
INFILTRATOR II
SATANAS HALLOW
JACK THE NIPPER
SPIKY-HAROD
SHOW JUMPING
SPINDIZZY
CORE
BOUNCES



**OSAM 16-22 Şubat tarihleri arasında Etap Marmara Oteli Tüyap Ticaret Merkezinde Commodore Fuarı'nda Fuar sürprizleri için OSAM standına uğrayınız.**

**Paket 8**

KAFA AYAR PRG
YIE AR KUNG-FU II.
STRİKE COBRA
TARZAN
AVENGER
FAKLAN'S 82
BASİLDON BOND
NEMESİS
WARHWK II.
PARKPATROL
BURGER TİME
CRACKER REVENGE I
CRACKER REVENGE II
WARLOK
PAP ERBOY
FORCE
GREMLİNS
CASİNO ROYALE
CANDALF SORCERER
GALİLAN
STAR PİNG-PONG
WORLD GAMES

**PAKET PROGRAMLARI..... 5500 TL**
**1 KASETTE 20 OYUN..... 1900 TL**
**1 KASETTE 40 OYUN..... 2750 TL**
**1 KASETTE 1 OYUN..... 1500 TL**
**COMMODORE T-SHIRT.....1200 TL**
LIGHT PEN.....10.000 TL.

1986 YILININ SÜPER PAKET PROGRAMLARININ ÜRETİMİ DEVAM ETMEKTEDİR.

7 SÜPER PAKETTEN BEŞ PAKETİ TOPTAN ALANA % 20 İNDİRİM UYGULANMAKTADIR.

# OSAM BİLGİSAYAR

Ebuzziya Caddesi Sakızlı Yalı Sokak No:3 Bakırköy- İstanbul Tel: 583 72 46

# Stereo-SID

**Ayhan KALAYLIOĞLU**

Daha evvel hiç hayal ettiniz mi? Bilgisayarımızın altı bağımsız ses kanalı olsaydı ve bunları stereo olarak kullanabilseydik! STEREO SID devremizle bu hayalinizi gerçekleştirebilirsiniz.

Her şey eninde sonunda sınırına ulaşır diye düşünülürdü eskiden. Fakat son zamanlarda Commodore 64'ün SID kanalları ile ilgili yeni uygulamalar insanı hayretten hayrete düşürüyor. Gerçekten de sınıra çoktan ulaşıldığını düşünenler şimdi büyük bir şaşkınlık içindeler. Bu yazımızda size Batı Almanya'da yapılmış çok ilginç bir uygulamayı aktaracağız. Bu uygulama ile Commodore 64 ikinci bir SID cipine daha kavuşuyor. Böylece yalnızca altı bağımsız ses kanalına sahip olmuyoruz; iki SID'in çıkışları ayrı olarak verildiği için gerçek stereo ses efektleri elde edebiliyoruz.

Artık yeni uygulama alanlarını düşünebiliriz. Diyelim ki aynı tonu hem sağ hem sol kanala aynı şiddette gönderdik, bu durumda sesi tam ortada algılayacağız. İki kanalın ses şiddet oranlarını farklılaştırırsak ses yavaş yavaş bir tarafa doğru kayacaktır. şayet iki kanal arasında belli bir zaman farkı (faz kayması) yaratırsak odamızın sınırları kalktığı ve sesin açık alanda yayıldığı hissine kapılırız. Bu kadarla da bitmiyor; şayet zaman farkını arttırırsanız bu sefer de HALL ve ECHO efektleri elde etmeniz mümkzün. Veya ses çıkış katınızı MONO pozisyonuna getirip aynı anda tek kanaldan altı ayrı sesi kullanabilirsiniz.

Müzikle yoğun şekilde uğraşanlar için daha birçok değişik imkân sunmakta bu uygulama. Görüldüğü gibi ikinci SID, imkânların iki katına çıkmasından çok daha fazlasını getiriyor.

Gelelim ekstra SID'in nasıl programlanacağına:

SID'in yerleştiği adres bölgesine göz atarsak ($D400-$D7FF) orada daha henüz boş olan yerlerin bulunduğunu görürüz. Aslında SID'in gereksinimi yalnızca 29 byte'lık bir bölgedir; fakat kendisine 1024 byte'lık bir yer ayrılmıştır. Daha da dikkatl iincelersek SID'in hafızalarının her 32 byte'ta bir tekrarlandığını görürüz. STEREO SID'in dizaynı başka bir görünüme yol açıyor. Artık 32 byte sonra ilk SID'in hafızalarının tekrarıyla değil yeni SID'in hafızalarıyla karşılaşacağız. Yani SID II'nin başlangıç adresi SID I'e göre 32 byte ilerde olacaktır. Programlarımızda kullanacağımız adresler ise SID I için tanıdık olan $D400 (54272), SID II için ise $D420 (54304)'tür. Aynı numaralı hafızaların fonksiyonu her iki SID için aynıdır; yalnızca her birinin arasında 32 byte'lık bir fark vardır. Yazımızın hemen bitişiğinde bulunan ufak bir test programı STEREO SID'in tüm fonksiyonlarını yapıp yapmadığını kontrol etmenize yarayacaktır. Bu programda da göreceğimiz gibi Streo SID'in programlaması getirdiği yeni olanaklara kıyasla oldukça basittir.

Gelelim STEREO SID'in montajını nasıl gerçekleştireceğimize...

Devrenin şemasında görüldüğü gibi iki SID (MOS 6581) çipi kullanılmıştır; bunlar SID I (IC 1) ve SID II (IC 2)'dir. Dikkati çeken başka bir nokta da bunların birçok bağlantıları ortak kurulmuştur. Fakat gene de farklı olan bağlantılar var. Özellikle dört tanesiyle başlayalım; bunlar CAP 1A, CAP 1B, CAP 2A ve CAP 28'dir. IC2'nin bu bacaklarına C1 ve C2 kondensatörleri bağlanmıştır. IC1'de bu bağlantıya gerek yoktur, çünkü bu kapasitörler bilgisayarların ana plaketinin üzerinde vardır; 1'den 4'e kadar olan kontak uçları bu bağlantıları bilgisayarın platekine iletmektedir.

IC2'nin PotX ve PotY girişlerini ayrıca iki analog dijital çevrimi için kullanabilirsiniz. Devremizin bilgisayar plaketi ile olan 28 bacaklık bağlantısı tümüyle dolu olduğu için bu girişler devrenin üzerinde bırakılmış ve kullanıcının isteği doğrultusunda bir bağlantı yapması sağlanmıştır. Yerleştirme planımızda bu girişler ayrıca işaretlenip belirtilmiştir.

Gelelim her iki SID'in ayırtedilmesine ve tek tek program tarafından seçilmesi işlemine. Daha evvel SID'in seçilmesi için kullanılan CS sinyali (8 numaralı bacak) burada gene kullanılacaktır. Fakat bu sefer bu bölgede tek SID değil iki SID adreslenecektir. Adresleme aralığını 32 byte olarak belirlemiştik; 32 rakamı ikinin 5'inci kuvveti olduğuna göre A5 sinyalini de her iki SID arasında seçim yapmak için kullanacağız. Her iki SID'in doğru olarak seçilmesinde gerekli olan bu karışık sinyallerin üretilmesini IC3 üstlenmiştir. SN74LS138N tipindeki bu çipin asıl görevi 3'ten 8'e Decoder/Multiplexer işlemleridir. Adı bu kadar kompleks olsa da yaptığı iş öyle değildir. Olayı daha iyi belirleyebilmek için bu elemanın iç yapısını verdik. Adres sinyallerinden oluşmuş 3 bitlik bilgi bu elemana girer ve bu bilginin gösterdiği (0 ile 7 arası) numaralı Q çıkışı şaseye çekilir. Şemada görüldüğü üzere A1 ve A2 zaten şaseye çekilmiş durumdalar; açıkça görülmektedir ki A0 (yalnızca) Q0 ve Q1 çıkışlarını etkileyecektir. Bunlardan başka çipin çıkış sinyali üretip üretmemesini belirleyen üç giriş vardır. Bunlardan E1 ve E2 şase seviyesinde (low activ), E3 ise şase seviyesinde olmadığında çıkışta sözü edilen sinyaller oluşur. Şimdi şemamızı daha dikkatli inceleyelim: Chip-Select sinyali E1'e şase seviyesinde ulaştığı zaman A5'in de yardımıyla SID'lerden birinin seçilmesi gerekiyor. A5 sıfır seviyesindeyken IC1 seçilir, bir durumunda ise IC2 aktif olur.

Şu anda kesin olarak her bir SID'in nasıl aktive edildiğini öğrenmiş bulunuyoruz. Fakat bilmediğimiz bir şey var; ses bilgisi stereo kuvvetlendiricimize (müzik seti veya başka biramplifikatör katı)

nasıl ulaşacak? Şayet herhangi bir şekilde Commodore 64'ün devresini görmüşseniz SID'in ufak bir ara kuvvetlendirici kat üzerinden AUDIO/VIDEO çıkışına bağlı olduğu dikkatinizi çekecektir. Benzeri bir ara kuvvetlendirici katı biz de ikinci ses kanalımız için monte etmeliyiz. Zaten şemada görülen transistör ve çevresindeki birkaç eleman sözünü ettiğimiz ara kuvvetlendirici katı oluşturmaktadır. Bu katın bilgisayar plaketi ile bağlantısı bir elektrolitik kondensatör üzerinden AUDIO IN sinyali için bulunan hattı kullanarak yapılıyor. Tabiî böyle bir durumda bilgisayara ekstra ses girişi, bilinen AUDIO/VIDEO port'u üzerinden yapılamıyor, fakat normalde çok az kullanım alanı bulan bu girişi daha yüce bir amaç uğruna kurban etmekte bir sakınca görmedik. İşimiz henüz burada bitmiyor; bilgisayarın plaketi üzerinde bulunan C12 kondensatörünü bir telle kısa devre edip lehimlerseniz, artık SID II'nin kuvvetlendirilmiş ses çıkışı doğrudan AUDIO/VIDEO port'unun 5 numaralı pin'ine ulaşacaktır. Bu arada çoğunuzun farketmiş olabileceği bir olay AUDIO/VIDEO port'unun giriş ve çıkışlarında eskisine göre bazı farklılıklar olacağı. AUDIO/VIDEO port'unun yeni şekli yazımızla birlikte verilmiştir. Şekil port'a dıştan bakıldığında görüldüğü gibi verilmiştir.

Bağlantı kablosu olarak kesinlikle standart 5'li ara kablo kullanmayın!!! Çünkü böyle bir durumda müzik setinize ses bilgileri dışında görüntü ve renk bilgileri de taşımış olursunuz. Böyle bir durumda hangi aletin daha önce zarar göreceğini kesin olarak kestirmek güçse de, hiç hoş olmayan bir durum doğacağı aşikar. Bu yüzden, orada yalnızca ses sinyallerini taşıyacak bir ara kablo yapmanız veya yaptırmanız gerekmektedir.

Plaketimiz tek taraflı olduğu için çok büyük karışıklıklar içermiyor. Gerekli malzemelerin hepsinin adı bu yazının sonunda yer alan malzeme listesinde var. Plaketi kendisi yapacak olanlara kolaylık olması bakımından ters basılmış olarak verdik.

Plaketi oluşturduktan sonra 0.8 mm'lik matkapla delikleri delin. Malzemeleri yerleştirirken dikkat etmeniz gereken bir husus da IC2'nin altındaki tel köprünün IC'nin soketinden daha evvel lehimlenmesi gerektiği.

A5 sinyali bizim devremize basit olarak ulaşamadığı için bu sinyali bilgisayar plaketinin üzerinde bulup bir tel yardımıyla devremize taşımalıyız. Bu sinyalin nerede bulunacağı resimde gösterilmiştir.

Şayet bilgisayarınızda SID soketli değil ise işiniz biraz daha zorlaşacak demektir. SID'i yerinden sökmek için ince uçlu düşük güçte bir havyaya ve lehim pompasına ihtiyacınız olacak demektir. Tabiî bu aletleri kullanabilmek oldukça hassas eller ve beceri istiyor, yoksa bilgisayarınızın plaketini kavurup bütün aleti tanınmayacak hale sokmanız işten bile değil. Bütün bacakların lehimlerini temizledikten sonra (ki buna yüzde yüz emin olun yoksa bilgisayarın plaketini kırarsınız!!!) ince uçlu bir tornavidayı SID'in altına sokun ve kaldıraç usulü kanırtarak çipi sökün. Aslında bu işi tecrübeli ve bilgili birine yaptırmanızı tavsiye ederiz. Başka bir tavsiyemiz ise söktüğünüz yere takacağınız soketi kaliteli tipte seçmeniz. Yarın öbürgün başka bir uygulama için bilgisayarınızın plaketine el attığınızda bu soket size sorun olmasın.

STEREO SID devresini bu sokete monte edebilmek için devremizden 28 tane (2 x 14) uç çıkmaktadır. Bu uçlar devremizin lehim tarafından çıkacaktır. Piyasada bu amaca hizmet edebilecek çeşitli uçlar bulunabileceğinden kesin bir sınırlama koymuyoruz. Dikkat edeceğiniz hususlar uçların sokete girmeyecek kadar kalın olması ve ucuz soketlerin bu takma işlemi sırasında hemen deforme olup temassızlıklar yaratıp baş ağrılarına neden olmaları. Bu konularda ne kadar doğru seçim yapabilirseniz devrenizi o kadar çabuk ve rahat monte edip çalıştırabilirsiniz.

Şimdi ilginç bir sorunumuz var: SID çipini nereden bulacağız? Bu sorunu en yakın TELETEKNİK bayisine danışın veya doğrudan TELETEKNİK'e gidin. Benim bildiğim, genelde yedek parça olarak stoklarında belli bir miktar bulundurdukları, fakat olur ya siz gittiğinizde stok geçici olarak tükenir. Gene de çok zorlanacağınızı tahmin etmiyorum.

Bir de değişik SID'ler hakkında bilgi verelim. Tüm SID'ler aynı çalışmıyor; mesela eski üretim SID'lerde filtre yapısı daha farklı ve gürültü oranı daha yüksek; bu yüzden aynı müzik parçası değişik SID'ler tarafından (çok çok küçük de olsa) farklı çalınabilir. En son tiplerinde (6581 yazısının hemen arkasında R3 yazıyor) bütün sorunların giderilmiş olduğunu söyleyelim.

STEREO SID'in fonksiyonlarını denetlemek için verdiğimiz test programını kullanabilirsiniz. Bu programı yazıp çalıştırdıktan sonra her iki kanaldan üç değişik akor çıkacaktır ve sizin space tuşuna basmanızı bekleyecektir.

Burada verdiğimiz uygulama örneğiyle Commodore 64'ten nasıl bir Stereo Sesli Bilgisayar elde edebileceğinizi gösterdik. Geri kalanı ise sizin yaratıcılığınıza bırakıyoruz. Denenecek o kadar çok şey var ki. Deneyin ve hepsini bulmaya çalışın. Mesela çılgınca bir müziği tam stereo ayrımıyla programlayın ve bunu 2x200W'lık müzik setinde çalıp yaratıcılığınızın tüm zevkine varıp, HALL ve ECHO efektleri elde etmeye çalışın. Veya mesela bir sesi bir kanaldan başlatın ve öteki kanala doğru yollayın. Yaptığınız en güzel STEREO müzik parçalarını bize yollayın, merakla bekliyoruz. □

**MMALZEME LİSTESİ**

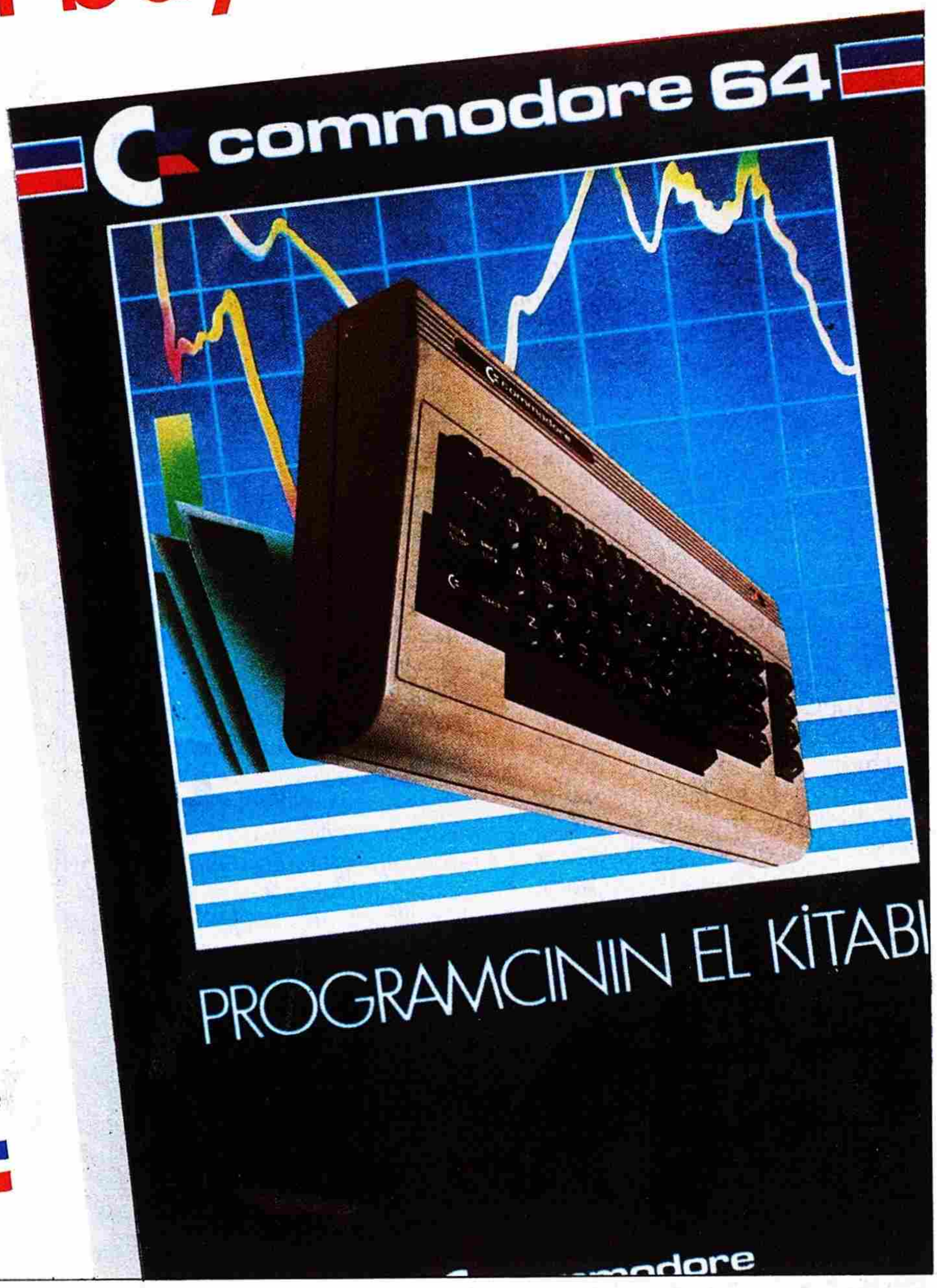
**Yarı iletken:**
2 6581 SID, IC1, IC2
1 SN74LS138N; IC3
1 BC 338; T1

**Kondensatörler:**
2 470 pF; C1, C2
1 1 nF; C3
1 10 mikrofarad/25V (Elko); C4

**Rezistanslar:**
2 1 kOhm; R1, R3
1 10 kOhm; R2

**Diğer:**
3 IC soketi (28 bacak için)
1 IC soketi (16 bacak için)
2 grup uzatma telleri (141'i ve entegre devre ile aynı bacak aralıklı).

# Commodore ile programlamada en büyük yardımcı




Teleteknik
ve tüm
bayilerde

# Harika CHIP'i Yaratmak

Hepimizin bildiği gibi, bilgisayarın kalbi -ve tabiî beyni- chipleridir. Chipler bu özel ve önemli konumları nedeni ile, pek çok kötü şakaya konu olmuşlardır.

Bir chip dışarıdan sevimli görünse de, pek heyecan verici değildir. Bu koyu renkli, dikdörtgen kutunun her iki yanında yaklaşık 40 metal bacak sıralanır. En basit tanımı ile bu bacaklar, chipin içindeki silikon tabakanın dışarısı ile olan bağlantılarıdır. Bu silikon tabaka, yaklaşık 5 mm² alanında ve 1 mm kalınlığındadır. Bu küçük boyutlarına rağmen bu tabakaya, 50.000'den fazla eleman sığdırılabilmektedir.

Öyle bir bilgisayar hayal edin ki, bu bilgisayarı oluşturan elemanları -örneğin transistörleri- ucuz bir radyonun içinde bile bulma olanağınız olsun. Gerçekten de bu fikir, bilgisayarın tarih-öncesinde -yaklaşık 25 yıl öncesi- bilgisayarların nasıl imal edildiklerinin ve neden o hantal donanımları ile bir odayı rahatça doldurduklarının en açık yanıtıdır. Bu arada şunu belirtmekte fayda var. O büyük donanımların çoğunu teller ve kaplamalar oluşturuyordu. Transistörlerin bu elektronik ve mekanik yığına olan katkısı çok küçüktü. Zaten transistörlerin hiçbir zaman büyük olmaları gerekmemiştir. Çünkü bunlar sadece bilgisayarların kullandığı sistemin 1 ve 0 durumlarını, elektrik dilinde temsil eden açık ve kapalı sinyallerini oluşturmak için, birer anahtar olarak kullanılmışlardır.

Sıra, chipinizin tarifine geldi. Önce chip için gerekli olan tüm elemanları, silikon dilim üzerinde önemsiz pürüzler sayılabilecek kadar küçük boyutlara indirin. Sonra bu elemanları bir dilim silikon üzerine sürüp fırına atın ve erimeye bırakın. Fırından çıkardığınız anda, artık elinizde tüm elemanları yerlerine yerleşmiş olan bir dilim silikon olacak: Daha doğrusu bir chip.

Doğal olarak günümüzün modern chiplerini üreten teknikler, bu basit tariften oldukça farklıdırlar. Fakat tüm farklılıklara rağmen, bu gelişkin chipler de tamamen, birçok mikroskobik parça içeren bir silikon tabakası fikri üzerine tasarımlanırlar.

Bu nedenle chipler ve radyolarda kullanılan transistörler, aslında farklı tatlara sahip silikon sandviçlerdir. Bir chipin yapımına, temel




olarak alınan bir silikon dilimle başlanır. Daha sonra bu dilime, birbiri arkasına yeni silikon tabakalar eklenir. Bu tabakalara farklı tatlar vermek için, içlerine çok az miktarlarda diğer bazı elementlerden de katılır. En sonunda tüm bu farklı tabakalar, özel teknikler kullanılarak hep birlikte eritilirler. Artık sandviçiniz, pardon, chipiniz hazırdır.

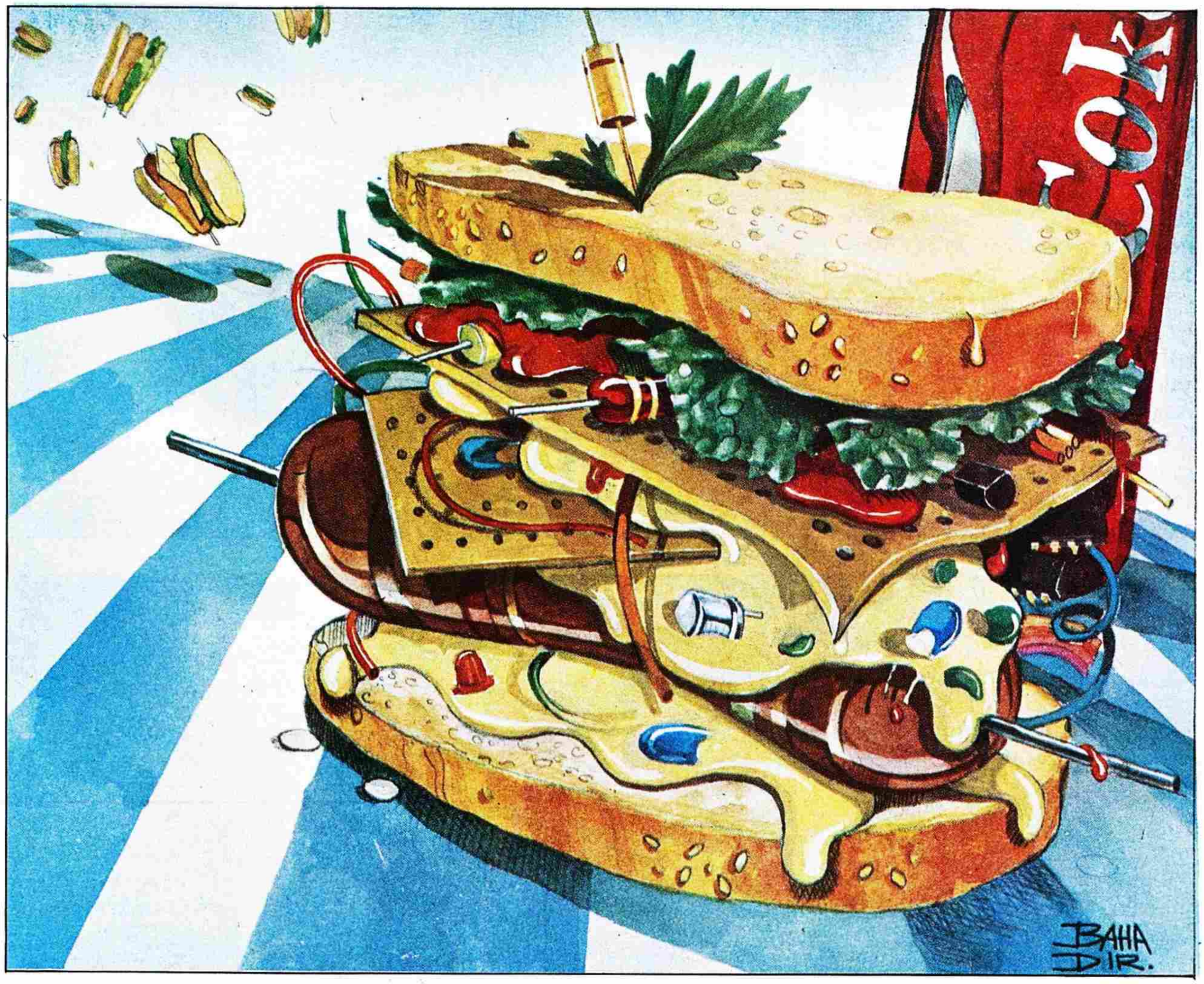
Özel transistörler, bu chip sandviçten bazı veya tüm dilimlerin çıkarılması ile elde edilirler. Bir transistör, temel dilimin üzerinde yükselen bir silikon adaya benzer. Mikroskobik bağlantı köprüleri bu adaların aralarında kurulurlar. Bu köprüler, elektronik devrelerdeki transistörleri birbirlerine bağlamak için kullanılırlar. Bu adaları birleştirirken kullanılan yol, elde edilecek olan chipin tipini belirler. Bu basit teknoloji, tüm chipler için geçerlidir.

Bu adaları daha küçültmek için kullanılan teknikler geliştikçe, chipler de daha kuvvetli hale gelmektedirler. Bir diğer dizayn problemi ise, tüm bu adaların bir chip üzerine -bir köprü diğerini kesmeden- nasıl yerleştirileceğidir.

Bu işi 50.000 eleman ile başarmak güçlü bir bilgisayar, gelişmiş bir grafik programı ve bol zaman gerektirir. Genellikle mesai saatinin bitiminde, chipin dizaynı büyük bir kâğıt üzerine, en ince detayına kadar geçirilmiş olur. Bu çizim daha sonra fotoğrafik yöntemlerle, bir chipin boyutlarına indirgenir. Artık bu çizimi, daha önceden bir silikon sandviçinin yüzeyine yerleştirilmiş bir fotoğrafik filmin üzerine oturtarak, transistör haline gelen adacıkları oluşturmak mümkündür.

Chipler iki ana gruba ayrılırlar: İşlemciler ve bellekler. İşlemci, bilgisayarın koşturup duran şefidir. Hesap işlemlerinin tümünü işlemci yapar, aynı zamanda her şeyi kontrol altında tutar. İşlemci -normal olarak- bir ana hesap makinesine ve çeşitli depo alanlarına sahiptir. Bu depo alanları veriyi, işleninceye kadar olduğu gibi saklarlar, ayrıca hesaplamaların sonuçlarını da oldukları gibi not ederler. Yani sonucun sıfır veya negatif olması farketmez, her iki durumda da sonuç kayıt olarak tutulur. Bunun yanında, bir sonraki verinin veya komutun nereden geleceğini gösteren kayıtlar da bu depo alanlarda saklanırlar.

Basit bir ifade ile bellek chipleri, çok fazla sayıdaki transistör dizileridirler. Bunlardan her biri diğerlerinden ayrı olarak adreslenebilir ve bir adet 0 veya 1 depolayabilir. Bu nedenle, örneğin 16 K'lık bir belleğe sahip bir chip, her biri diğerlerinden ayrı olarak se-




çilebilen, daha sonra da okunan veya değiştirilebilen, 16.000'den fazla farklı cihaza sahip demektir. Fakat bu geniş kullanım alanı ve esnekliği, büyük bir sorunu da beraberinde getirir. Bu, chipin içindeki bağlantı problemidir. Açıkçası tüm hatları gerekli yerlere düzgün olarak ulaştırıp, araları kontrol altında tutmak işin en zor yanıdır.

RAM ve işlemci chiplerle ilgili bir diğer sorun da sıcaklıktır. Chip devamlı kendisini besleyen elektrik yükünün altında çalışır. Sıfır'ı bir veya bir'i sıfır yapmak için kullanılan enerji, sonuçta sıcaklık olarak ortaya çıkar. Bu dönüşüm tüm bilgisayar tarihi boyunca chipin boyutları hakkında bir sorun olarak ortaya çıkmıştır. Öyle ki teorik olarak, kulanıldıkları anda kendi kendilerini eritebilecek küçüklükte chipler yapmak mümkündür.

ROM chipler sıcaklık problemlerinden bu kadar çok etkilenmezler. Çünkü üzerlerinde devamlı bir elektrik yükü taşımazlar. Bunun nedeni, değiştirilemez oluşlarıdır. Böylece bu yalnız-okunur bellekleri aktif tutmak için, üzerlerinde elektrik akımı dolaştırmaya gerek kalmaz.

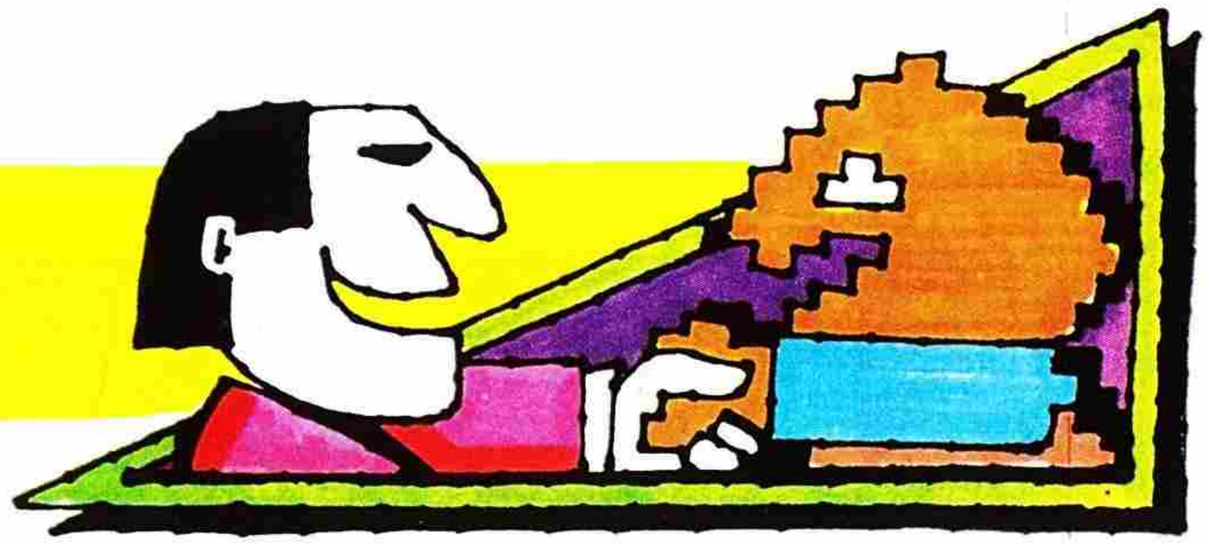
ROM'ların tek farklı tipi vardır ve maalesef bu tip yeni bilgisayarlarda yaygın olarak kullanılmaya başlanmıştır. Bu; PRom, başka bir deyişle programlanabilir yalnız-okunur bellektir. Bu belleklerde bazı elemanların değerlerini silebilir ve yerine istediğiniz değerleri yazabilirsiniz. Doğal olarak bu işlem yeni bir chip dizayn etmekten çok daha kolay ve hızlıdır. Bu nedenle üreticiler bu tip chipler kullanan bilgisayarların üretimine yönelmişlerdir. Sinclair QL gibi.

Son olarak genel amaçlı bir chip olan ULA'dan söz edeceğim. Uzuncası Uncommited Logic Array, (İşlenmemiş Mantık Dizisi) olan bu chip, aynı PRom gibi, bilgisayar üretimcisine kesin bir serbestlik verir.

Bu serbestlik üretimciye chipi, chipin -dolayısı ile kendisinin- gereksinimleri için yeniden programlama olanağını verir. Bunun yanında ULA PRom'un tersine, para tasarrufunda ve kullanımı alanında tatminkâr bir yolu temsil eder. Bu yol, geleceğe uzanan kuşağın müjdecisidir. Günümüzün en gelişmiş chiplerinden bazıları, hem bellekleri, hem de işlemcileri olan çok amaçlı kompleks cihazlardır. Başka bir deyişle tamamen birer bilgisayar konumundadırlar. □

**Çev.**|Cenk EDEN

# TANITIM



## Paperboy

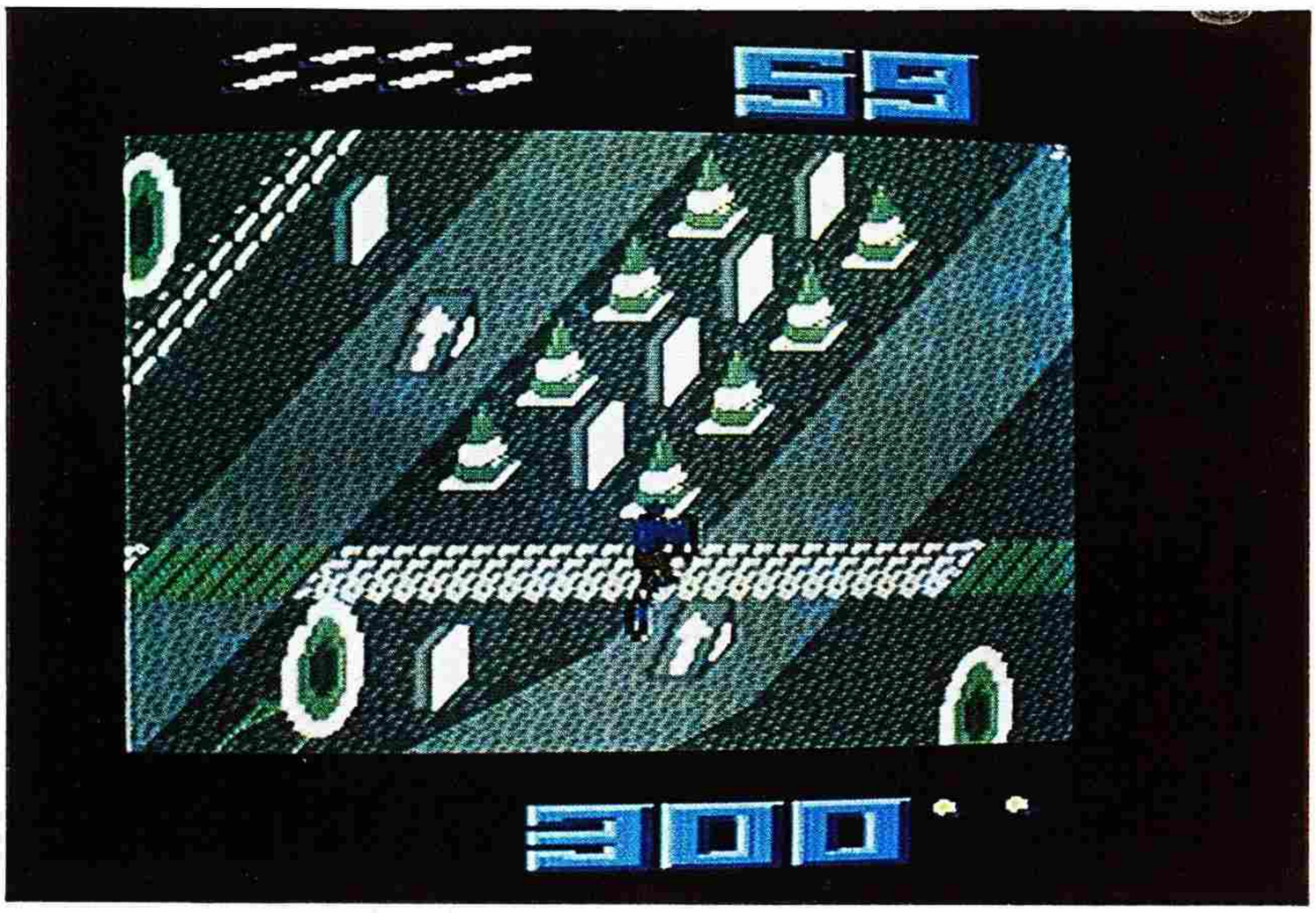
Hiç gazete dağıtıp dağıtmadığınızı bilmiyorum. Eğer dağıtmadıysanız bu oyunu oynadıktan sonra buna çok memnun kalacaksınız. Elite firmasının uzun zamandır beklenen bu oyunu nihayet ülkemize de geldi.

İşiniz bir Amerikan orta sınıf mahallesinde abonelerinize gazete dağıtmak. Yol, son zamanlarda moda olan korku filmlerindeki yollara benziyor. Böyle bir yerde gazete dağıtmaya kararlı iseniz oyuna başlayabilirsiniz.

Oyuna oldukça çirkin bir pazartesi günü başlıyorsunuz. Tehlikeli bölgelerden uzak durup gazeteleri abonelerinizin posta kutularına bırakmanız gerekiyor. Oyunun başında kendinizi ekranın altında bisikletinizin pedalına basarken buluyorsunuz. Bisikletinizi kontrol etmeniz oldukça kolay. Karmaşık joystick hareketleri gerekmiyor. Hızlanmak için joystick'i ileri itmeniz, yavaşlamak için joystick'i geri çekmeniz yeterli. Gazeteleri bırakmak için "ateş" tuşuna basmanız gerekiyor. Gerisi sizin yeteneğinize ve zamanlamanıza kalmış bir şey.

Abone leriniz sarı boyalı evlerde yaşıyorlar ve üstünde 'SUN' (Daily Sun) yazan posta kutuları var. Gazeteyi bırakmak için bisikletten inmeniz gerekmiyor. Evlerin yanından geçerken 'ateş' tuşuna şöyle bir basmanız yeterli. Gazeteyi mutlaka posta kutusuna isabet ettirmeniz gerekmiyor, kapıya gelse de yeterli. Ama is-

Commodore

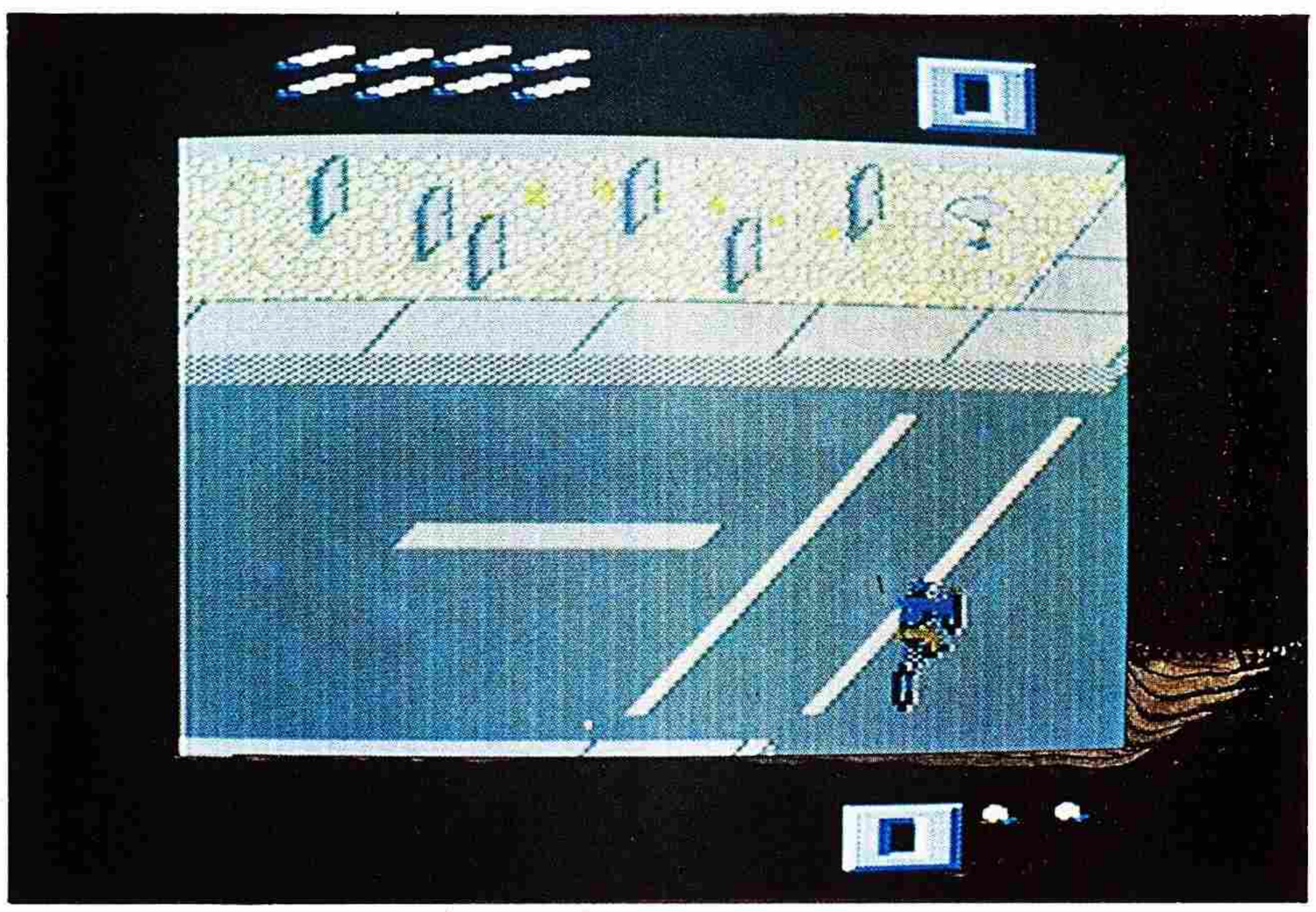



muhtemelen diğer gazeteci çocukların. Çünkü işiniz oldukça tehlikeli. Peşinize bekçi köpekleri düşüyor, önünüze yuvarlanan tekerlekler çıkıyor, uzaktan kumandalı oyuncaklar, tornetler, arabalar yolunuzu kesiyor. Bütün bunların yanısıra kanalizasyon kapaklarının üstünden de geçmemeniz gerekiyor. Bunlar yetmiyorsa yolda dolaşan çılgın bisikletliler var. En kötüsü, Cumartesi günü ölümün en son evden çıkması. Doğrudan size hamle yapacak üstelik.

Bütün bunlar için bir sakinleştiriciye ve ses geçirmeyen bir odaya ihtiyacınız olabilir.




kalarsanız camları indiriyorsunuz. Bu da aboneliğinizin iptaline yolaçıyor. Dağıtım yaptığınız evlerin tüm camlarını kırarsanız ya da kuş banyolarına gazete atarsanız banus puan da alabiliyorsunuz.






Çevredeki mezar taşları

Ama dağıtımcı patronunuz eğitim pistinde yaptığınız çalışmayla yetinmeyip özel antrenmanlar da hazırlamakta. Engeller üzerinde pazarlık edin ve hedefleri vurarak bonus puan alın.

Eğer hafta sonuna kadar başarılı bir çalışma yürütürseniz, hafta sonu içinde tebriklerimizle yazan bir gazete alacaksınız. Bu, bütün abonelerinizi koruduğunuz ve işinizi kaybetmediğiniz anlamını taşıyor. Eğer abonelerinizi kaybettiyseniz, gazetede kovuldunuz yazacak. Buna şükredip o akşam yatın. □

# Flight Simülatör II




Oyunumuzun en ilginç özelliklerinden birisi birçok bilgisayara adapte edilmiş olmasıdır. Bu simülasyonda kullanılacak uçağın kumanda edebileceğimiz 47 fonksiyonu var. Bunlar bize uçağı penceresinden üç boyutlu ve renkli bir görüş sunuyorlar.

PIPER PA-28-181 ARCHER II tipi tek motorlu bir uçak kullanılan Model hızı 148 mph. Bu uçağın seçilme sebebi kullanılma kolaylığı ve buna rağmen uçağın kullanıcısına verdiği büyük performanstır.

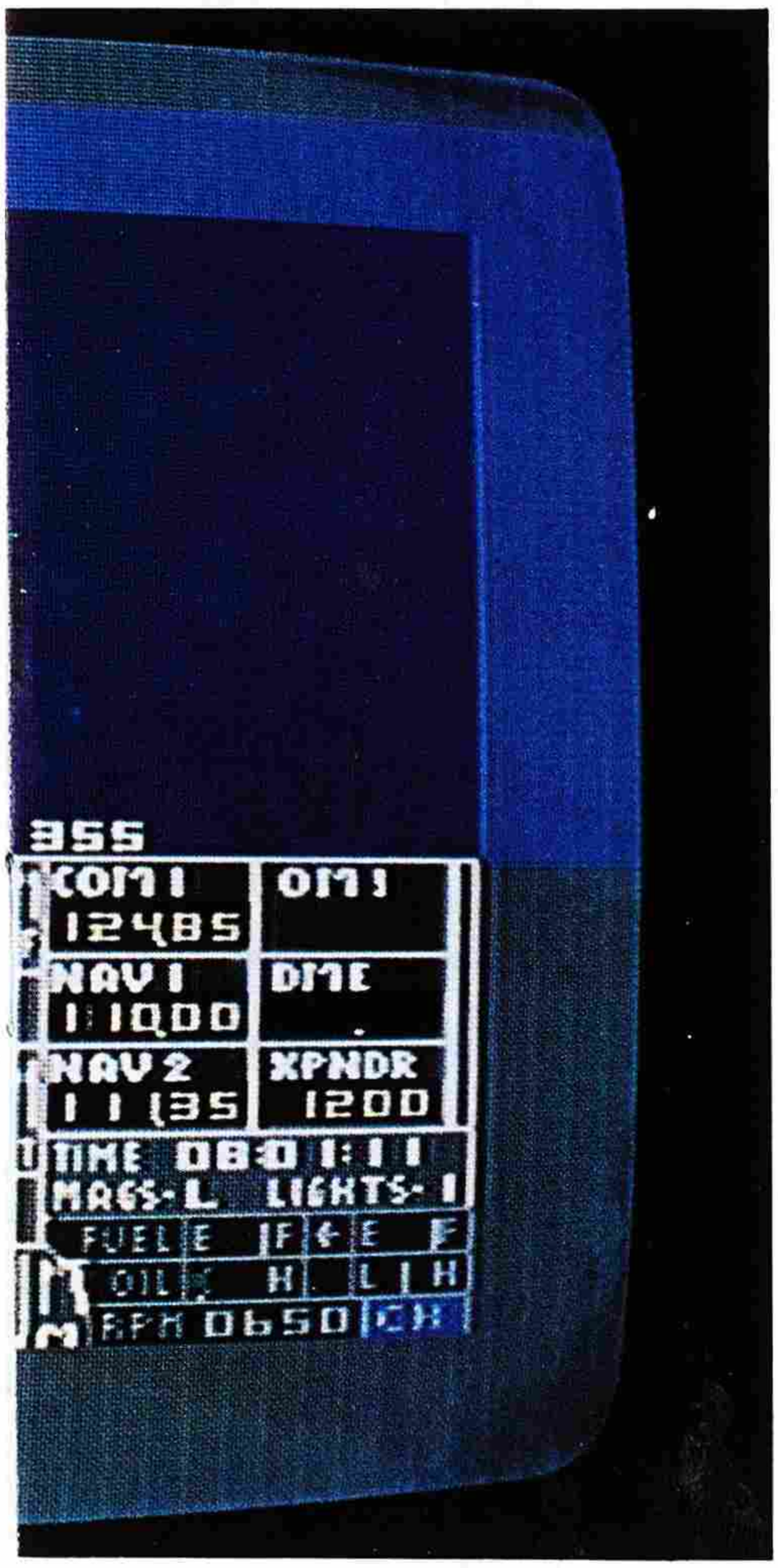
Oyun, uçuş simülasyon kısmından ayrı olarak, bir de 1917'lerde geçen bir savaş oyunu içeriyor. Bu oyunda uçak bomba, makineli tüfek ve radarla donatılıyor. Kendinizi bilgisayar tarafından yönetilmekte olan düşman uçaklarına karşı savunacaksınız.

Gösterimde bütün Amerika Birleşik Devletleri ve burada yeralmakta olan bellibaşlı 80'den fazla havaalanı yeralmaktadır. Uçuş faktörlerine gerçek zaman, bulutlar, gün ışığında, şafak sökerken veya gece uçuşu eklenmiştir.

İsterseniz kolay modda, fazla başınızı ağırtmadan uçabilirsiniz. Kendinize güveniyorsanız gerçek uçak modunda uçabilirsiniz, fakat burada uçabilmek için gerçek anlamda pilot olmak gerekmektedir. Bu tür seçenekleri seçebilmek için 'E' tuşuna basarak parametre ayarlama yerine girebilirsiniz.

Uçağımızın içine gelince ekran başlıca iki bölüme ayrılmış durumdadır. Üst bölümde dışarının görüntüsü, alt bölümde ise uçakta bulunan aletlerin görüntüsü. Alt kısmı iki ana kısma ayırabiliriz. İlk kısım uça-




ğın solunda yeralan 6 göstergeden oluşmaktadır. Bunlardan en sol üstteki uçağın hızı-



nı gösterir. Bunun altında uçağın yatay konumu, yanda ise uçağın yönünü gösteren pusula yeralır. Pusulanın üstünde uçağın dikey durumunu bildiren gösterge, bunun yanında uçağın yüksekliğini gösteren altimetre yeralmaktadır. Altimetrenin altında uçağın dikey hızı (yükselme veya alçalma) gösterilmektedir. İlk kısmı oluşturan göstergelerin sağındaki iki gösterge, radar kullanılarak uçulduğunda hava alanını izlemeye yarar. Ekranın solundaki bölümde çeşitli aletler yeralmaktadırlar.

Üstünde COM yazan cihaz haberleşme radyosudur. Üzerinde DME yazan cihaz ise uçağın deniz mili cinsinden uzaklığını göstermektedir. TIME göstergesi ile 24 saat üzerinden gerçek zamanlı saat gösterilmektedir. LGTS ile uçaktaki ışıkların açık/kapalı durumu gösterilmektedir. FUEL ile uçağın iki yakıt tankının durumu gösterilmektedir. REM ile motorun devir sayısı, bunun tam altında da kabın ısıtıcısının açık/kapalı durumu gösterilmektedir.

Simülasyonda kullanılacak tuşlar:

## Uçuş Kontrolleri

T Kumanda kolu aşağı
B Kumanda kolu yukarı
F Kumanda kolu sola
G Kumando kolu ortada
/ Tam gaz
[ Gaz kesme
] Gaz verme

boşluk çubuğu frenler

ctr1 I Kabın ısısını açma kapama
ctr1 Fsağ yakıt tankı
ctr1 Fsol yakıt tankı

## Görüntü Kontrolleri

5 Görüntü seçmek için ilk nu tuşa basınız.
T Önden görünüş
F Soldan görünüş
H Sağdan görünüş
B Arkadan görünüş
G Aşağının görünüşü
R Ön soldan görünüş
Y Ön sağdan görünüş
V Arka soldan görünüş
N Arka sağdan görünüş
4 Radar görüntüsü
) 4'e bastıktan sonra radar görüntüsünü küçültür.
( 4'e bastıktan sonra radar görüntüsünü büyültür.

## Çeşitli Kontroller

ctr1 L Işıkları açma
ctr1 B Altimetre ayarlama
ctr1 D Pusula ayarlama
E Parametre menüsünü çağırma
P Oyunu dondurma / başlatma

# WAR GAMES

BAHADIR İŞLER

BOMBALAR

VURDUK VURDUK

BABAA!

İŞTE, BİZİ SAVAŞTIRAN...

BABAA!

ÇOK YALNIZIM

ORADA CAMIN ARKASINDA.

ONU YOK EDELİM!

N'OLDU BUNA

CRASH

NE!

ONLARI YOK ETMELİ.

YEEaah!

TAK

SIIIV

Hia!

SIIIV

ŞİMDİ, YOK ETME SIRASI BİZDE

AMA BEN...

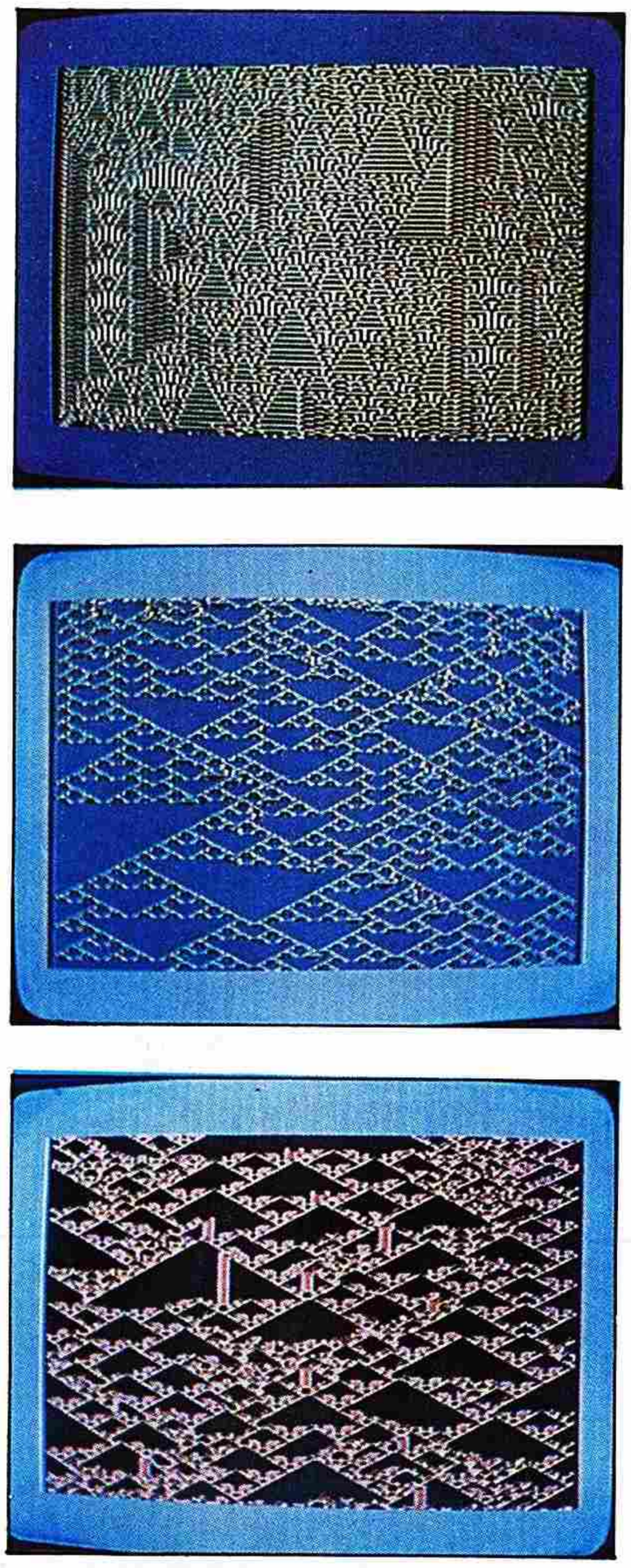
SIIIV

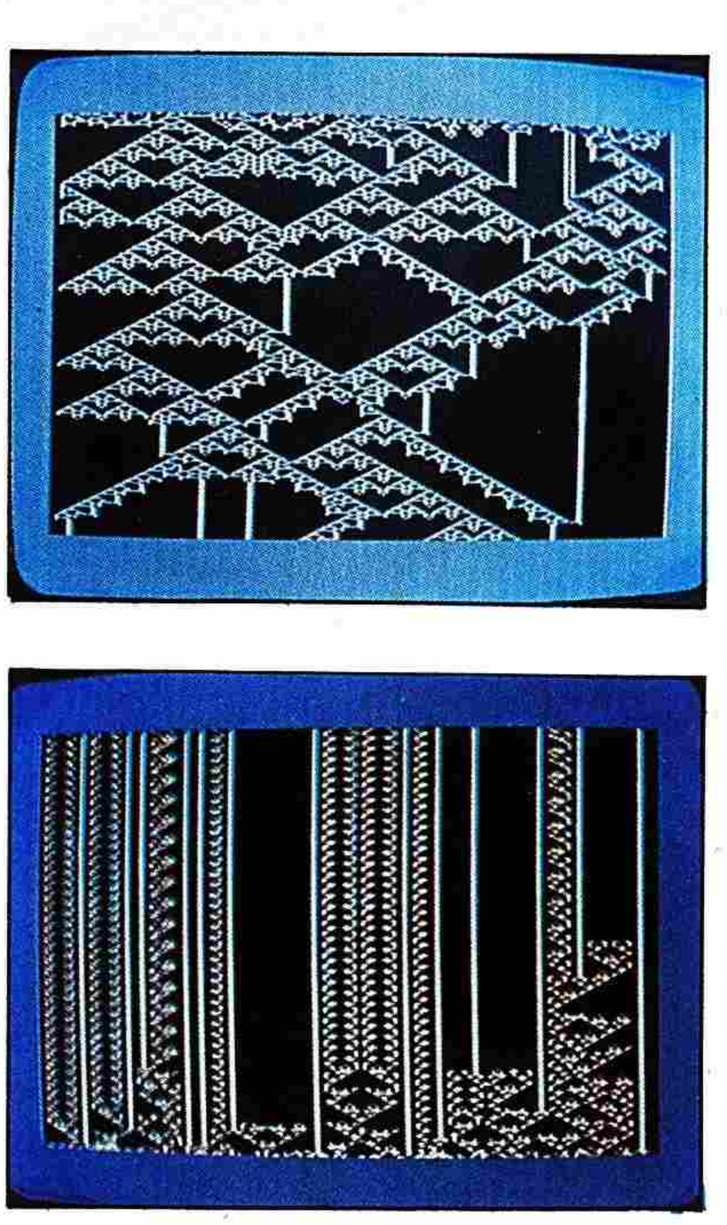
GAME OVER!

AAH!

BAHA BIR-

SON









# Renkli Yaşam

Tuna ERTEMALP

Onuncu sayımızda bir program yayınlamıştık. Adı LIFE GAME idi. Bir yaşam oyunuydu bu. Bir alandaki canlıların, gelişen nesiller boyu geçirdikleri evrim, matematiksel olarak ortaya konuyordu. Şimdi, bu ölüm kalım savaşının biraz daha karmaşık, çok daha renkli ve inanılmayacak kadar ilginç bir başka türünü sunuyoruz sizlere.

En temel fark, eski oyun dikdörtgen bir ALAN üzerindeyken, bu oyun bir ÇİZGİ üstünde gerçekleşiyor. Bu çizgiyi yanyana konmuş yaşam hücreleri olarak düşünüyoruz. Her hücrede 0, 1, 2 veya 3 tane canlı var. Sadece üçe kadar canlı kullanmamızın nedeni, her canlı sayısı için farklı bir renk kullanmamız ve C-64'ün yüksek çözünürlüklü çokrenkli grafikte dört değişik renk kullanabilmesi. Eğer grafik özellikleri AMİGA gibi olan bir aletimiz olsaydı, bir hücreye 0-31 arası sayıda canlı koyabilirdik. Gene aynı nedenle, çizgimiz üstünde yanyana 160 hücre olduğunu düşünüyoruz. Çünkü çokrenkli grafikte X-ekseni 160 noktadan oluşuyor.

Oyunu başlatmadan önce, 160 hücrenin içine değişik sayılarda canlılar yerleştiriyoruz. Tabiî en az 0 en çok da 3 olmak üzere. Peki, her hücrenin bir sonraki nesilde içereceği canlı sayısı nasıl hesaplanacak?

Bunu, o hücrenin sağındaki ve solundaki hücrelerde bulunan toplam canlı sayısı ile kendi içinde o anda bulunan canlı sayısını toplayarak belirliyoruz. Dolayısıyla, elde edeceğimiz toplam 0-9 arası bir sayı olacaktır. Ama her hücrede en fazla 3 canlı olabileceğini söylemiştik?!

İşte bu nedenle, bir de kural tanımlıyoruz. 10 elemandan oluşan bu kuralın elemanlarını sırayla 0, 1, ...9 olarak numaralıyoruz ve her elemanın içine 0-3 arası bir sayı koyuyoruz. Diyelim ki yukarıdaki toplamımız 7 çıktı. Bu durumda kuralımızın yedinci elemanına bakıyoruz. Diyelim ki içinde 3 var. Bu durumda toplamını gerçekleştirdiğimiz hücrenin içinde bir sonraki nesilde 3 canlı olacak. İsterseniz bir örnek görelim...

Örneğimiz için, sadece 5 hücre uzunluğunda bir yaşam çizgisini ele alalım ve içlerine de, tamamen tesadüfi olarak, şu sayıları yerleştirelim:

```
2 0 3 1 2
```

Çizginin geri kalan kısımlarının 0 canlı içeren hücreler olduğunu varsayalım. Şimdi de bir kural belirleyelim:

```
0 2 1 3 2 0 1 3 2 3
```

Bu kuralı sözle şu şekilde belirtebiliriz:

0) Eğer bir hücrenin toplamı 0 ise bir sonraki nesil için de 0 canlı olsun.

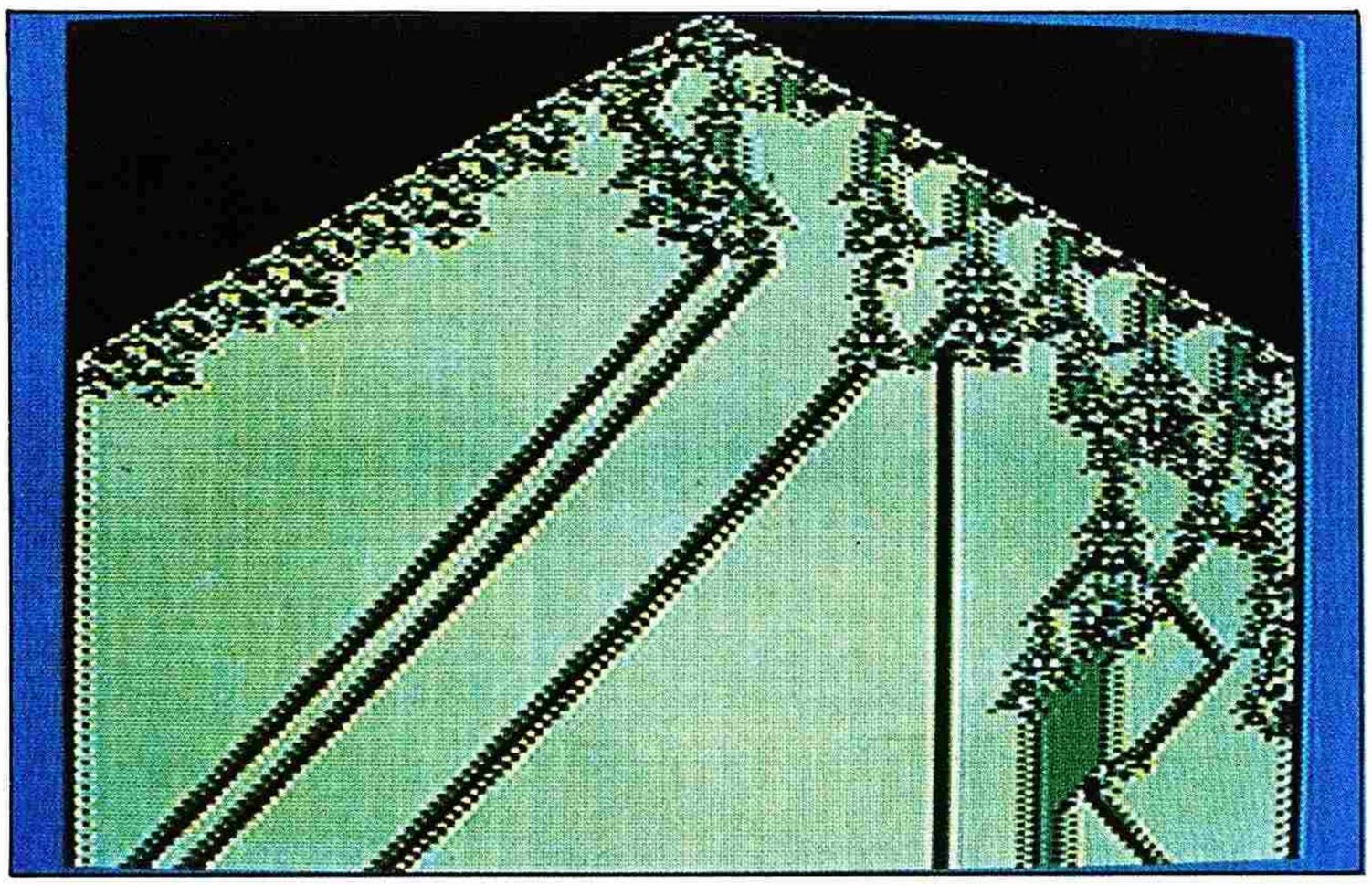
1) Eğer bir hücrenin toplamı 1 ise bir sonraki nesil için de 2 canlı olsun.

2) Eğer bir hücrenin toplamı 2 ise bir sonraki nesil için de 1 canlı olsun.

3) Eğer bir hücrenin toplamı 3 ise bir sonraki nesil için de 3 canlı olsun.

4) Eğer bir hücrenin toplamı 4 ise bir sonraki nesil için de 2 canlı olsun.

5) Eğer bir hücrenin toplamı 5 ise bir sonraki nesil için de 0 canlı olsun.



6) Eğer bir hücrenin toplamı 6 ise bir sonraki nesil için de 1 canlı olsun.

7) Eğer bir hücrenin toplamı 7 ise bir sonraki nesil için de 3 canlı olsun.

8) Eğer bir hücrenin toplamı 8 ise bir sonraki nesil için de 2 canlı olsun.

9) Eğer bir hücrenin toplamı 9 ise bir sonraki nesil için de 3 canlı olsun.

Şimdi bu beş hücrenin gelecek nesildeki durumunu gözönüne alalım.

İlk hücrenin içinde 2 canlı var. Solunda ve sağında ise 0 canlı. Toplamı 2 eder. İkinci kurala bakıyoruz, sonuç 1. Yani bir sonraki nesilde bu hücrede 1 canlı olacak. Aynı işlemi üçüncü hücre için yaparsak, 0+3+1=4 sonucunu ve dördüncü kuralda da 2 sayısını buluruz. Demek ki bu hücrenin canlı sayısı bir sonraki nesilde bir azalacakmış. Tüm çizgi için hesaplamayı yaparsak elde edeceğimiz yeni nesil şöyle olacaktır:

```
1 0 2 1 3
```

Aslında kolay, değil mi? Zaten tüm bu işlemleri bilgisayar yapacak.

Şimdiii... Eğer C-64'ün çok-renkli grafiğinde bize sunduğu 160*200 noktalık grafiği ve 4 renk özelliğini kullanırsak ne olur? Her nesil bir grafik satırı olarak gösterilebilir ve her değişik canlı sayısı değişik bir renk anlamına gelir. Böylece bir defada, 200 nesil boyunca sistemin geçirdiği değişimi bir anda görebiliriz. Ve tüm bu işlemin sonucunda, inanılmaz güzellikte görüntüler elde etmek mümkün.

Gelelim programımız hakkındaki teknik bilgilere (programın kendisini Döküm Eki'nde bulabilirsiniz). Programımızın ana iskeletini, MDE ile yazacağınız kısa bir makine dili programı oluşturuyor. Tüm çizim ve hesaplama işlemi bu program tarafından yapılıyor. Bu nedenle de, her ekran 4 sanayide hazırlanıyor. Eğer BASIC kullansaydık (hatta SIMONS' BASIC), bu süre 25-30 dakika olacaktı. İşte makine dilini öğrenmeniz için bir sebep daha!

Eğer BASIC kullansaydık (hattâ SIMONS' BASIC), bu süre 25-30 dakika olacaktı. İşte makine dilini öğrenmeniz için bir sebep daha!

Ancak bu rutini SYS 49152 ile çağırmadan önce bazı hazırlıklar yapmanız gerekiyor. Hücrelerin ilk nesildeki durumunu ve kullanılacak kuralı saptamak lazım. Bunun nasıl yapıldığını da örnek demo programlarını inceleyerek görelim.

YAŞAM DEMO 1 programı önce, POKE 56,32 ile kullanacağı hafıza sınırını aşağıya indiriyor. Bu çok gerekli, yoksa programınız ve değişkenler grafik tarafından silinebilir.

Sonraki işlem, yaşam çizgisinin ve kuralın hafızada yerleştirilecekleri adreslerin başlangıçlarının saptanması. Yaşam çizgisi, $5000 adresinden başlar ve bunu takip eden 159 byte boyunca devam eder. Makine dili programı, ilk nesille ilgili bilgiyi bu bölgeden okur.

Kural ise $4000 ve bunu takip eden 9 byte içinde yeralır. Bir kez daha tekrarlayalım, tüm bu byte'ların içine POKE ile 0-3 arası sayılar yerleştirilmelidir.

İlk satırın son komutu, bilgisayarın rasgele sayı üretimini yeniden başlatıyor, 20-23 arası satırlar, çizim renklerini ayarlıyor. Tesadüfi olarak birbirinden farklı üç renk seçiliyor. Dördüncü renk hep siyah olarak kalacak ve boş hücreler bu renkle çizilecektir.

25. satır, kuralın belirlendiği yer. Dikkat ederseniz, kuralın ilk elemanına 0 konuyor. Bu genelde daha iyi sonuçlar elde etmek için yapılıyor. İsterseniz değiştirip ortaya çıkan

sonuçlara bir bakın. Kuralın geri kalanı gene tesadüfi olarak belirleniyor. İsterseniz, YAŞAM DEMO 2'de olduğu gibi DATA satırlarından da okuyabilirsiniz.

30. satır ise yaşam çizgimizin ilk halini belirliyor. Gene dikkat ederseniz, en baştaki ve en sondaki hücrelere 0 konuyor, geri kalanları tesadüfen belirleniyor. Böylece, grafiğimizin sağ ve sol kenarlarında bir tür çit yaratmış oluyoruz. Siz isterseniz değiştirin. Örneğin, ikinci örnekte tüm çizgi sıfırlanıyor, sadece ortasına 2123 dizisi yerleştiriliyor. Sonuç oldukça ilginç.

Bu arada RND(8)*4 komutuna açıklık getirelim. Bu işlem 0-3.999999 arasında bir sayı yaratır. POKE komutu ile kullandığımız için, bu sayının tamsayı kısmı kullanılır, ki bu da bize 0-3 arası bir sayı verir. Bizim istediğimiz de buydu zaten.

40. satırda makine dili rutin çağırılıyor. Bu rutin, ekranı siliyor, grafik ekranına geçip multicolor'u açıyor, renkleri ayarlıyor ve çizimi yapıyor. Tüm programın SUPER ASSEMBLER dökümü açıklamalı olarak verilmiştir. Gene bu sayımızda verdiğimiz C-64 İLE GRAFİK yazısının ışığı altında inceleyip, istediğiniz değişiklikleri yapabilirsiniz.

50-60 arası satırlar, grafik çizimi bitince sizin bir tuşa basmanızı bekliyor. Eğer boşluk tuşuna basarsanız yeni bir grafik çiziliyor. Başka bir tuşa basarsanız 70-80 arası satırlarda grafik ekranı ve multicolor kapatalıp yazı ekranı normale döndürülüyor. Ve program bitiyor. Burada, isterseniz yeniden 40. satıra atlayarak, sistemin bir sonraki 200 nesilini de görebilirsiniz. Ama bunun için programa bir GOTO 40 eklemeniz lazım.

Dediğimiz gibi, bu örnek programlar sadece birer iskelet program. Üstlerde istediğiniz değişiklikleri yaparak yeni canlı türleri yaratabilirsiniz.

Son olarak matematikçi, bilgisayarcı ve ilginç şeylerden hoşlanan tüm arkadaşlara düşünmeleri için birkaç soru: İlk nesili, kuralı ve hattâ renkleri bile tamamen tesadüfi olarak seçtikten sonra, nasıl oluyor da tüm sistem iki-üç nesil sonra kendiliğinden bir düzene giriyor ve bu düzeni sürdürüyor. Ekranın üst tarafında ortaya çıkan bir koloni grubu, ortadan kaybolduktan 50-60 nesil sonra aynı düşey doğrultuda nasıl yeniden ortaya çıkabiliyor? Bazı sistemler üç-dört satır sonra ölürken neden bazıları periyodik olarak devam ediyor? Neden bazı koloni grupları, ekranda hiç engel tanımadan önlerine çıkanı yiyerek belli bir doğrultuda hareket ediyor?

Kalın sağlıcakla... □


# T.E.P.K.İ.

Murat Aykul
Halkçı Sok. Sakoğlu Özbey İşhanı Altçarşı No.4 Bakırköy/İstanbul

# MATHMASTER 5000 TL.

**Öğrenciler, öğretmenler, mühendisler!** **Commodore artık güçlü bir matematik makinası. Geometri, trigonometri, logaritma v.s.**

- ☐ Frıends Commodore Clup (Üyelik formunu isteyiniz)
- ☐ Ücretsiz teknik danışmanlık servisi
- ☐ Baskılı devre servisi
- ☐ Oyun ve diğer programların yanında Türkçe açıklamalar
- ☐ Kopya, çizim, müzik, turbo-protect v.s. tipi özel açıklamalı programlar
- ☐ İstenilen programların kartuşta hazırlanması
- ☐ Siparişle program hazırlama

**FREEZE FRAME III 25.000 TL.**
**Extra fonksiyonlar**

GR-BASIC
C-64 için 128 ek komut
Müzik, çizim ve tüm işlemlerde kolaylıklar
BASIC ile Interrupt kontrolu
TÜRKÇE EL KİTABI İLE 5000 TL

10 şanslı arkadaşımız
1'er tane Mathmaster
1'er tane GR-BASIC
10'ar tane kendi istedikleri oyun programları kazandılar

Gürkan Engin (İstanbul)
Birsen Ongun (Bursa)
Murat Pakyürek (İzmir)
Mustafa Genel (Aydın)
Hasan Selçuk Turan (İstanbul)
Mesut Biçer (Adana)
Ferhat Kabaoğlu (İstanbul)
Göksel Eriş (İstanbul)
Tarkan Yamanoğlu (Samsun)
Çağlar Mete (Manisa)

**Şanslı arkadaşlarımız, seçtikleri oyunları kendileri gelerek veya mektup yazarak seçebilirler.**

# BİLGİSAYARLARIN DÜNYASINDA YAŞAMAYA HAZIR MISINIZ?

*Bilgisayar alacaksınız! Hangi bilgisayar neyi, ne kadar yapar? Karşılaştırmak, en uygun olanı seçmek için bilmeniz gereken her şey BİLGİSAYAR ANSİKLOPEDİSİ'nde var.*

*Bilgisayarınız var! En iyi nasıl kullanacaksınız? Daha iyi nasıl yararlanacaksınız? Temel bilgilerden başlayarak öğrenmek, öğrendiklerinizi uygulamak için BİLGİSAYAR ANSİKLOPEDİSİ okuyun.*

*Bilgisayarla çalışıyorsunuz! Daha iyi ve daha etkin programları nasıl yazacaksınız, işinize nasıl uyarlayacaksınız? Basic, Fortran, Cobol, Pascal, Assembler veya Logo dillerini öğrenmek için BİLGİSAYAR ANSİKLOPEDİSİ alın.*

*Gelecekle ilgileniyorsunuz! Bilgisayar çağında ne olacak? İnsanlar nasıl yaşayacak, ne yapacak? Bilgisayarların toplumsal düzenine hazırlanmak için BİLGİSAYAR ANSİKLOPEDİSİ'ne başvurun.*

7 cilt, 2008 sayfa, sayısız şema ve tablo, ofset baskı, 1. hamur kâğıt, lüks cilt.

Kullanılan terimlerin Almanca, İngilizce, Fransızca karşılıkları.




7 ciltte 6 Bilgisayar dili öğretiyoruz.

© Armando Curcio Editore, İletişim Yayınları.

T.C. Milli Eğitim, Gençlik ve Spor Bakanlığı tarafından okullara tavsiye edilmiştir.

| | | Peşin | TAKSİTLİ SATIŞ 4 ay | 8 ay |
|---|---|---|---|---|
| BİLGİSAYAR ANSİKLOPEDİSİ (7 CİLT) | TOPLAM FİYAT | 55.300 | 62.250 | 69.300 |
| | PEŞİNAT | | 18.250 | 15.700 |
| | TAKSİT | | 11.000×4 | 6.700×8 |

Adım: .....
Soyadım: .....
Adresim: .....
Telefonum: .....

Yazışma Adresi: İletişim Pazarlama-Klodfarer Cad. No: 7 34400 Cağaloğlu/İST.
Tlf: 512 17 32-33

BİLGİSAYAR
ÇAĞDAŞ TEKNOLOJİ
ANSİKLOPEDİSİ

*İletişim Çağında Bilgisayar;*
*Bilgisayar Çağında İletişim*

*şim Yayınları • İletişim Yayınları • İletişim Yayınları • İletişim Yayınları • İletişim Yayınları • İletişim Y*

# Trap Door




Oyunumuzda Berk (ekranda kumanda ettiğiniz sevimli yaratık) 'THE THING'in iğrenç yemeklerini hazırlamak zorunda. Bu konuda Berk değişik canavarların ve nesnelerin yardımını alıyor. Canavarlar 'trap door'dan (tuzak kapısından) çıkıyorlar, nesneler de bodruma yerleştirilmiş. Eğer Berk bir görev için gerekli nesnelerden bazılarını kaybederse, o görevi bırakıp sonrakine hazırlanabilir. Nesneler her zaman Berk'in son koyduğu yerde olacaklar. Eğer sonraki göreve geçmek istemiyorsanız, kendinizi 'trap door'dan aşağı atın. Böylece oyun baştan başlayacak.

Berk nesneleri taşıyabilir, itebilir. Ama bunları yapması için uygun pozisyonda olması gerekli. Ayrıca Berk istediği bölgede dolaşabilir. Her yeri dolaşmanız oyunun gelişimi içinde çok işinize yarayacaktır. Dolaştığınız yerlerde bulunan nesnelerin yerini de ezberleyin. Dolaşacağınız yer toplam üç kattan oluşuyor; başladığınız yer, balkon ve bodrum. Berk ileri geri de hareket edebilir. Ayrıca 'trap door'u açıp kapatan kolu kullanmaya alışmanızda da yarar var. Böylece acil bir durumda bu işi kolayca yapabilirsiniz.

Berk'in iki yamağı var. Boni konuşan bir kafatası. Alındığında bazen tavsiyelerde bulunuyor, bazen de hiç aldırış etmiyor. Adı Drutt olan örümcek ise yalnızca ayakbağı oluyor ve hiçbir işe yaramıyor.

## Birinci Görev

Berk'in birinci görevi oldukça basit. The Thing için bir kutu kurt hazırlayacak. Bu görevi başarıyla tamamlamak için doğru tenekeyi almaya dikkat edin. Trap Door'u açıp kurtları bırakın ve hemen kapağı kapatarak başka şeylerin çıkmasını önleyin. Drutt onları yemeden mümkün olduğunca fazla kurt yemeğe yakalamaya çalışın. Burada Drutt'dan kurtulmak için trap door'u kullanabilirsiniz. Trap door'u açarak daha fazla kurt yakalayabileceğinizi unutmayın. Konserve kutusunu alın ve aptal garsona yerleştirin. Sonra onu The Thing'e göndermenin bir yolunu bulmalısınız. Kurtların işini bitirdiğinizde size iyi şeyler söyleyecek ve ikinci emrini verecektir.

## İkinci Görev

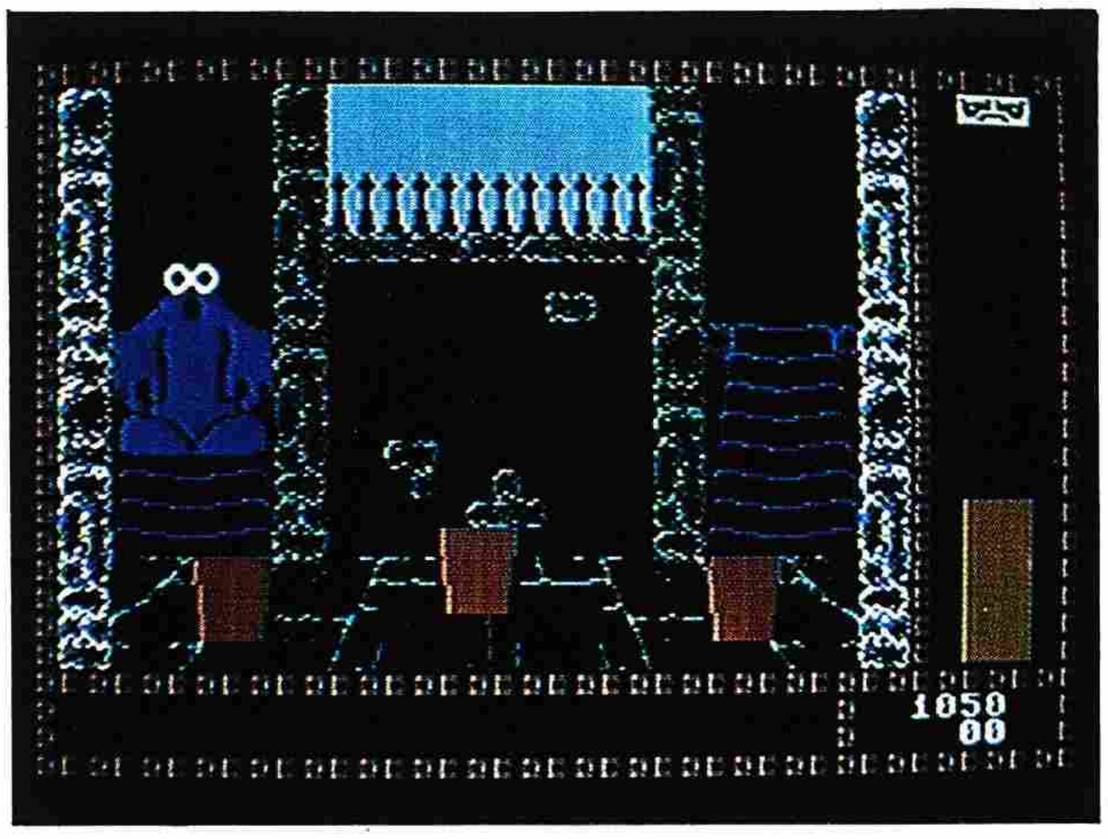
İkinci göreviniz kızarmış yumurta hazırlamak. Bunun için kapıyı açıp tavuğu dışarı bırakın ve kapıyı hemen kapamayı unutmayın. Tavuğu yumurtlatmak için arkasına mermi ile hamle yapmanız gerekli. Bu mermiyi nasıl bulacağınızı söylemeyeceğiz, ama mermi yerine (mermiyi kaybetmeniz ihtimali için de geçerli) başka nesneler de kullanabilirsiniz. Hattâ Drutt'u bile. Yumurtaları koymak için bir kap bulmanız gerektiğini ve yumurtaların kolay kırılır nesneler olduğunu unutmayın.

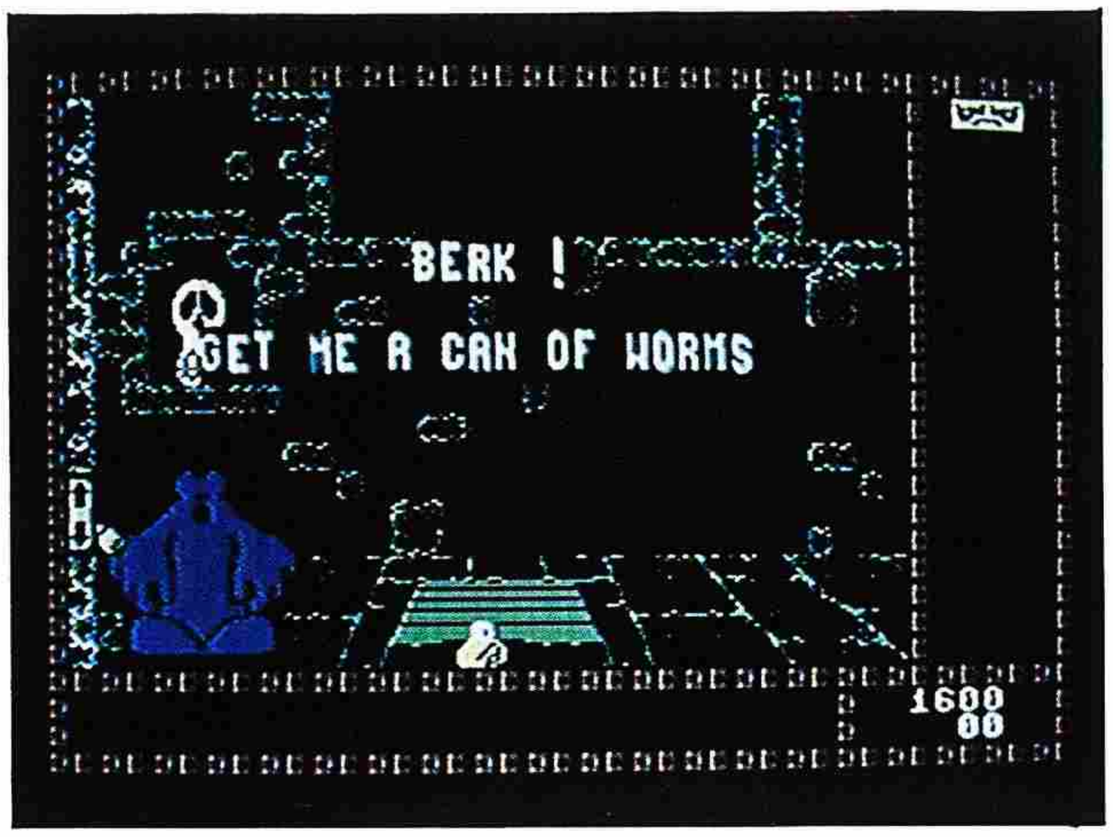
Tavuğun yumurtlamaya hazır olduğunu gözlerinin büyümesinden anlayabilirsiniz. Yumurtaları bir kapta topladıktan sonra pişirin ve yollayın.

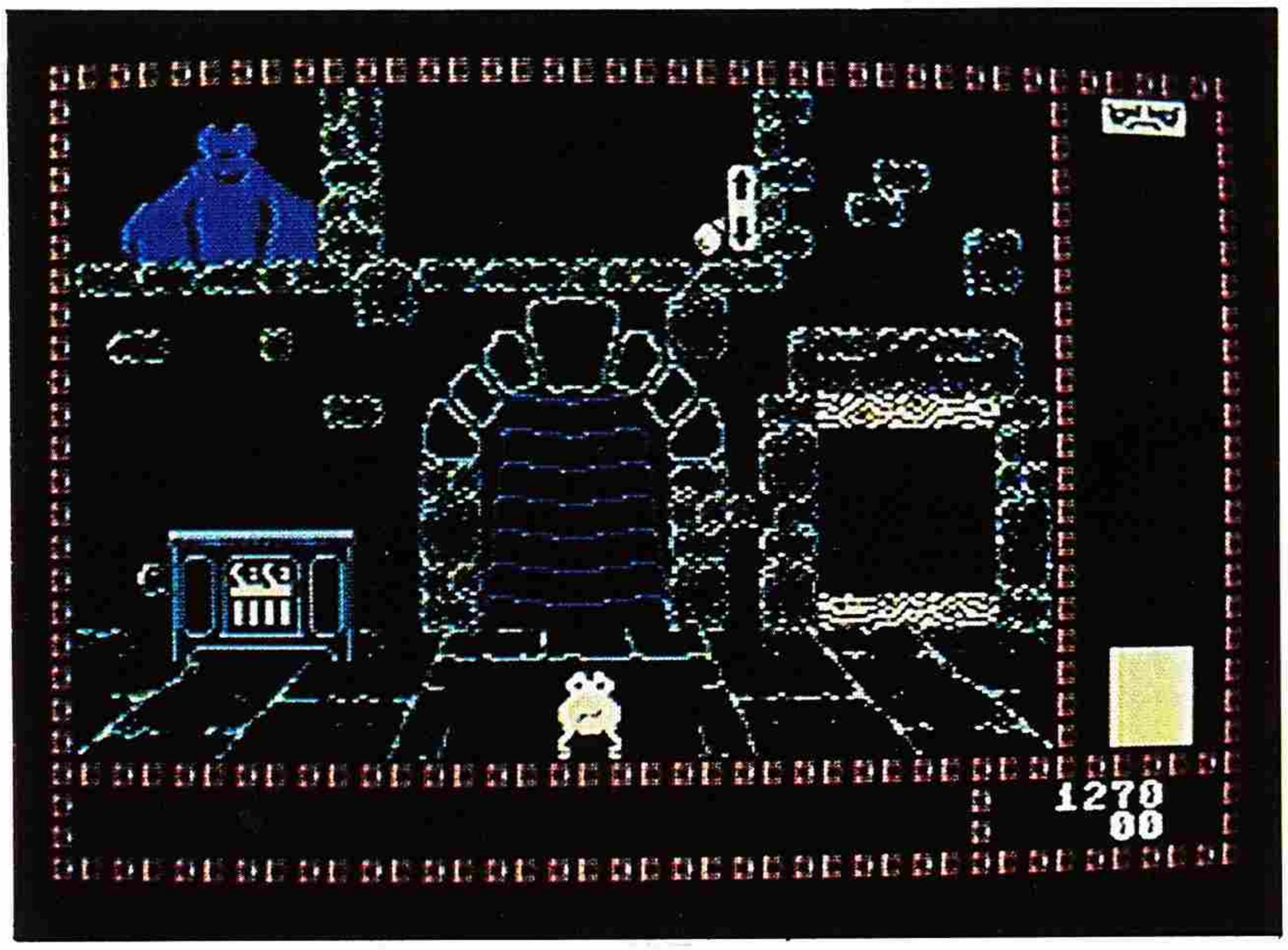
## Üçüncü Görev

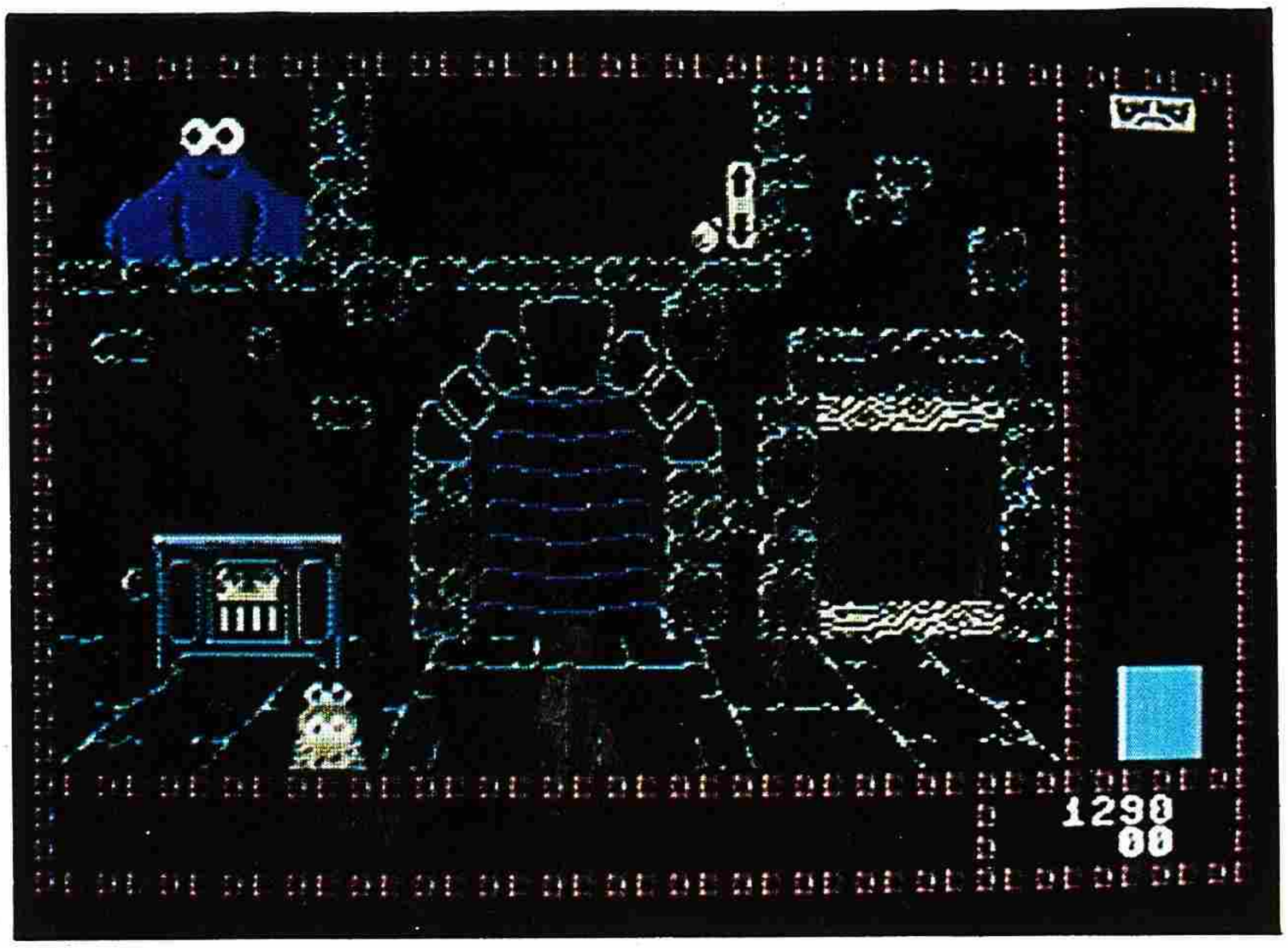
İşiniz gittikçe zorlaşıyor. Üçüncü göreviniz, kaynamış solucan hazırlamak. Bir kova ile su basmış bodruma inin ve birkaç solucan yakalayın. Bunları kaynatmak için büyük kırmızı kaba koymanız gerekiyor. Ama boyunuz kaba ulaşmak için yeterli değil. Bu nedenle daha yüksek bir yere çıksanız iyi olur.

Bu işlemden sonra ateş püskürten canavarı trap door'dan bırakmanız gerekiyor. Onu kullanarak solucanları kaynatacaksınız. Eğer kap kaynamazsa bu iş olmadı demektir. Öyle ki, yolladığınızda kap hâlâ kaynar vaziyette olmalı. Bu iş bittikten sonra canavardan kurtulmanız gerekli ama canavar trap door'a gitmeyecektir. Onun için ağır bir




şey bulup canavarı ezerseniz daha iyi olur.

## Dördüncü Görev

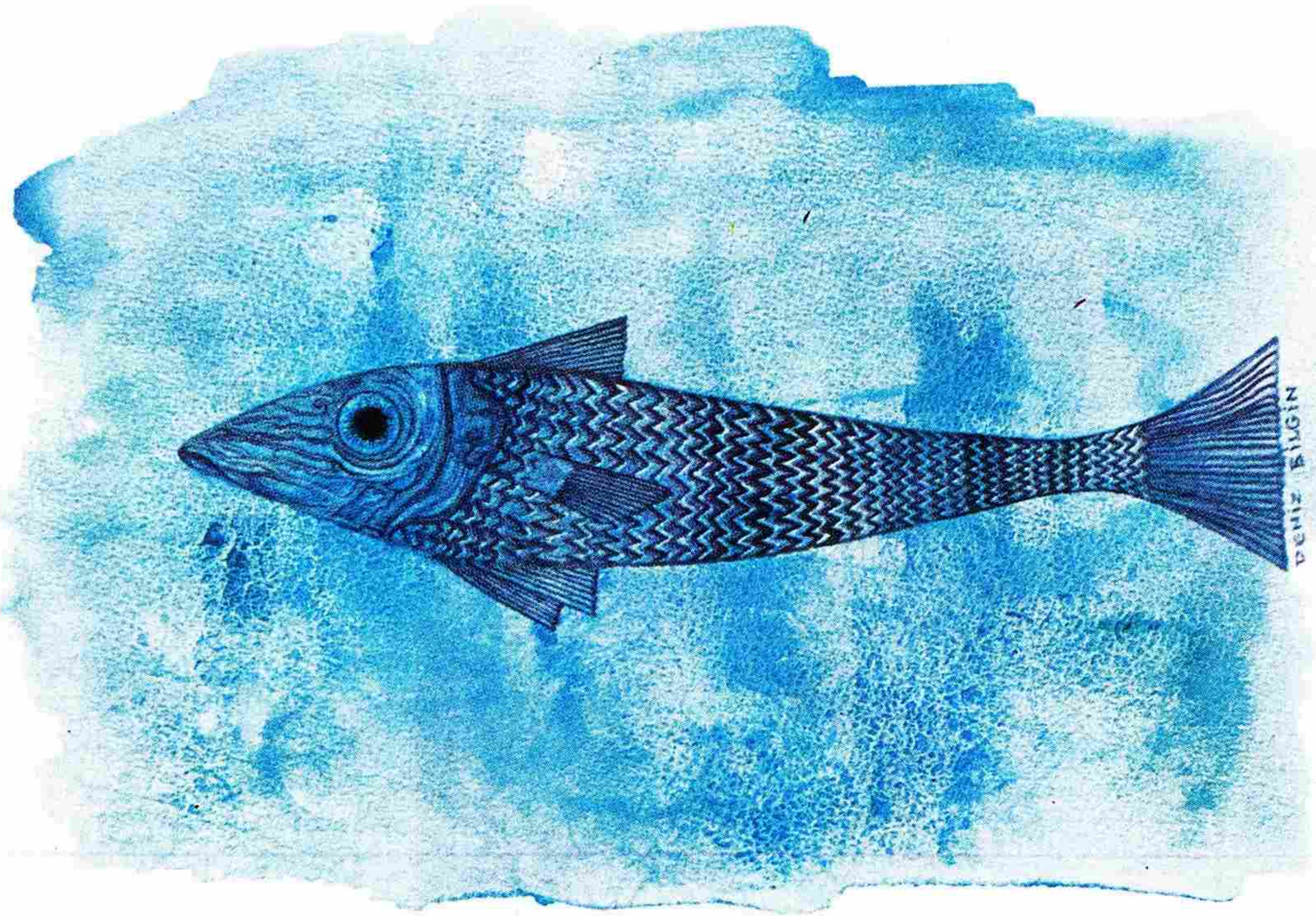
Dördüncü göreviniz, gözbebeği ezmesi yapmak. Gözbebeğini hiçbir yerde bulamazsınız. Sizin yetiştirmeniz lazım. Bunun için kullanabileceğiniz bir paket tohum var. Tohumları ekmek için makul bir yer bulun ve tohumları ekin. Büyülü bir şekilde bitki olmalarını izledikten sonra ürettikleri gözbebeklerini toplayabilirsiniz. Ezmeyi hazırlamaya başlayabilirsiniz. Sıvıyı uygun bir kapta toplayın ve The Thing'e yollayın, oldukça susamış olmalı (bu yediklerinden sonra).

## Beşinci Görev

Son göreviniz ortalığı toplamak. Bunun birçok yolu var, ama en uygunu hareket ettirilebilecek her şeyi toplayıp trap door'dan atmak. Hayalet göründüğünde yiyebilsin diye yenilebilir bir-iki şey saklamanız doğru olur. Kapıyı kapatmayı unutmayın.

İşte şimdi kasayı açmanın tam zamanı. Ne kasası diye soracaksınız. Onu aşağıda bulamayacaksınız, o zaman yukarıda olmalı. Ayrıca bulunca kasayı açmanın bir yolunu da bulmalısınız. Size bir ipucu verelim, kasayı açmak için oldukça ağır bir nesne kullanmanız lazım. □

# COMPU-HOBİ COMPU-FOBİ




## *Olmak ölmektir: Balık Burcu*

Ne zaman balık burcu dense, aklıma bir incir gelir- olmuşluğunun doruğunda ama dalından henüz düşmemiş bir incir. Görüntüsünden önce kokusuyla farkına varılır. Kendine özgü bir kırmızılıktaki damlalarıyla usaresini akıtırken, sanki yere, yerçekimine doğru eğilimini simgeliyor gibidir.

Eskiler bir şeyin sonunun, vardığı sonucun aynı zamanda onun yetkinliği, en mükemmel ân'ı olduğunu söylerlerdi. Ama betimlemeye çalıştığım türden incirin, bu bakımdan bir yanıltıcılığı vardır. Elbette o da eninde sonunda dalından düşecektir. Ama bu gerçeklik, olgunluğunun doruğunda olduğu o son ân'ın tılsımını azaltmaz. Sonu sanki sadece kendi geçici olgunluğunu değil, bütün toprağın ve özellikle de kendisine en yaraşan yer olan Ege'nin yetkinliğidir. İnsan ona bakar ve doğum-ölüm çemberinin kırılabileceğini, olgunluğun süreklilik kazanarak bir varoluş biçimine dönüşebileceğini sanır.

Ya da başka bir imge: Alabildiğine gerilmiş bir saz teli düşünün. Odadaki her titreşimden etkileniyor; her gölgeye, her renge, her duygu kıpırtısına ses veriyor: Ya da daha doğrusu her şey kendi sesini onda buluyor.

### Balığın Kötülüğü

Tekilliğinizi, bireyselliğinizi bir an için unutup onun muğlaklığına, bulanıklığına gözlerinizi dikmeye cüret edin. Orada kendinizi yeniden keşfedeceksiniz. Çünkü siz ona baktığınız sürece, sizi bir şekilde yansıtmayan, hesaba katmayan hiçbir şey yapmayacaktır.

Bu yüzden ona kızmak neredeyse olanaksızdır. Çünkü size en büyük kötülüğü yaparken bile, sizin bir yanınızı hesaba katmakta olduğunu farkedersiniz- bu yan, sizin kendinizde en sevmediğiniz, kendinize en karşı yan bile olsa. Size yapılmakta olan kötülüğü, aslında sizin kendi kendinize yapmakta olduğunuz, bunu hakketmiş olduğunuz duygusundan kurtulamazsınız. Belki de teninizi yarmakta olan soğuk madeni tutan el, kendi intihar tutkunuzdur.

Bütün bunlar kısaca "anlayışlılık" diye adlandırılabilecek -yanlış ve eksik olarak "anlayışlılık" diye adlandırılabilecek- bir özellikten kaynaklanıyor. Balık burcu insanı, her kesi içerden, kendisi oymuşçasına tanır.

Anlaşılmak mutluluksa eğer ve aşktan vazgeçebildiğiniz takdirde, kimse sizi onun kadar mutlu edemez.

Bütün bunlar dışardan bakılarak söylenmiş sözler. Balık burcunu sevmiş, ondan incinmiş ve onu sevmeye devam etmiş birinin söyleyebileceği sözler.

Ya balık burcunun kendisi?.. Onun yalnızlığını tarif edecek sözleri kim bulup çıkarsın? □

*Samet KUMAŞ*

# Yapısal Programlama

**Cenk EDEN**

*Siz; bilgisayarcılar arasına yeni yeni katılan yeni bilgisayar sahipleri. Siz; bunun gururunu, coşkusunu taşıyan ve sorumluluklarınızın bilincinde olan yeni gönüllüler. Bilgisayara "merhaba" dediniz ve belki de kullanım kılavuzundan veya derginiz Commodore'dan ilk programınızı yazdınız. Daha ilk adımda hata yapmamak ve ilk run'ınızda sevimsiz bir hata mesajı görmemek için, programınızı büyük bir dikkat ve özenle, en küçük ayrıntısını bile birkaç kez kontrol ederek yüklediniz.*

Tüm bu çabalarınıza rağmen, eğer programınız on satırdan uzunsa, büyük bir olasılıkla ilk denemenizde çalışmayacaktır. Olacak şey değil, programınız yapabileceği tüm yanlışlıkları yapıyor ve siz bunun nedenini bir türlü bulamıyorsunuz.

Programınıza bir daha -belki de 101. kez- göz attığınızda onu bir tabak spagettiden daha karışık buluyorsunuz. Açıkçası programı düzeltme şansınız o kadar zayıf ki, fişi çekip yeniden başlama fikri bir anda çok cazip hale geliyor.

Bilgisayarın başına oturduğunuz o heyecan dolu, coşku dolu ilk günlere dönerseniz herhalde yukarıdaki tablonun size de pek yabancı olmadığını göreceksiniz. Pek çoğumuz yaşadık o umutsuzluk dolu saatleri ve belki de yeni bir kavram kattık dağarcığımıza; errorofobi. Çözümü yok muydu, cevapsız mıydı bu sorun? Elbette var; çözüm de sorun kadar önemli üstelik; disiplinli programlama daha doğru bir ifade ile programlarınıza disipline edilmiş bir yaklaşımla eğilmenizdir. Bilgisayarınızın tuşuna bastığınız ilk andan itibaren bu tekniği kullanmaya ve geliştirmeye çalışmalısınız. Profesyonel software yazarları -yazılımcılar- yapısal programlama tekniklerini kullanırlar.

Yazılımcılar ilk olarak çözmek zorunda oldukları problemlerin -fonksiyonel blokların- çözüm yollarını bulurlar ve tüm bu çalışmalar yapıldıktan sonra, işin nispeten kolay yanı olan aşamaya gelinir. Bu aşamada, ortaya çıkan program planı bilgisayarın işleyeceği şekle sokulur. Yani program bilgisayar dillerinden birine çevrilir.

Bir amatörden tüm bu çalışmaları yapmasını beklemek belki biraz fazladır. Bir de buna amatörün mütevazi sayılabilecek amaçlarını eklerseniz, isteklerimin bir amatör için insafsız sayılabilecek kararlar olduğunu düşünebilirsiniz. Çünkü amatör, programlamanın getireceği nimetlerle ilgilenmez, onun için biçimsel ve yöntemsel kazançlar anlamsız birer kavram olmaktan ileri gidemezler. Amatörün tek istediği, programının sonucunu mümkün olduğu kadar kısa sürede ekranda görmektir. Bununla birlikte eğer bu tür bir disiplini programlarınızda biraz olsa bile kullanırsınız, programlarınızın daha kolay yazılır hale geldiğini görecek ve hepsinden önemlisi yanlışlarınızı daha kolay düzelttiğinizi farkedeceksiniz.

Programınız için çizilecek birkaç akış diyagramı, size programınızın bir sonraki adımda ne yapacağını görme olanağını verir. Böylece programınızın bölümler arası koordinasyonunu sağlama olanağınız olacaktır. Şimdi sıra programın gerçek anlamda yazımına geldi.

Madem bir süre sonra Basic kullanıyor olacaksınız en azından onunla başlayacaksınız, öyleyse iyi kurulmuş bir Basic programında neler olması gerekiyor bir de buna bakalım. Öncelikle altprogramları gözden geçirelim. Programınızda her biri akış diyagramınızdaki bir kutuya karşılık gelen altprogramlardan bir dizi olmalıdır. Bunun sonucunda program parçalarınız küçük bloklar haline gelecek ve böylece tüm altprogramı ekrana aynı anda listeleyebileceksiniz.

Her altprogram açıkça tanımlanmış bir fonksiyonu yerine getirmeli ve ana programdan bağımsız olarak test edilebilmelidir. Bu testi gerçekleştirmek için sonuçlarını bildiğiniz test değerlerini altprograma girin ve altprogramı çalıştırın, eğer çıkan sonuçlar sizin kendi sonuçlarınızla aynı ise, altprogram, bu kez farklı değerlerle, çalıştığında çıkacak sonuçların kesinlikle doğru olacağından şüpheniz olmasın.

Basit bir örnek ele alalım. Farzedin programınızda bir dairenin alanının hesaplanması gerekiyor. Bunu hesaplayacak altprogram aşağıdakine benzer bir rutin olacaktır.

```
998 REM R YARIÇAPLI
99 REM DAİRENİN HESABI
1000 A = π xRxR
1010 RETURN
```

Bu rutin oldukça basit olmasına rağmen değindiğim prensibi gerçekleştirir niteliktedir. Çünkü rutin sadece bir işlevi yerine getiriyor ve kolayca test edilme olanağı var. Aşağıdaki satırı yazıp RETURN ederseniz, sonucun 28.274 veya buna yakın bir

```
R = 3: GOSUB 1000:PRINT A
```

sayı olduğunu göreceksiniz. Bu rutin, her ne kadar, görevini yukarıda saydığım prensiplere göre eksiksiz yapıyorsa da, mükemmel bir altprogram sayılmaz. Çünkü bu işlemi bir kez yaptıktan sonra kullandığınız modülü (satır) unutabilirsiniz. Ayrıca; gerçi biraz ayrıntı sayılsa da, programınızda R değerinin belirli bir sınır içinde olup olmadığını kontrol eden daha derin bir detay olabilirdi. Bu yöntemle eğer R belirlenen sınır içinde değilse uygun bir hata mesajı basılabilir, böylece programın ana kısmı bir hata yapıldığından haberdar olabilirdi.

Öyleyse daha düzgün, daha kullanışlı bir altprogramın nasıl düzenlenmesi gerekiyor? Öncelikli programın doğrusal akışını korumanız gerekiyor. Mademki altprogramın çağrıldığı satıra geri dönmesi için Return komutunu kullanmak zorundasınız, o zaman kontrol de düzgün bir şekilde ilk satırdan son satıra kadar akmalıdır.

Bir altprogramdan programın başka bir bölümüne hiçbir atlama olmamalıdır. Ancak bu genellemeye bir başka altprogram girmez; yani alt-

programlar arası yolculuk serbesttir. Ayrıca bir altprogramı tamamladığınızda, altprogramda "gevşek son" olmaması gerekir. Gevşek son deyimi tamamlanmamış For-Next döngüleri için kullanılır. Bu basit kurallara uyulduğu takdirde, bir hata durumunda altprogram ana programdan kolayca sökülebilir.

Basic çoğunlukla bilgisayarcıların ilk göz ağrısıdır. Ancak bu özel konumuna rağmen, disiplinli bir programlama tekniğinin geliştirilmesi için kullanılabilecek en iyi dil olduğunu söylemek zordur. Çünkü pek çok şeyi geçiştirmenize izin verir; For-next döngüleri buna iyi bir örnektir. Bu döngüler genellikle programcı için bir sorun kaynağı olurlar. Örneğin alttaki rutin insana oldukça makul gözüküyor.

```
100 FOR I=1 TO 100
110 GET TB$
120 IF TB$ "" THEN 350
130 NEXT I
```

Bu basit bir "reaksiyon testi" rutinidir. Oyuncuya herhangi bir tuşa basması için belirli bir süre tanır. Eğer oyuncu bir tuşa basarsa program hangi tuşa basıldığını bulmak için başka bir satıra atlar (350), eğer basılmazsa program altprogramda devam eder -daha doğrusu döner de döner-.

Hem altprogram, hem de açıklaması makul gibi gözükse de, bir tuşa basmakla For-Next döngüsünden çıkmış oluyorsunuz, yani ikinci kuralımızı -gevşek son- ihlal ediyorsunuz. Bu altprogramın işlevini bir Repeat-Until komutu ile çok daha iyi bir şekilde gerçekleştirebilirsiniz. Eğer bu komut Basic komutlarımız arasında yoksa ve maalesef yok, yukarıdaki fonksiyonu aşağıdaki rutin ile yapmak zorunda kalacaksınız.

```
 99 REM TUŞA BASILDI MI?
100 FOR I=TO 100
110 GET TB$
120 IF TB£     "" "" THEN I=
    100:TB=1
130 NEXT I
140 IF TB=1 THEN GOSUB
    1000:REM TUŞA BASIM
    ALTPROGRAMI
150 RETURN
```

Satır 120'de I limit değerine (100) ayarlanıyor. I 100'e eriştiğinde bilgisayar TB'nin değerine bakarak döngüden çıkıyor ve o anda, bir tuşa basılıp basılmadığını görmek için TB'yi test ediyor. Eğer TB 1 ise bilgisayar tuşabasım altprogramına atlıyor. Tabii bu rutinin her kullanımından önce TB değerinin 0'lanması gerekiyor. Bu örneğimizde tuşa basıldığında döngünün açık kalmaması için satır 120'de I'nın limit değerine ayarlandığını görüyoruz. Bu durumda rutinimiz bu kez ikinci kuralı ihlal etmekten kurtuluyor.

Yapısal programlama tekniği demek sadece altprogram yazmak değildir. Bu nedenle biraz da esas programımıza göz atalım. Esas program, yazdığınız tüm programın başında başlamalı, sonuna gitmeli ve orada bitmelidir.

Kontrol ilk satırdan itibaren akmaya başlamalı ve sanki gidiyormuşçasına altprogramları çağırarak, END komutunda sona ermelidir. Son olarak GO TO komutundan bir beladan sakınıyormuşçasına kaçınmanızı tavsiye ediyorum.

Üst paragtafta söz edilen ve öğütlenen düzenlemenin ancak katı ve tekdüze bir program için geçerli olacağını düşünebilirsiniz. Öyle ya eğer programın akış yönün değiştiremiyorsak, programa nasıl farklı işler yaptıracağız? Sorun ne kadar ciddi ise cevabı da o kadar basittir: Ana rutininiz altprogramları duruma göre çağıracaktır. Yani farklı sonuç farklı altprogramın çağırılması demektir. Yukarıda verdiğim tuşabasım örneğinde bu prensibin işleyişini kolaylıkla görebilirsiniz.

Altprogramların güzelliği istenildiği kadar uzun yazılabilmelerinden kaynaklanır. Altprogramlar, her zaman için, kontrolden geçtikten sonra, kontrolü çağırılmış oldukları noktaya -satıra- geri gönderirler. Böylece programın doğrusal akışı korunmuş olur.

En doğrusu tüm altprogramların asıl programın arkasına, bir End komutu ile ayrılarak, yerleştirilmesidir. Bununla beraber, eminim siz de farkındasınız, Basic derleyiciler satırları yukarıdan aşağıya doğru tararlar. Bu nedenle eğer programınızda hız önemli bir faktörse, en çok çağırılan altprogramınız programınızın başlangıcında olmalıdır. Daha doğru bir yaklaşımla programınızın ilk satırında en çok çağırılan altprogramınıza atlayacak ve ana programın işleyişini başlatacak olan bir GO TO komutu olmalıdır. Bu düzenleme pek sevimli olmasa da oldukça yararlı ve etkilidir.

İyi programlama tekniği hakkında söylenebilecek daha pek çok şey var, bu makale bu tekniğin önemli odaklarından sadece biri olan ve programcıya programın akışını takip edebilme olanağını veren, disipline edilmiş yaklaşım tekniği üzerine kurulmuştur. Çünkü programcının, programı çalıştırdığında, programın kullanacağı yolu seçmiş olması ileriki adımlarda hayati önem taşır.

Bu makalede yeterince vurgulanmayan tek bir nokta kaldı o da: Programlama tekniğinin öğrenilme ve geliştirilmesinin göründüğü kadar zor olmadığı, ancak bir makale okumakla da başarılamayacağıdır. Eğer koyduğum kurallara uymaz; canınızın istediği yerden sıçrar, gevşek sonları dikkate almaz, GO TO'larla istediğiniz gibi oynarsanız eninde sonunda çuvallarsınız. Programlarınız zamanla gelişecek ve daha karmaşık bir hal olacaktır; o zaman sözünü ettiğim bu problemler de gittikçe çözümü daha zor hale gelecektir. Bunun yanında programınızı büyütmek için yeni satırlar girmek istediğinizde, klavye başında güç saatler geçirebileceğinizi de unutmayın çünkü -eğer yarım saattir okuduğunuz tekniği kullanmazsanız- en fazla bir hafta sonra programınızda kullandığınız mantık zincirini unutabilirsiniz.

Bir süre sonra Basic size dar gelmeye başlayacak ve büyük bir ihtimalle makine diline yöneleceksiniz. Orada karşılaşacağınız problemlerin Basic'tekilerden tek farkı: çözümlerinin daha zor ve uzun oluşudur. Bu nedenle makine dilini de kullanırken Basic'teki temel prensiplere başvurmanız gerekecek ve tabii iyi bir program için iyi bir program tekniğine (hangisine acaba?).

Sonuç olarak; programlama düzeyiniz ne olursa olsun yapısal programlama tekniğini öğrenmeniz, geliştirmeniz ve basit, karmaşık demeden her programınızda kullanmanız gerekiyor. Bu yaklaşım, eğer iyi bir programcı olmak istiyorsanız, hem sizin hem de programlarınızın vazgeçilmez ögesi olmalıdır. Unutmayın; yanlış hesap Bağdat'tan döner örneği; temelsiz akış diyagramsız, plansız bir program da ALU'dan (Aritmetik-Mantık Ünitesi) döner. □

# Teleteknik Matematik Dersinde

## BİLÇAĞ

MAT.0/1 .1a - Matematik

Kesirler
Standart Form
Sayı Tabanları

Öğrencinin aynen sınıfta kara tahta üzerinde çalıştığı gibi, adım adım, simülasyon tekniğiyle hazırlanmıştır. Konuyu yeterince anlatacak örnek mevcuttur.

MAT.C/I .1b - Matematik test

Öğrencinin bu paketteki tüm konuları iyice anlayıp anlamadığını ortaya koyar. Eğer başarı oranınız düşükse hatalı olduğunuz soruları ilgilendiren konuları tekrar çalışın.

MAT.0/I .2b - Matematik uygulama

E.K.O.B.
E.B.O.B.
Çarpanlara ayırma

Öğrencinin aynen sınıfta kara tahta üzerinde çalıştığı gibi, adım adım, simülasyon tekniğiyle hazırlanmıştır. Konuyu yeterince anlatacak örnek mevcuttur.

MAT.O/I .2b - Matematik

Kümeler

Öğrencinin aynen sınıfta kara tahta üzerinde çalıştığı gibi, adım adım, simülasyon tekniğiyle hazırlanmıştır. Konuyu yeterince anlatacak örnek mevcuttur.

MATEMATİK Programı yüklendiğinde karşınıza programın içindeki konular gelir. Bu sayfadan sonra sıra ile sorular sorulacaktır. Sorunun cevabını a,b,c,d veya e şıkkından seçerek o şıkkın harfini yazarak RETURN tuşuna basınız.

Eğer cevabınız doğru ise iki seçeneğiniz vardır. Bu sorunun nasıl çözüldüğünü görmek istiyorsanız RETURN tuşuna basınız. Bir sonraki soruya geçmek istiyorsanız SPACE tuşuna basınız.

Eğer cevabınız yanlış ise, size hangi cevabın doğru olduğu gösterilecek ve bir müddet bekledikten sonra sorunun çözümü adım adım anlatılacaktır. Cevap açıklaması bittiğinde karşınıza iki seçenek gelir. Bu soruyu tekrar için RETURN tuşuna basınız. Bir sonraki soruya geçmek için SPACE tuşuna basın.

Bütün sorular bittiğinde ekrana PROGRAM SONU yazısı ile beraber ikişer seçenek gelir.

Eğer bu programı tekrar çalışmak istiyorsanız t ve RETURN tuşlarına basınız. Programı tekrar baştan itibaren çalışabilirsiniz. Eğer programı bitirmek istiyorsanız b ve RETURN tuşlarına basınız.

Eğer programlar herhangi bir nedenle kesilirse GO TO 1 yazarak programın başına dönebilirsiniz.

MAT.0/II.1a - Matematik uygulama

Ortalama hız
Orantı
Değişme

Öğrencinin aynen sınıfta kara tahta üzerinde çalıştığı gibi, adım adım, simülasyon tekniğiyle hazırlanmıştır. Konuyu yeterince anlatacak örnek mevcuttur.

MAT.0/II.1b - Matematik test

Öğrencinin bu paketteki tüm konuları iyice anlayıp anlamadığını ortaya koyar. Eğer başarı oranınız düşükse hatalı olduğunuz soruları ilgilendiren konuları tekrar çalışın.

MAT.0/II.2a - Matematik uygulama

Denklemler

Öğrencinin aynen sınıfta kara tahta üzerinde çalıştığı gibi, adım adım, simülasyon tekniğiyle hazırlanmıştır. Konuyu yeterince anlatacak örnek mevcuttur.

MAT.0/II.2b - Matematik uygulama

Grafik çözümler

Öğrencinin aynen sınıfta kara tahta üzerinde çalıştığı gibi, adım adım, simülasyon tekniğiyle hazırlanmıştır. Konuyu yeterince anlatacak örnek mevcuttur.

MAT.0/III.1a - Matematik uygulama

Yüzdeler
Maliyet fiyatı
Satış fiyatı
Basit faiz
Faizler

Öğrencinin aynın sınıfta kara tahta üzerinde çalıştığı gibi, adım adım, simülasyon tekniğiyle hazırlanmıştır. Konuyu yeterince anlatacak örnek mevcuttur.

MAT.0/III.1b - Matematik test

Öğrencinin bu paketteki tüm konuları iyice anlayıp anlamadığını ortaya koyar. Eğer başarı oranınız düşükse hatalı olduğunuz soruları ilgilendiren konuları tekrar çalışın.

# KÜÇÜK İLANLAR

120 tane oyun + Sam İngilizce konuşan program + Sam Türkçe konuşan program. Hepsi 4000.TL.- (Kaset parası dahil).
Alkım Erentok
*Yenisubay Loj.Karaağaç Ap.No: 2/5 Kaleiçi-EDİRNE*

---

Bir kasette adres masraf banka ajanda defteri programları ve Simon's Basic ayrıca 40 oyun 3500.TL.-
Emre Güngör
*Tel: 357 36 28*

---

Satılık Commodore 64 + Datasette + Quick Shot II + 70 Oyun + Program Dökümleri ve İngilizce Türkçe Kullanım Kılavuzları ile birlik (170.000.-TL.)
Murat Uğur
*Tel: 173 10 21*

---

Atari 800XL Teyp + Joystick + Oyun programları satılık ya da Commodore ile değiştireceğim.
Metin Vural
*521 83 40*

---

C-64'ü olan arkadaşlar için satılık veya değiştirilebileceğim oyun ve müzik programları. Liste isteyiniz. 1 kasette 30 oyun + Simon's Basic 6500.TL. ilgilenenlere.
A.Yekta Nazlı
*Refik Belendir Sk. No: 28/9 Park Ap. Y.Ayranıca/ANKARA*
*Tel: 38 24 64*

---

120 oyun kaset parası dahil olmak üzere 4000.TL.-
Alkım Erdinç
*Karaağaç Ap. K:2 No:5 EDİRNE*

---

Spectrum 48 K + bilgisayarımı yanında joystick, Interface ve bir Commodore 64 ile değiştirmek istiyorum.
*Tel: 584 11 74*

120 tane oyun + Sam İngilizce konuşan program + Sam Türkçe konuşan program. Hepsi 4000.TL.- (Kaset parası dahil).
Alkım Erentok
*Yenisubay Loj.Karaağaç Ap.No: 2/5 Kaleiçi-EDİRNE*

---

1 kasette 50 oyun 6000.TL.- (Liste isteyiniz). İsteyenlere oyun değiştirebilirim.
Tahir Tekeli
*Ergenekon Cad. Beyaz Ap.No: 21/8 Yenişehir-KARABÜK*

---

SVIqSpectra Video Bilgisayar satılık veya C-64 ile değiştirilebilir.
Murat İslamoğulları
*Tel: 166 12 05*

---

Serhat Güç
Eski Londra Asfaltı Uçak Cad. No:55 K:319 Haznedar/İSTANBUL
Eğer izniniz olursa küçük ilanlar bölümüne ben şu ilanı vermek istiyorum:
reset swich = 1000.TL.-
50 C-64 oyunu = 2000.TL.-
Hepsi birden = 2500.TL.-
Taner Kaya
*Tel: 337 79 35*

---

Bir kasette 15 tane 86 oyunları 3000.TL.- (liste isteyiniz). Bir kasette 50 oyun + 50 SAM + 50 SIMON'S BASIC + 50 M.O.A. + sözlük programı (Listeden 86 oyunlarından seçeceğiniz her oyun 500.TL.- İstanbul'dakilere ellerine teslim edilir.
Koray Renkoğlu
*Tel: 333 80 89*

---

100 adet C-64 oyunu M.D.E.SAM Sam/Reciter ve 5 alt programı Matematik ve İngilizce testleri, Simon's Basici 10.000.TL.-
Tanık Aldeniz
*Londra Asf. 51/3 EDİRNE*
*Tel: 4222*

Kullanılmış Disk Drive arıyorum. En son oyunlar 500 TL. Ayrıca 30 derste Basic programı 2000 TL. Ayrıca ITT Schaub-Lorenz P40-L Miniset satılık.
Ozan Çırak
*Tel: 160 81 66*

---

Satılık Commodore 64 + 1530 Datasette unit + 2 adet C-64 özel joystick + Quick Shot II + C-64'ün 72 tane programı (Kitap halinde) + C-64 GRAPHICS AND SOUND kitabı + Uygulamalı 99 Basic programları + Basic Bilgisayar Programlama Dili + C-64 Uygulamalı Basic Programcılık + Bilgisayar Dökümanları + 100 adet oyun + 8 İngilizce kaset ve 2 tane ses kaseti + Makine Dili Editörü + Basic Dili Editörü + Copy 190 + Kafa Ayarı + Adres Masraf Banka Ajanda Defter Programları (kaset halinde) Fransızca Kullanma Kılavuzu Fiyatı = 450.000.TL.- Not: Simon's Basic de var.
*Tel: 19776*
*Adapazarı-SAKARYA*

---

Satılık Sinclair 2X Spectrum 48 K bilgisayar + adaptör adaptör teyp TV kabloları + 40 yakın klavye eğitimi oyun ve eğitim kasetleri + 2 orijinal kullanım kitabı ve Basic ile 280 makine dili kitapları 100.000 TL.- Commodore 64 ile değişebiliriz. İlgilenenler şu adrese mektup göndersinler:
Kemal Kaplan
*Yıldıztabya Mh. Divriğ Sk. No:15 G.O.Paşa-İSTANBUL*

---

Adanalı C-64'cüler, 200 oyun + Copy, 190 + Simon's Basic + SAM + Paint-Magic + V-3 Turbo programlarımızı takas etmek istiyoruz. İsteyenlere satabiliriz.
Burak - Hakan
*37047* *32972 ADANA*

Commodore

COMSOFT program yazılım kulübünü kurduk. Her çeşit bilgi alış verişi için üye olun. Ayrıntılı bilgi için: Adres: Hava Subay Lojmanı No:37/3 10050 Balıkesir
*İzmir tel: 233814 veya 234848*

---

C-64
Termoprinter + Joystick + Kasetli oyunlar. 600.000 TL.
*Tel: 360 77 43*

---

İstanbul içinde olmak kaydıyla kaset dahil satılık; 83-84 yılı 28 adet 1500.-TL, oyunları 30 adet 2500, 82 adet 2500, 30 adet 1500.-TL. Ayrıca ayrı kasetlerde Simon's Basic, Robotek, Terrormolinos tanesi 1100.-TL.
Ali Duyar
*Anadolu Sitesi Çiftlik Sk. No:9 Beyoğlu Apt. A Blok Daire:22 Acıbadem-Kadıköy/İstanbul*

---

C-16/+4 + Joystick + 1531 datasette + 20'ye yakın oyun kaseti + Basic kaseti + Yabancı oyun programlama kitabı + Yabancı kullanım el kitabı üstüne ödeme yaparak C-64 veya 64'le değişmek istiyorum.
*Adres: 56/5 SK. No:7 Bahar Ap. Daire: 10 Güzelyalı-İzmir Tel: 31-5583'den Serkan Çil*

---

İstanbul'da olanlar için 15 oyun + İngilizce + Sam + Kaset parası = 5000 TL. Papatyalı Sk. No:8 Çiftehavuzlar-Kadıköy/İstanbul
Ali Kaya
*Tel: 369 41 69*

---

100 oyunluk hazır bir kasedi değişmek ya da satmak istiyorum.
Tayfun Çil
*Tel: 366 22 67*

---

50 Seçme Oyun + Simon's Basic + MDE + BDE + SAM = 5000 TL. (Kaset benden)
Lütfen liste isteyiniz.
Erkan Ücük
*İvedik Cad. 9. Sk. No:12/7 06200 Demet-Gül/Ankara Tel: 451655*

---

Bende 50'yi aşkın oyun var, bir de Freezee Frammim ben bu oyunlardan arkadaşlarımın da yararlanmasını istiyorum, ilanım şöyle:
Turbosuz oyunlarda 100 adet C-64 oyunu 6000 ve sonrası 500 TL'sı.
Kaset posta masrafları dahil.
Liste isteyin.
Alp Kaleli
*Dilaver Bey Mh. Türk Ocağı Sk. Hocaoğlu Ap. D:6 22100 Tel: 4404 Edirne*

---

Turbolu güzel oyunlar ve Sam Rasaytır programı çok ucuza satılık.
Eylem Cülcüloğlu
*Tel: 360 30 92*

---

Program Kopyacıları Müjde! Teybinize çok ucuz fiyata Halt Copy cihazını takıyorum. Sadece 7500 TL. Ayrıca en son oyunlar ile yazışmak istiyorum. (Liste isteyiniz.)
MAGIC SOFTWARE
Tolga Sağlam
*Mithatpaşa Cad. No: 488-490/5 İzmir Tel: 140424-144815*

---

C-64 klavye, teyp, üçyüz oyun satılık.
Burak Oğuz
*Tel: 356 93 35*

---

Amstard
6128 + COBOL + FORTRAN + M.BASIC + TURBO PASCAL + COMPILER + OYUN PROG. 700.000 TL.
Kenan Yakut
*Tel: 339 75 21*

---

Tanesi 400 liradan 100 adet spectrum oyunu ve joystick arabirimi satılıktır.
Mustafa Cihan
*Narlıca Mh.341 Sk.No:34 01180 Adana*

---

120 adet C-64 oyunu Sam/Reciter ve 5 alt programı Koala Painter 1 ve II Simon's Basic ve I ve II Matematik ve İngilizce Testleri M.D.E, B.D.E Spectrum Simülatör Sam, Machine Code, 8 Derslik Basic Kursu sadece 5000 TL.
Tarık Aldeniz
*Londra Asfaltı No:51/3 2210 Edirne Tel: 14222*

---

Satılık ZX Spectrum + Bilgisayar 130.000 TL. (Violet Computer)
Osman Yaldız
*Bağdat Cad. No:378/2 Şaşkınbakkal/İstanbul*

---

Oyun değiştirilir. Ucuz oyun.
Ferit Nacar
*Tel: 572 01 18*

---

Kaliteli oyunlar ve ilginç programlar 500 TL. Ayrıca satılık kopya cihazı 20.000 TL.
Ozan
*Tel: 160 81 66*

---

Satılık C-128 + Datasette + 2 joystick + 7 oyun
Ahmet Verimli
*Tel: 140 84 48*

---

1700 C-64 oyun, 2000 C-64 utility ve 120 Amiga programım var. Benimle değişmek ister misiniz? (Sadece disket) Kendi listenizle birlikte müracaat.

Özcan Şentürk
*Başaran Sk. No:42/2 Mimar Sinan Mh. 45010 MANİSA*

# Modüler Programlamanın Anahtarı PASCAL (1)

**Burak KİPER**

Bilgisayar programlarını yazmak için kullanılan dillere PROGRAMLAMA DİLLERİ denir. Hepimizin bildiği gibi birçok programlama dili bulunmaktadır. Bunları kabaca üç ana dala ayırmak mümkündür: Makine dili, assembly ve yüksek düzeyli diller.

Yüksek düzeyli diller genellikle her programcı tarafından kullanılır. Yüksek düzeyli dillerin bu derece popüler olmasının sebebi makine dilinden veya assembly'den çok daha kolay bir kullanimı olmasıdır. Başka bir neden de yüksek düzeyli dillerin taşınabilir olmasıdır, yani herhangi bir makinede yaptığınız programı, çok ufak değişikliklerle başka makinelerde de çalıştırabilirsiniz. Buna karşın bir makine dili veya assembly ile yazılmış programı sadece bir bilgisayarda çalıştırabilirsiniz.

Bazı yaygın yüksek düzeyli diller, BASIC, FORTRAN, COBOL ve PASCAL'dır. Her dilin kendisini belirleyen özellikleri vardır. Bilgisayardan bilgisayara, bu özelliklerin dışında bazı özellikler de bu gibi dillere eklenmiş olabilir. Mesela grafik özelliklerini katmak için o dilin standardında olmayan komutlar eklenebilir.

Yüksek düzeyli dillerin bize sağladığı en büyük özelliklerden birisi de, kendi kullanmakta olduğumuz lisana benzer terimleri kullanabilmemizdir. Kullanacağımız bilgileri koyacağımız adresleri, onları tanımlayabileceğimiz isimler altında saklayabiliriz. Bundan başka çeşitli aritmetik işlemleri alışık olduğumuz şekillerde tanımlayabiliriz. Mesela birçok yüksek düzeydi dilde 'Z: = X + Y' komutu X 'ı Y'ye ekle ve sonucu Z'de sakla anlamına gelmektedir.

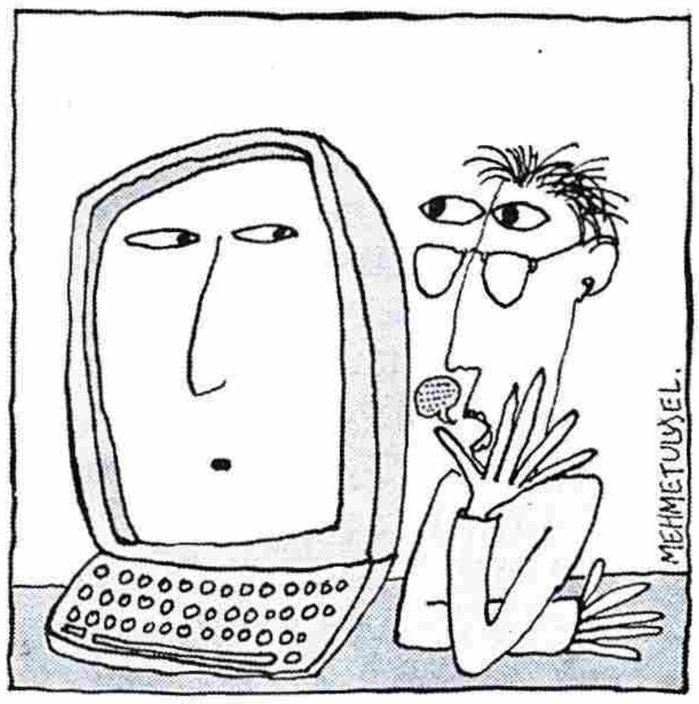
PASCAL, 1971 yılında İsviçre, Zürih'te Profesör Nicklaus Wirth tarafından geliştirilmiştir. Şu sıralarda programlama öğretmek için yaygın bir şekilde kullanılan bir programlama dilidir. Popüler olmasının sebebi yapısını öğrenmenin kolay olmasındandır.

Bundan başka PASCAL size modüler programlar yazma imkânı verir. Bu tip programları okuma, anlama ve çalışma sırasını düzenlemek daha kolaydır.

Yazımızın bu bölümünün geri kalan kısmında PASCAL komutlarının verileri okuma, basit hesaplamalar yapma ve elde edilen neticeleri gönderme ile ilgili komutlarını anlatacağız.

## *Örnek Program*

PASCAL çalışmamıza başlamadan çok kısa bir program inceleyeceğiz. Bu programın detaylarını anlamak için uğraşmayın, bunu daha sonra açıklayacağız.




```
PROGRAM inchsant (input,cutput);

CONST
     santinch  =  2.54;
VAR
     inch,sant : real;
BEGIN
     riteln ('Inch cinsinden bir
     uzunluk girin');
     writeln ('Bu ',sant,'e eşittir.');
END.
```

```
  Inch cinsinden bir uzunluk girin.

30.0

Bu          7.620000E+01'e eşittir.
```

Bu program inch cinsinden verilen bir uzunluğu santimetre cinsine çevirmektedir. Inch cinsinden olan uzunluk INCH değişkeninde saklanmaktadır. Bunu yapan komut 'READLN (INCH);'dır.' SANT: = SANTINCH* INCH' komutu inch cinsinden uzunluğu 2.54 ile çarparak santimetre cinsine çevirmektedir.

## *Standard Belirteçler ve Ayrılmış Kelimeler (Standard Identifiers and Reserved Words)*

Değişik tipte elemanlar içerir: Ayrılmış kelimeler, standard belirteçler (reserved vords), özel semboller, ve veriler için isimler. Ayrılmış kelimelerin PASCAL'da özel bir anlamı vardır ve bunları başka amaçlar için kullanılmazlar.

Verilen örnek programdaki ayrılmış kelimeler şunlardır:

PROGRAM,
CONST, VAR, BEGIN, END.

Programdaki standard belirteçler ise şunlardır:

READLN, WRITELN,
REAL, INPUT, OUTPUT.

'=, *, :=' gibi işaretler de özel sembollerdir.

## *Belirteçleri Bildirme (Declaring Identifiers)*

PASCAL'a programımızda hangi değişkenleri kullanacağımızı belirtmemiz lazımdır. Bunu yapmak için bildirme komutlarından yararlanacağız. Bunlardan ilki CONST bildirme komutudur.

```
CONST
   santinch = 2.54;
```

SANTINCH değişkenini 2.54 sabit sayısına eşitler. Böylece 2.54 sabitinin adı olarak santinch'ı kullanabiliriz.

CONST komutu ile bildirilen değişkenlere sabit (''constants'') deriz. Programımızda hiçbir zaman değişmeyerek hep aynı değerini koruyacak olan değişkenleri CONST komutu ile bildirmemiz doğru olacaktır. Bir kere bir değişkeni CONST komutu ile bildirdiğinizde o değişkenin değerini program içinde değiştirmeye çalışmanız bir hata mesajı ile sonuçlanacaktır.

Commodore

## *Değişken Bildirimi (Variable Declaration)*

VAR
inch,sant: REAL;

Inch ve sant değişkenleri reel sayı olarak bildirdiğinizde, bu değişkenler programımızda istediğiniz sayısal değerleri alacaktır. Bu değişkenlerin alabilecekleri değerlere örnek :30.0, 561.578

VAR komutu ile bildirilen değişkenlere 'VARIABLES' (değişken) deriz. Değişkenler programda kullanıcıdan alacağınız veya programın çalışması sırasında elde edeceğiniz değerleri saklamanız için kullanılacaklardır. Değişken bildirimi (variable declaration) komutlarında kullanılan standart belirteçler (REAL, CHAR, INTEGER) PASCAL'a değişkende ne tip veri saklanacağını belirtirler. REAL, CHAR, INTEGER veri tiplerini ileride detaylı olarak anlatacağız.

Değişkenlerinize (variables) istediğiniz adları vermekte oldukça serbestsinizdir. Değişkenlere isim verirken şu kurallara uymak zorundasınız.

1. Bir değişken ismi her zaman bir harfle başlamalıdır.

2. Bir değişkende sadece harfler ve sayılar bulunmalıdır.

Değişkeninize PASCAL'da ayrılmış kelimelerin (reserved words) birini isim olarak veremezsiniz. Örnek olarak aşağıda bazı geçerli olan ve olmayan değişken isimleri verilmiştir.

Geçerli değişken isimleri :İSİM, SOYADI, TARİH

Geçersiz değişken isimleri :İSİM, const, VAR, İKİ*ÜÇ

Bir değişkenin ismini sınırlayan bir kural olmamasına rağmen bazı PASCAL COMPILER'ları sadece ilk sekiz karakteri kabul ederler. Böyle bir durumda TELEVİZYON1 ile TELEVİZYON2 gerçekte aynı değişkeni temsil etmektedirler. Bu durumda bu iki ismi aynı programda farklı değişkenler olarak bildirmek bir hata mesajına yolaçacaktır. ***J***

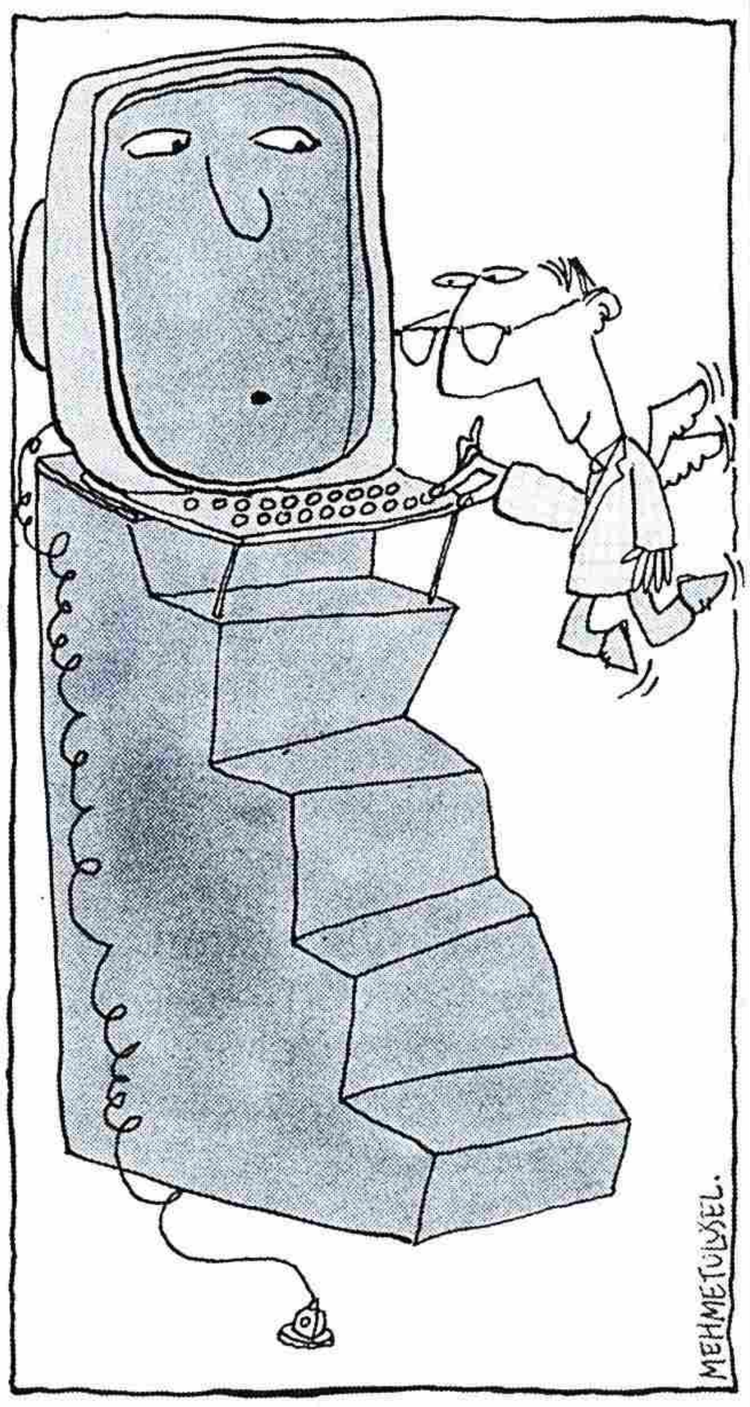
## *Belirteçlerin Sınıflandırılması*

Aslında ayrılmış kelimeler de (reserved words), standart belirteçler de, değişkenler de, belirteçler grubuna girmektedir. Bir PASCAL programındaki bütün belirteçler ya önceden tanımlanmış olmalı ya da programcı tarafından programda tanımlanmalıdırlar. Ayrılmış kelimeler ve standart belirteçler daha önce PASCAL dilinde tanımlanmışlardır. Sizin bu belirteçleri tekrar tanımlamanız gerekmez. Aşağıda belirteçlerin üç türüne de örnekler bulacaksınız.

## *Ayrılmış Kelimeler (Reserved Words) Standart Belirteçler Değişkenler*

program, var, const, begın, end
ınput, output, char santınch, ınch
real,readın,wrıteln
herf, ısım1

Aşağıda şimdiye kadar öğrenmiş olduğumuz komutları toplu halde göreceğiz.




## *Program Komutu*

PROGRAM prog-ismi
(INPUT, OUTPUT);

Açıklama :Programın ismi prog-ismi tarafından belirtilir. Giriş verileri INPUT adlı dosyadan okunacaktır. Çıkış verileri de OUTPUT adlı dosyaya yazılır. INPUT ve OUTPUT önceden tanımlanmış belirteçlerdir.

## *Constant (Sabit) Bildirimi*

CONST sabit = değer;

Açıklama :Değer, sabit belirtecine eşitlenmektedir. SABİT belirtecinin içeriğini programdaki herhangi bir komutla kesinlikle değiştirmemelisiniz. Mesela programınızda PI sayısını kullanacaksınız, bunu CONST bildirimini kullanarak bir belirtece eşitlemeniz doğru olacaktır. Çünkü bu değer programın bütün çalışma süresi içinde sabit kalmak zorundadır.

## *Variable (Değişken) Bildirimi*

VAR değişken-listesi :veri tipi;

Açıklama :Değişken listesindeki her değişken (belirteç) için hafızada bir yer ayrılmaktadır. Değişken listesinde değişkenleri birbirlerinden virgülle ayıracaksınız. Değişken listesinden hemen sonra ":" işaretini koyarak veri tiplerini (REAL, CHAR, INTEGER) belirteceksiniz.

Programlarınızda sizce anlamı olan değişken isimleri kullanmanız çok önemlidir, çünkü bu tür bir kullanım programınızı anlaşılır yapacaktır. Mesela MAAŞ bir kişinin ücretini belirten bir değişken için çok uygun bir ad olacaktır. Bunun yanısıra aynı değişkene S ismini verirseniz bunun bir kişinin maaşını temsil ettiğini sizden başka kimse anlamayacaktır. Bundan başka programınıza uzun bir süre sonra baktığınızda siz de S'ın neyi temsil ettiğini unutabilirsiniz.

Değişkenler için kullanacağınız isimlerin uzunluklarında bir kısıtlama yoktur. Fakat üç harften az kullanarak anlamlı isimler bulmak zor olur. Diğer tarafta eğer bir değişken ismi fazla uzunsa bunu programınızda her kullanmanız gerektiğinde doğru olarak yazmanız zor olacaktır. Bu durumda değişken isimlerinin uzunlukları üç ile on karakter arasında olması doğru olacaktır.

## *Atama Komutu*

netice := ifade

Açıklama :NETİCE tarafından belirlenen değişken, İFADE'deki işlemlerden elde edilen sonuca eşitlenir. NETİCE'yi taşıyan değişkenin daha evvelki değeri bu ifadenin sonunda

kaybolacaktır. İFADE ;tek bir değişkenden, tek bir sabitten veya değişkenler, sabitler ve aritmetik işlemlerden oluşabilir. Aritmetik işlemler şunlar olabilir:

| Aritmetik işlem | Anlamı |
|---|---|
| + | Toplama |
| — | Çıkarma |
| * | Çarpma |
| / | Bölme |

```
SANT := SANTINCH * INCH;
```

SANT değişkenine SANTINCH ve INCH değişkenlerinin değerlerinin çarpımı eşitlenmektedir. Bu işlem yapılmadan evvel SANTINCH ve INCH değişkenlerine geçerli değerler eşitlenmelidir., Bu da örnek programımızda yapılmıştır.

PASCAL'da şu tip bir atama komutu kullanmak mümkündür:

```
TOPLAM := TOPLAM + VERİ ;
```

Burada TOPLAM değişkeni ':=' işaretinin her iki yanında da bulunmaktadır. Bu elbette ki bir aritmetik işlem değildir., TOPLAM değişkeninin şu anki değerine VERİ değişkeninin değeri eklenerek tekrar TOPLAM değişkenine eşitlenir.




## *Readln Komutu*

Eğer bir bilgi ilk önce bilgisayarın hafızasına alınmazsa bilgisayar tarafından işlenemez. Bir veriyi bilgisayarın hafızasına yerleştirmenin üç yolu bulunmaktadır: Veriyi sabit olarak kullanma, veriyi bir değişkene yerleştirme, veya veriyi okuma (READ). İlk iki metodu kullanılacak değer program her çalıştırıldığında aynı olacaksa kullanılabilir. Eğer program her çalıştırıldığında değişik değerler kullanmak istiyorsak, o zaman verileri program çalışırken bilgisayara okumalıyız.

Örnek: READLN(ücret);

Programımız çalışması sırasında bu komuta geldiğinde sizin bir değer girmeniz için bekleyecektir. Bir değer girdikten sonra RETURN tuşuna basmalısınız.

readln (veri-listesi);

Açıklama: Veri listesinde yeralan bütün değişkenlere sıra ile girilen değerler eşitlenecektir. Veri listesindeki DEĞİŞKENLERİ ayırmak için virgül kullanacağız. Veri listesindeki her değişken için mutlaka bir değer girilmelidir. Verilerin giriliş sırası değişkenlerin sırasına uyması gerekmektedir. Sayısal değerleri girerken her bir değerin bitiminden sonra bir boşluk bırakılması gerekir. Değişkenin veri tipi karakter ise bilgisayara giriş yaparken boşluk bırakmanız gerekmez.

Şimdilik hoşçakalın. □


# ALFA SHOP

**Teleteknik Yetkili Satıcısı**

İstanbul Cad. Güven İşhanı No.36 Giriş Kat Tel: 583 40 93

- Her çeşit Commodore ve Spectrum prg. bulunur.
- Commodore 64/128 Frequenz veya Voltmetre olarak kullanmak istiyorsanız bize gelin
- Her ayın modül haline getirilir.
- Her türlü Hardware bulunur.

## Super Oyunlarımız (Kaset)

1) World Games
2) Tarzan
3) Silent Servis
4) Two on Two
5) Leader Bard Golf 3
6) Highlander
7) Ace of Aces
8) It's a Knockout
9) Movie monster
10) Super Cycle
11) Law of the west
12) Dragon's Lair (8 bölüm)
13) Stollone Cobra
14) Sigma 7
15) Footbolter of the Year
16) Red Max
17) Future Knight
18) Antiriad
19) Legend of Kage
20) Sky Runner

## Fiyatlarımız

| | |
|---|---|
| Kasette 1 oyun | -1000 |
| Kasette 1 oyun (özel kutuda) | -1200 |
| Disket ve kasette 1 prg. | -500 |
| 1 kasette 25 süper oyun | -5000 |
| 1 kasette 15 süper oyun | 3000 |

**Ödemeli olarak gönderilir**

Mart ayının 2 haftası ucuzluk haftası
1 oyun alana 1 oyun bedava

# Commodore 64 ile Grafik (5)

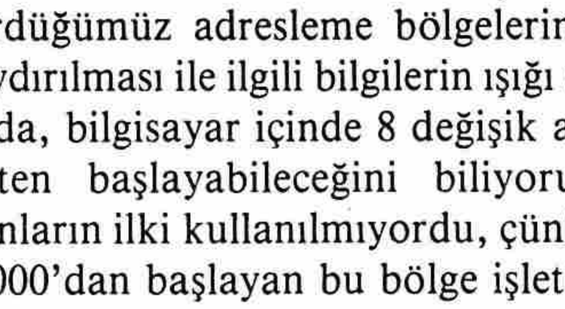
***Yazıya ilişkin programları Döküm Eki'nde bulabilirsiniz.***

**Tuna ERTEMALP**

*İşte geldik dizimizin en heyecanlı yerine. 320*200 tekrenkli ve 160*200 çokrenkli grafik kullanımına. Başlıyoruz...*

Daha önce de hafifçe değindiğimiz gibi, bu tür grafikte 320*200 = 64000 nokta kullanılır. Her nokta bir BİT olduğuna göre, toplam 64000 bit'e, yani 8000 byte'a ihtiyacımız var. Bu da yaklaşık 8K hafıza demektir. Bu hafızanın, daha önce gördüğümüz adresleme bölgelerinin kaydırılması ile ilgili bilgilerin ışığı altında, bilgisayar içinde 8 değişik adresten başlayabileceğini biliyoruz. Bunların ilki kullanılmıyordu, çünkü $0000'dan başlayan bu bölge işletim sistemi için çok önemliydi. Biz de diğer 7 bölge ile yetineceğiz.

Şimdi geldik işin en çetrefilli yerine. Bu grafik hafızasının organizasyonu oldukça karmaşık bir yapıya sahip. anlamak için lütfen gayret edin. Önce yanda verdiğimiz tabloya bakın.



Bunlar ne demek? Tablodaki noktaların her biri, grafik ekranın bir noktasına karşılık geliyor. Örneğin (0,0) koordinatları en sol üst köşedeki noktayı gösterir, (0,1) ise onun altındakini.

Gördüğünüz gibi noktalar 8*8'lik alanlara ayrılmış durumdalar. Bu size bir şey hatırlatıyor mu? Mesela, 8*8'lik karakter tanımlamalarını?.. Dolayısıyla yazı ekranın satır ve kolon yapısını... İşte tablomuzda o satır ve kolonları da görebiliyorsunuz.

Gene tabloda gördüğünüz yanyana duran 8'er noktanın oluşturduğu her bir grup, bir byte'a karşılık geliyor. Her bir byte'ın içindeki bit'lerin numaralanmasına dikkat! En soldakine 7, en sağdakine de 0 adı veriliyor.

Sıra geldi byte'ların dizilişine. Grafik ekranında her 8*8'lik grubun, yazı ekranındaki bir karakterlik bölge gibi düşünülebileceğini demin söyledik. İşte bu bölgelerin her biri, üstüste konmuş 8 byte'dan oluşur. Bu 8 byte'ın hafıza içindeki adresleri arka arkaya gelir. Sonra gelen 8 byte, bir sonra gelen 8*8'lik alanı tanımlar. Böylece 40 kolon da bitince, sıra alt satıra gelir. Bu arada grafik ekranının ilk 8 satırı da tamamlanmıştır. 9-16 arası grafik satırları gene 8*8'lik bloklar halinde 8'er byte tarafından tanımlanır.

Bunu bir örnekle görelim. Diyelim ki grafik ekranımızı $2000 (8192) adresinden başlattık. (0,0) koordinatı, 8192 + 0 = 8192 adresli byte'ın 7 numaralı bit'i olacaktır. (2,4) koordinatları, 8192 + 4 = 8196 adresli byte'ın 5 numaralı bit'i olacaktır.

Peki, (15,15) koordinatı ne olacak?

```
-------------------------------------------------------------
  !          !         !          !  111111!          !11112222!
  !  X -->   !01234567!          !83012345!          !67890123!...
  -------------------------------------------------------------
Y !          !KOLON 0  !          !KOLON 1  !          !KOLON 2  !...
  -------------------------------------------------------------
  ! !BIT'ler !76543210!          !76543210!          !76543210!...
-------------------------------------------------------------
0 !S!BYTE 0  !........!BYTE 8   !........!BYTE 16  !........!
1 !A!BYTE 1  !........!BYTE 9   !........!BYTE 17  !........!
2 !T!BYTE 2  !........!BYTE 10  !........!BYTE 18  !........!
3 !I!BYTE 3  !........!BYTE 11  !........!BYTE 19  !........!
4 !R!BYTE 4  !........!BYTE 12  !........!BYTE 20  !........!
5 ! !BYTE 5  !........!BYTE 13  !........!BYTE 21  !........!
6 !0!BYTE 6  !........!BYTE 14  !........!BYTE 22  !........!
7 ! !BYTE 7  !........!BYTE 15  !........!BYTE 23  !........!
-------------------------------------------------------------
8 !S!BYTE 320!........!BYTE 321!........!BYTE 329!........!
9 !A!BYTE 321!........!BYTE 322!........!BYTE 330!........!
10!T!BYTE 322!........!BYTE 323!........!BYTE 331!........!
11!I!BYTE 323!........!BYTE 324!........!BYTE 332!........!
12!R!BYTE 324!........!BYTE 325!........!BYTE 333!........!
13! !BYTE 325!........!BYTE 326!........!BYTE 334!........!
14!1!BYTE 326!........!BYTE 327!........!BYTE 335!........!
15! !BYTE 327!........!BYTE 328!........!BYTE 336!........!
-------------------------------------------------------------
. !.!   .   !        !   .   !        !   .
.   .     .                .                .
.   .     .                .                .
```

Tablomuzun X satırının ve Y kolonunun yardımıyla o noktayı bulun. 8192 + 328 = 8520 adresinin 0 numaralı bit'i olduğunu göreceksiniz. Hemen yanındaki (16,15) koordinatı ise, 8192 + 336 = 8528 adresinin Y numaralı bit'i oluyor.

Kâğıt üzerinde bulunması pek de zor olmayan bu adres ve bit'lerin, hesap yoluyla bulunması oldukça karmaşık. Birazdan grafik rutinlerini geliştirirken bu konuya geleceğiz.

Çokrenkli mod (multicolor) kullanıldığında byte'ların düzeni değişmiyor. Tek değişen, yanyana iki noktanın bir noktaymış gibi davranması. Örneğin (0,0) multicolor koordinatı 8192'nin 7. bit'i iken, (1,0) koordinatı 8192'nin 5. bit'i olacaktır. Çünkü 7. ve 6. bit'lerin ikisi de (0,0) koordinatları için kullanılacaktır. Bu iki bit'in içeriği %00, %01, %10 veya %11 olabileceğinden, o nokta dört değişik renkte olabilecektir.

Böylelikle renk konusuna girdik. Her iki modda da renk kullanımının nasıl olduğunu görelim...

320*200'ü kullanıyorsak, grafik ekranındaki her 8*8'lik bölgede iki değişik renk olabilir. Biri çizgi rengi, diğeri de arka plan rengi. Her bir 8*8'lik bölgenin bu iki rengi farklı farklı olabilir. Bu renklerle ilgili bilgi, VIDEORAM dediğimiz bölgede saklanır. Normalde yazı yazdığımız ekranı saklayan bu hafıza bölgesi, grafik ekranına geçtiğimizde bir renk deposu haline gelir. Bu hafızanın her byte'ındaki sayısal değer, karşılık geldiği 8*8'lik alanın rengini belirler. Bir alandaki çizgi rengini RC ile, arkaplan rengini de RE ile gösterirsek, VIDEORAM'da karşılık gelen byte'ın içeriğinin RC*16 + RE olması gerekir. Bir başka deyişle, o byte'ın üst 4 bit'i çizgi rengini, alt 4 bit'i de ekran rengini belirler.

VIDEORAM'ın organizasyonu daha basittir:

```
-------------------------
 Y ! X! 0-7 ! 8-15!16-23!....
-------------------------
0-7   ! 00  ! 01  !  02 !....
8-15  ! 40  ! 41  !  42 !....
16-23 ! 80  ! 81  !  82 !....
  .   ! .   ! .   !  .  !
  .   ! .   ! .   !  .  !
```

Gördüğünüz gibi her 8*8'lik grafik noktası grubuna karşılık gelen byte kolaylıkla hesaplanabilir. Örneğin VIDEORAM'ın başlangıç adresi $0400 (1024) ise (7,18) koordinatının renk byte'ı 1024 + 80 = 1104 adresinde olacaktır.

Multicolor'da ise, demin de değindiğimiz gibi her iki bit'in oluşturduğu çift, o noktanın rengini belirleyecektir:

---

% 00 = Renk 0 = VIC'ın 33 numaralı yazmacı
%01 = Renk 1 = VIDEORAM'ın alt dört bit'i
%10 = Renk 2 = VIDEORAM'ın üst dört bit'i
%11 = Renk 3 = Renk RAM'ı

---

Renk RAM'ı 55296-56295 adresleri arasında yer alır ve yeri değiştirilemez. Organizasyonu VIDEORAM ile aynıdır.

VIC'in 33 numaralı yazmaçı ise normalde yazı ekranının rengini belirler. Şimdi ise 0 numaralı renk yerine geçiyor.

Artık bu bilgileri değerlendirerek programlar yazmanın zamanı geldi. Nasıl grafik ekranı açılır / kapanır, renklendirilir, nokta çizilir / silinir, çizgi çizilir / silinir, elips çizilir / silinir? Tüm bu temel rutinleri birer altprogram halinde yazarsak, artık grafik konusunda oldukça büyük bir adım atmış oluruz. Çünkü, tüm cisimlerin temelinde yatan şekiller bunlardır ve bunları kullanarak diğer tüm geometrik çizimleri yapabilen altprogramlar yazabiliriz.

Önce bir iki noktada anlaşalım. Kullanacağımız grafik sayfası, BANK0'ın ikinci yarısı, yani $2000'den (8192) başlıyor. Burayı seçmemizin nedeni, bilgisayarın açıldığı sırada BANK0'a ayarlı olması. Eğer CIA2'nin bit'leri ile oynayarak diğer BANK'lardan birini kullanırsak bir sürü kaydırma işlemi yapmamız gerekecek. İsteyenler, daha önceki bölümlerde gördüğümüz POKE komutlarını kullanarak başka bir grafik sayfası kullanabilirler. Ancak dikkat etmeniz gereken bir nokta var. BANK2 ve BANK3'ü BASIC ile kullanamazsınız. Makina dili programlaması gerekir.

BANK0'ı kullanmamıza bağlı olarak, VİDEORAM da $400'den (1024) başlıyor.

Programlama dili olarak, herkesin anlayabilmesi için, BASIC kullanacağız.

Bunun yaratacağı en büyük sorun programların yavaşlığı olacak. Ancak makina dili kullanmak isteyenler ve programlamasını bilenler, bizim verdiğimiz rutinleri kolayca makina diline çevirebileceklerdir. Ayhan Kalaylıoğlu'nun makina dili dizisini okumaya devam edin. İlerde, grafik rutinleri de vermeye başlayabilir.

Grafik rutinlerimizi geliştirdikçe onları birleştirip bir program haline getireceğiz. Vereceğimiz satır numaraları 50000'den büyük olacak ve her bir parçası GOSUB ile çağrılabilecek. Gerekli olduğu zaman, bazı değişkenler parametre gibi kullanılacak. Örneğin XN = 100:YN = 20 verip nokta koyma rutinini GOSUB ile çağırdığımızda, 320*200 ekranının (100,20) koordinatına bir nokta konacak.

Bir başka gereklilik de, grafik işlemlerine geçmeden önce POKE 56,32 vererek BASIC kullanım alanının tavanını $2000 olarak belirlemeniz. Yoksa, BASIC programınız ile değişkenleriniz grafik sayfasının bilgileri ile silinebilir.

Artık başlayalım. İlk işimiz grafik sayfasını açmak.

320*200'lük ekrana geçmek için üç işlem yapmamız gerekiyor:

---

1) Yüksek çözünürlü grafiği aktif yapmak.
2) Multicolor'ı kapatmak.
3) Grafik sayfasının $2000'den başladığını belirtmek.

---

İlk işlemi yapmak için, VIC'in 17 numaralı yazmaçının 5. bit'ini yakmak gerekiyor. Multicolor'u kapatmak içinse 22. yazmaçın 4. bit'ini sıfırlamak lazım. Bir numaralı grafik sayfasını kullanmak için vereceğimiz POKE komutunu da evvelki sayımızda görmüştük.

160*200'lük ekrana geçmek içinse, birinci ve üçüncü işlemlerin yanında ikinci işlemin tam tersini yapmak, yani 22, yazmaçın 4. bit'ini yakmak gerekiyor.

Programımızın 50000-50060 arası satırları, 320*200'e geçmek için kullanılan komutları kapsar. 52000-52060 arası ise multicolor'a geçer. Sadece

GOSUB ile çağrılmaları yeterlidir. Parametreye gerek yok.

Sıra geldi grafik sayfamızı silmeye. Grafik sayfasını açar açmaz fark edeceksiniz, daha önceden hafızada kalan bir sürü garip byte değeri, ekranı bir nokta karmaşasına döndürecek. Bunları silmek için 8192'den başlayan bölgeye 8000 byte boyunca sıfır POKE etmemiz gerekiyor. Bu işlemi de, 50100-50140 arasındaki altprogram yapıyor. Parametresi yok.

Çizim yaptığımız renkleri belirlemek, kullandığımız çizim moduna göre değişiyor. Bu nedenle iki rutinimiz var.

320*200'de renklendirme işlemi bir tek FOR... NEXT içinde bitiyor. 1024'ten başlayıp 1000 byte boyunca devam eden VIDEORAM'a, çizgi ve ekran renginden hesaplayacağımız sayıyı POKE edince işimiz bitiyor. Bunu 50200-50240 arası satırlarda görüyorsunuz. Parametre olarak, RC içinde çizgi rengi, RE içinde de ekran rengi olması gerekiyor. Ekran, bu iki değişkene göre renklendiriliyor.

160*200'de ise, hem VIDEORAM doldurulacak, hem de 55296-56295 arasındaki renk RAM'ı. Bir de VIC'in 33. yazmaçı. Bu rutini de 52200-52250 arasında görüyorsunuz. Parametre olarak, R0, R1, R2 ve R3 içinde kullanılacak renkleri gönderiyorsunuz.

Grafik ekranıyla ilgili son işlem, işimiz bittiğinde yazı ekranına geri dönmek. Grafik sayfasını kapatmak için yapılacak işlem, açmak için yaptığımız işlemin tam tersi. Önce VIC'in 17. yazmaçının 5. bit'ini söndüreceğiz. Sonra 22. yazmaçta multicolor kapatılacak. Ve nihayet, ekranı $2000'e yerleştirirken değişen karakter üreteci adresini düzeltip $1000'e getiriyoruz. Grafik ekranı ile karakter üreteci adresi arasındaki bu ilişkiye geçen sayımızda dikkatinizi çekmiştik. Tüm bu işlemler, 50300-50360 arası satırlarda yapılmaktadır. Parametre gerekmiyor. Hem multicolor hem de normal ekran için bu altprogram kullanılabilir. Bu altprogram GOSUB ile çağrıldıktan sonra yazı ekranının silinmesinde fayda var. Tüm ekran, kullanılan çizim renklerine göre baştan aşağı karakterlerle dolmuş olacaktır.

Şimdi nokta, çizgi ve elips gibi temel şekilleri çizmeye başlıyoruz. Önce nokta...

Grafik sayfamız soldan sağa 320, yukarıdan aşağı 200 noktadan oluşuyor. Multicolor da kullansak, soldan sağa 320 tane bit'i kontrol etmemiz gerekiyor. Tek fark, bu sefer onları ikişer ikişer kontrol etmemiz gerektiği.

Sağdan sola noktalar, 0-319 olarak, yukarıdan aşağıya da 0-199 olarak numaralandırılırlar. Bizim yazacağımız rutin, bu koordinatlar verildiğinde, hangi byte'ın hangi bit'inin yakılması gerektiğini bulmalı. Peki, bu iş nasıl olacak? 50400-50460 arasındaki rutin, (XN, YN) değişkenleri ile gönderilen noktayı basmakta. Şimdi bu işi nasıl yaptığını görelim.

Öncelikle basılmak istenen koordinatın ekranda olup olmadığını kontrol ediyoruz. Eğer değilse, ana programa doğrudan geri dönülüyor. Aksi durumda byte ve bit hesabı yapılıyor.

Y-koordinatından başlayalım. Her ekran satırında 8 grafiksatırı olduğuna göre (grafik ekranının organizasyonunu hatırlayın), INT (YN/8) ile hangi ekran satırında olduğumuzu bulabiliriz. Her ekran satırı 320 byte'dan oluştuğuna göre, 320*INT (YN/8) ile noktamızın yer aldığı ekran satırının ilk byte'nin grafik ekranının başlangıç adresinden uzaklığını buluruz. Sıra geldi, noktayı bulunduran grafik satırının ilk byte'nın adresini bulmaya. Demin bulduğumuz sayıya (YN AND 7) ile bulacağımız YN/8 işleminin kalanını eklersek, grafik satırının başını buluruz.

Şimdi de XN koordinatını kullanarak, noktanın sol taraftan kaç byte ve kaç bit uzakta olduğunu bulalım. INT (XN/8) ile sola olan uzaklığı byte cinsinden buluruz. Adres olarak hesabı yapmak için, bu sayıyı 8 ile çarparız. Hatırlarsanız, yatay bir çizgi üstünde byte'ların adresleri sekizer sekizer artıyordu. Daha önce YN değerini kullanarak bulduğumuz byte değerine son bulduğumuz değeri de eklersek, noktamızın kaç numaralı byte içinde olduğunu saptamış oluruz. Sıra geldi bit numarasına. YN ile yaptığımız gibi (XN AND 7) yaparak XN'in byte içindeki yerini soldan itibaren buluruz. Bunu 7'den çıkarırsak, bit'in sağdan itibaren olan numarasını hesaplamış oluruz. İki üssü bit numarası ile, bulduğumuz adresin eski içeriğini OR'larsak, hedeflediğimiz nokta yanık duruma geçer. Tüm bu işlem 320*200 için geçerli. Yanan noktanın rengi de, daha önce ekranı renklendirirken verdiğiniz çizgi rengi olacaktır.

Multicolor ekranda da buna benzer bir işlem yapılır. 52400-52460 arası satırlarda görmektesiniz. Parametreler gene XN ve YN'dir. Bir de hangi renk ile çizim yapılacağını belirleyen RR değişkenini göndermelisiniz. İçeriği 0-3 arası bir sayı olmalıdır. Böylece, ekranı renklendirirken belirlediğiniz 4 renkten hangisinin nokta konurken kullanılacağı belirlenir.

Bu kez,tüm hesaplamalarda XN yerine XN*2 kullanıyoruz. Diyelim ki XN = 20. Multicolor'da bu X değeri, soldan 40 ve 41 numaralı bit'lere hitap eder. Biz de hesaplamalarımızda bunu gözönüne alıyoruz. Tüm işlemler sonucunda byte'ın adresini bulduktan sonra, o noktamızı oluşturan iki bit'in hangileri olduğunu hesaplıyor, sonra bunları AND ile sıfırlayıp, RR'nin gösterdiği değer ile OR'luyoruz. Tabii bunu yapmak için de bazı ufak hesaplamalar gerekiyor.

Bir de nokta silme işlemi var. Bu işlemi sadece 320*200 içindüşünebiliriz. Multicolor'da nokta silmek anlamsızdır. Dört renkten herhangi biriyle çizim yapılınca eski nokta ortadan kalkar, yerine yeni bir nokta gelir.

Nokta silmenin nokta koymaktan pek bir farkı yok. Tek farkı, OR işlemi için kullandığımız sayının tam tersini (yani 255'ten çıkarılmış halini) byte'ın eski içeriği ile AND'liyoruz. Böylece hedeflediğimiz nokta sönüyor. Bu işlemi de, 50500-50560 arası yapıyor ve (XN, YN) parametrelerini kullanılıyor.

Geriye çizgi ve elips kaldı. Çizgi ile devam edelim.

Programımızın 50800-51040 arası, 320*200'lük ekrana, başlangıç ve bitiş koordinatları X1, Y1, X2, Y2 ile belirlenmiş bir çizgi çizer. Bu koordinatlar ekranın dışında da yer alabilir. O zaman çizginin ekranda kalan kısmı çizilecektir.

Ortaokulda geometri dersini görenlerin hatırlayacağı meşhur bir düz çizgi formülü vardır:

Y = MX + T

M ve T harfleri değişik de olsa, formül budur. Şimdi bu formülü nasıl kullanacağımız görelim.

X1 ve Y1 değerlerini yerlerine koyar ve denklemi T'ye göre ayarlarsak şu formülü elde ederiz:

T = Y1-M*X1

aynı işlemi X2 ve Y2 ile de yapalım:

T = Y2-M*X2

Her iki denklem de T'ye eşit olduğuna göre,

Y1-M*X1 = Y2-M*X2

diyebiliriz. Ve bu denklemi de M'ye göre çizersek,

M = (Y2-Y1) / (X2-X1)

elde ederiz. Bu bize doğrumuzun eğimini verir. Sıra geldi T'yi bulmaya.

T = Y2-M*X2 olduğuna göre, M'yi yerine oturtarak

T = Y2-(Y2-Y1) / (X2-X1)*X2

de diyebiliriz. Bu T ve M değerlerini Y = MX + T denklemine yerleştirir ve Y'ye göre çözersek aradığımız formülü buluruz:

Y = (X-X2)*(Y2-Y1)/(X2-X1) + Y2

Tüm çizimimiz bu formüle dayalı. Tek sorun, çizgimiz aşağı doğru olduğu zaman çıkacaktır. O zaman X1 = X2 olacağından, X2-X1'in onucu sıfır olacaktır. Hiçbir sayı sıfıra bölünemeyeceği için, bilgisayar hata mesajı verip duracaktır. Bunu da bir iki IF ile çözümlüyoruz.

Programa bir gözatalım. Önce X ve Y koordinatları en yakın tamsayılara yuvarlanıyor. Sonra da, çizginin tamamen ekran dışında olup olmadığı kontrol ediliyor. Eğer öyleyse geri dönülüyor. Değilse gerekli X ve Y farkları hesaplanıyor. Eğer X1-X2 sıfır değilse, normal çizgi çizme kısmına atlanıyor. Eğer sıfırsa, çizgimiz bir dik çizgi demektir. Bu durumda önce çizginin ekran içine sığıp sığmadığı kontrol ediliyor ve eğer sığmıyorsa başlangıç ve bitiş koordinatları ekran içinde kalacak şekilde yeniden hesaplanıyor. Ardından da çizgi çiziliyor.

Eğer dik çizgi değilse, gene çizginin ekran içine sığıp sığmadığı kontrol ediliyor, sğmıyorsa başlangıç ve bitiş koordinatları ekran içinde kalacak şekilde yeniden hesaplanıyor ve çizgi çiziliyor. Bu arada çizgi çizmek için hep nokta koyma rutini çağırılıyor.

51100-51340 arasındaki çizgi silmenin, çizgi çizmekten hiçbir farkı yok. Sadece nokta koyma yerine nokta silme rutini çağırılıyor.

Multicolor'da çizgi çizmek de aynı işlemleri gerektiriyor. Farkı, maksimum X değeri olarak 159'u kabul ediyor ve çokrenkli nokta koyma rutinini çağırıyor. Bu altprogramı da, 52800-53040 arasında bulabilirsiniz. (X1, Y1) ve (X2,Y2) koordinatlarının yanısıra RR içinde de 0-3 arası bir sayı ile hangi numaralı rengi kullanacağınızı belirtmelisiniz.

Son şeklimiz elips. Neden daire değil de elips? Daire, X ve Y yönündeki yarıçapları eşit olan bir elipstir. Dolayısıyla elips için bir rutin yazdığınızda, otomatik olarak daire çizmeniz de sağlanmış olur.

Elips içinde üç altprogramımız var. Elips çizmek, silmek ve multicolor'da elips çizmek için.

Elipsin geometrik denklemi şu şekildedir:

Y = B*SQR (1-X 2/A 2)

Eğer dikkat ederseniz, bir elipsin sağ üst, sağ alt, sol üst ve sol alt parçalarının birbirlerine simetrik olduğunu görürsünüz. Yani biz elipsin sadece bir çeyreğini hesaplayarak tüm elipsi çizebiliriz. Zaten yavaş olan BASIC programı için önemli bir kazanç.

50600—50670 arasında, XR ve YR ile iki doğrultudaki yarıçap uzunlukları belirlenmiş ve (XM, YM) ile merkezinin koordinatı verilmiş elipsi 320*200 ekranında çizen rutini görmektesiniz. Bu rutinde elipsin bir çeyreği hesaplanmakta ve yeni hesaplanan nokta ile eski nokta arasına çizgi çizilerek kapalı bir elips çizilmektedir. Eğer elips tamamen ekrana sığmıyorsa, sığan kısmı çizilecektir.

50700-50770 arasında aynı işlemler elips silmek için yapılıyor. Çizgi çizmek yerine çizgi silmek için kullanılan rutin çağırılmakta.

52600-52670 arası ise multicolor ekranda RR ile belirtilen renkte bir elips çizmekte. Tek farkı, cokrenkli çizgi çizme rutinini çağırması.

İşte bu kadar. Artık bu rutinleri kullanarak, hatta onları daha iyi bir hale getirerek kendi grafik programlarınızı yaratabilirsiniz. Gene de BASIC'in yavaş kalacağını, makina dilinin veya SIMONS' BASIC'in daha kullanışlı olacağını unutmayın. Gelecek aya kadar hoşçakalın...□


**UFO**

İstasyon Cad. Pelin Pasajı
Kat:2 No:35 Bakırköy-İSTANBUL

**572 13 17**

**İDDİA EDİYORUZ EN UCUZ UBİZİZ**

FREEZE FRAME 3: En güçlü Kop. Kartuşu.....**25.000**
5 Ek Fonksiyon ve Kısa
ICE MACHİNE : Yeni Bir Kırıcı Kartuş.....**25.000**
En Ucuz Bizde
FREEZE FRAME 1: Hâlâ aranan Kırıcı Kartuş.....**17.000**
GEOS İŞLETİM DİSKETİ: Özel Vers. 1+2.....**12.500**
MEMOSTAR Orijinal Utilite D. .....20.000
Gerçek Değerinin Beştebirine
UFO CR-2 : Kaset için Tek Süper Kopya Prog. ..... **5.000**
Bu Fiyata Son Fırsat
MANTHMASTER : Matematik Makinası ..... **5.000**
Çok Fonksiyonlu Matematik Pr.
GR-BASIC : Simon's Basıc'ten daha fazla.....**5.000**
Komut ve Fonksiyon. Basıc ile
Interrupt ve El Kitabı

**Normal oyun programları (Disk-Kaset)..... 500**
UFO-HR TURBO: Grafikle yükleyen Tek Turbo .....**25.000**
UFO-RES: UFO-HR için resim hazırlar.....**25.000**
KASETTE TEK OYUN..... **1.000**

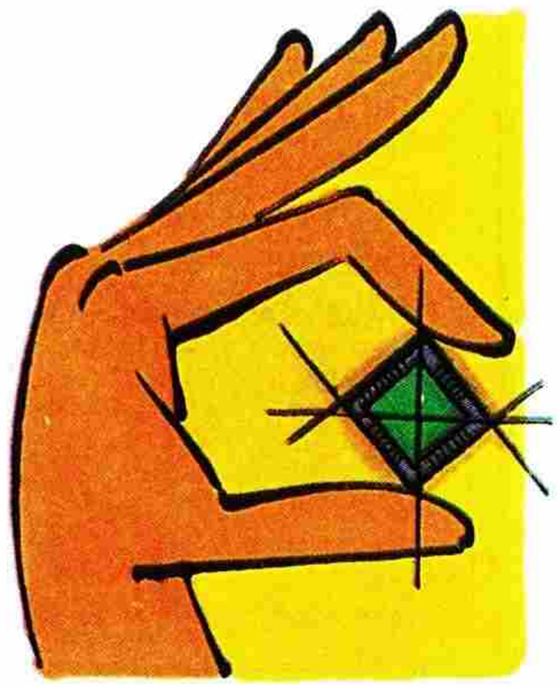
**PRATİK KOPYALAMA YÖNTEMLERİ**.....**3.000**

**Amatör merakını giderecek, beceri kazandıracak, piyasadaki program ve kopya kartuşlarını tanıtacak, tercihinizi kolaylaştıracaktır.**

**ÜCRETSİZ BROŞÜR-ÖDEMELİ SERVİS**

**KAFA AYAR MK 1**
**KOPYA-RESET CİHAZLARI**

UFO TD-1 Teypten Diske .....
Kopya Programı ..... **7.500**
1 KASETTE 20 PROGRAM ..... **5.000**
1 KASETTE 10 PROGRAM..... **3.000**



# Her İşin Başı Çekirdek 1

Seyfettin ÇANKAYA

Bilgisayarın bilgisayar olmasını sağlayan en önemli faktör bilgilerin eksiksiz ve sorunsuz bir program olarak çalışmasını sağlayan yazılımıdır. İşletim sistemleri bilgisayar adı verilen elektronik kütlenin ne yapması, nasıl yapması gerektiğini, o kütlenin anlayacağı dilde anlatan programlardır. İşletim sistemlerinin gelişmesini ve yaygınlaşmasını sağlayan mikro işlemcilerin en büyük özelliği, kendilerine özgü dilin kolay anlaşılabilir ve programlanabilir olmasıdır. C-64'ün kullandığı 6502 uyumlu 6510W bu alanda en popüler olanıdır.

C-64 işletim sistemi aslında yeni bir sistem değildir. Yeni C-64'ten önce tasarlanmış, denenmiş bir sistemdir. Commodore firmasının C-64'ten önce çıkardığı ve C-64'ten oldukça basit bir yapıya sahip VIC-20 tasarlanırken KERNAL adı verilen sistem kullanıldı. C-64 gelişmiş bir model olmasına karşılık bu sistemi olduğu gibi kullandı.

Bu nasıl gerçekleşti? Kernal sisteminde bir işletim sisteminin en önemli rutinleri, sıçrama tablosu (JUMP TABLE) olarak adlandırılan tabloyla adreslendirildi. Bu tablo $FF81'den $FFF6 adresine kadar uzanır.

Böylece yeni bir model tasarlanırken gelişme sadece bu işlevleri yerine getiren rutinde ve tablodaki adreslerde olmakta, tablonun konumu aynı kalmakta, bu da yeni birtasarımı oldukça kolaylaştırmaktadır. Bunun örneğini C-16 ve Plus-4 Kernal sisteminde de görebiliriz. Buna ilginç bir örnek VIC-20 C-64 C-16 ve Plus-4 programlarının çoğunun, çok az bir değişiklikle türünde kullanılabilir olmasıdır.

## Kernal Rutinleri Ne İşe Yarar?

Kernal tanıtılırken bir işletim sistemin en temel işlevlerini yerine getirir demiştik. Bu işlevler tüm sisteminin ilk açılışından kapanışına kadarki sürede kontrolü sağlamak Disk Drive, Dataset, Printer, Klavye, Video devresi ve RS-232 portuna bilgi gönderme, bilgi alma gibi işlemlerdir. Bütün bu işlevler sözünü ettiğimiz Kernal rutinleri ile belirlenmiştir.

Bu yazı dizisinde tüm Kernal rutinlerini inceleyeceğiz ve bu rutinler yardımıyla çalışan makine dili kullanılarak yazılan programlardan örnekler vereceğiz. Bu nedenle temel makine dili bilgisi gerekeceğinden, dergide çıkan bu konuya ilişkin yazıları incelemenizi öneririz.

## C-64 Power-On Kernal İşbaşında!..

C-64 ilk açıldığında ne olur dersiniz? Bunu Kernal içindeki bilgilerden öğrenelim. Kernal hexadecimal olarak $E000-$FFFF aralığında yer alır. Bunun nedeni C-64'ün işlemcisi olan 6510'un ilk çalışma adresini $FFFC'deki bellekten alıyor olmasıdır. Şimdi bu adreste ne var öğrenelim.

$FFFC=65532 olduğunu hatırlayarak

```
PRINT PEEK (655232)+256*
PEEK(65533)
```

Sonuç 64738 bulunur. Bu değer RESET vektörünü gösterir. Örnek olarak biz SYS 64738 yaptığımızda C-64 sanki yeni açıklıyormuş gibi davranır. Bu adres $FCE2'ye eşittir. Şimdi bu adresten incelemimize başlayalım.

```
.
FCE2  A2 FF      LDX #$FF
FCE4  78         SEI
FCE5  9A         TXS
FCE6  D8         CLD
FCE7  20 02 FD   JSR $FD02
FCEA  D0 03      BNE $FCEF
FCEC  6C 00 80   JMP ($8000)
FCEF  8E 16 D0   STX $D016
FCF2  20 A3 FD   JSR $FDA3
FCF5  20 50 FD   JSR $FD50
FCF8  20 15 FD   JSR $FD15
FCFB  20 5B FF   JSR $FF5B
FCFE  58         CLI
FCFF  6C 00 A0   JMP ($A000)
FD02  A2 05      LDX #$05
FD04  BD 0F FD   LDA $FD0F,X
FD07  DD 03 80   CMP $8003,X
FD0A  D0 03      BNE $FD0F
FD0C  CA         DEX
FD0D  D0 F5      BNE $FD04
FD0F  60         RTS
FD10  C3         ???
FD11  C2         ???
FD12  CD 38 30   CMP $3038
FD15  A2 30      LDX #$30
```

Yukarıdaki JSR $ FDD2 ile çalışan altprogram kartuş bağlantısı olup olmadığını test etmektedir. Bu işlemi de £FD10 adresinde yer alan CBM80 şifresinin kontrolü ile sağlamakta, kartuş bağlantısı varsa JMP ($8000) işlemiyle kartuştaki ROM üstünden bu adrese sıçramaktadır. Bazı programların reset ile tekrar çalışması da bu adresteki kodların CBM80 olmasıdır. Bu bağlantı olmadığı durumda JSR $FDA3 ile giriş/çıkış devrelerinin çalışması, JSR $FD50 ile RAM TESTİ VE Basic başlangıç

adresi ekran adresinin belirlenmesi JSR $FD15 ile giriş/çıkış rektörlerinin adreslere atanması, JSR$FF5B ile video kontrol devresinin çalışması sağlanır. Bu işlemlerin sonunda JMP ($A000) ile Basic ROM devreye girerek Basic çalışmaya geçilir.

## İlk Kernal Rutini

Şimdi Kernal rutinlerini incelemeye başlayalım. Bunlardan en basiti karakter çıkışını sağlayan CHROUT rutini. Kullanılışı oldukça basit; A kütüğü karakterin ASC kodunu içerir, JSR CHROUT ile bu karakter belirli bir çevre birime gönderilir. Burada bilgi ekran üzerine gönderilmektedir. Aşağıdaki örnekte MERHABA KERNAL yazısının ekrana çıkışını sağlayan bir program görülmektedir.

Bu programda X kütüğü mesajı indekslemek için kullanılır ve "*" karakterinde işlem sona erer. Bu programı bir monitör program yardımıyla yazıp deneyebilirsiniz. Yalnız adresleri siz belirleyiniz.

CHROUT rutini printer, disk driver, RS-232 gibi ünitelere de bilgi gönderir. İlerde buna ilişkin örnekler de vereceğiz.

```
10 ;ILK.KERNAL.RUTINI
20 CHROUT     EQU $FFD2
30 ;
40            LDX #$00 ;POINTER
50 DEVAM      LDA MESAJ,X
60            CMP '* ;*(BITIS ISARETI)
70            BEQ BITTI
80            JSR CHROUT
90            INX ;POINTER BIR ARTTI
100           JMP DEVAM ;ISLEME DEVAM ET
110 BITTI     RTS ; CIKIS
120 MESAJ     DFB 'MERHABA KERNAL*'
```

## İkinci Kernal Rutini: PLOT

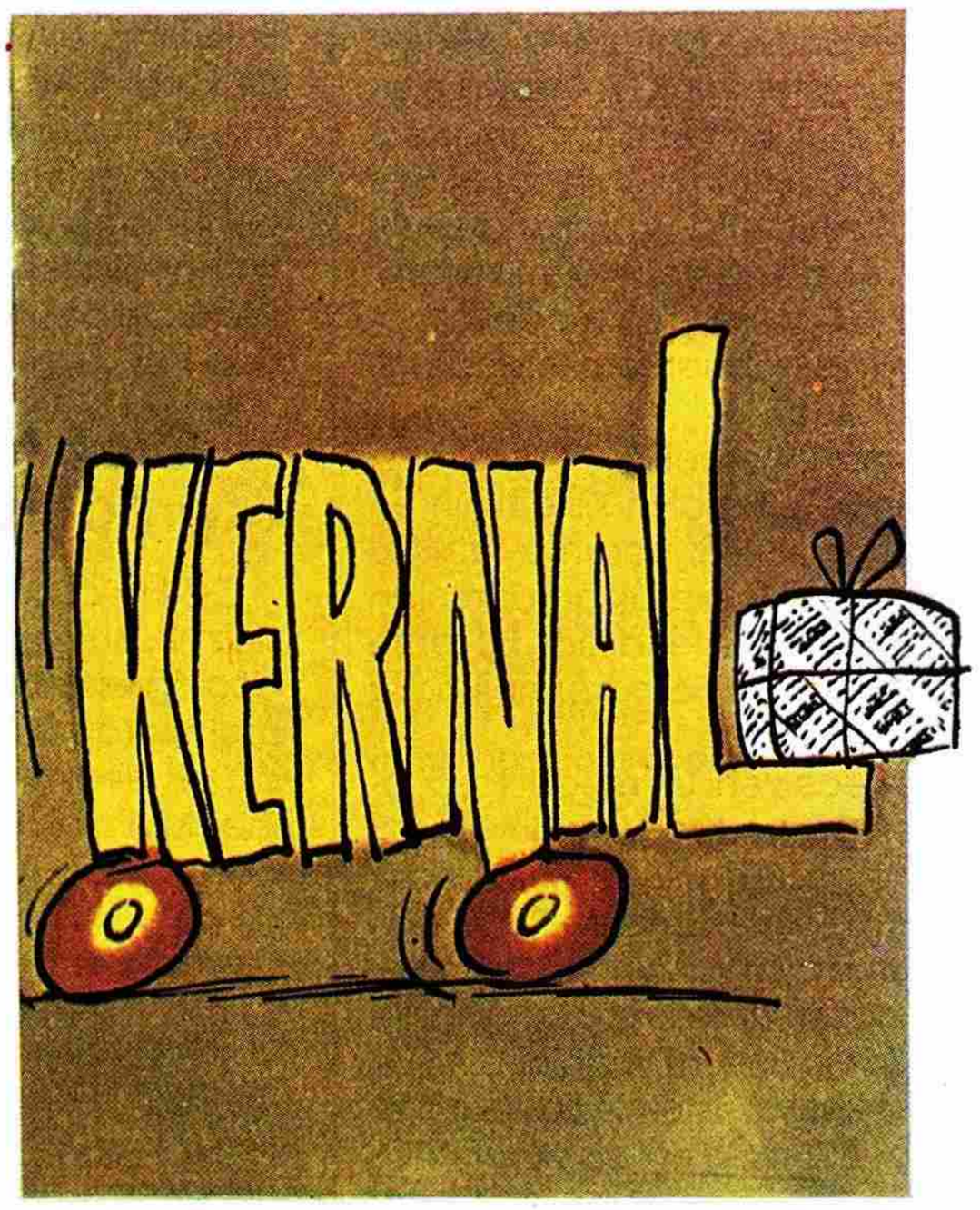
PLOT işemi bir an önce gördüğünüz CHROUT işlemi sırasında ekranın herhangi bir konumunu belirlemede kullanılır. Bu işlem iki ayrı modda çalışır. CLC (carry=0) olduğunda konumu belirler, SEC (carry=1) olduğunda da konumu okur. Bu rutinin tablodaki adresi $E50A'dır. İncelediğimizde yatay ve dikey konumların X, Y kütükleriyle belirlendiğini görürüz. Ayrıca Carry işlemine de dikkat ediniz.

```
.; E50A B0 07     BCS $E513
.; E50C 86 D6     STX $D6
.; E50E 84 D3     STY $D3
.; E510 20 6C E5  JSR $E56C
.; E513 A6 D6     LDX $D6
.; E515 A4 D3     LDY $D3
.; E517 60        RTS
```

Şimdi ise CHROUT ve PLOT rutinlerinin beraberce kullanıldığı bir program görelim.

```
 10 ;IKINCI.KERNAL.RUTINI
 20 ;
 30 CHROUT     EQU $FFD2
 40 PLOT       EQU $FFF0
 50            CLC ;KONUM YAZMA
 60            LDX #12 ;YATAY KONUM
 70            LDY #8  ;DIKEY KONUM
 80            JSR PLOT ;KERNAL RUTINI
 90 ;
100            LDX #$00 ;POINTER
110 DEVAM      LDA MESAJ,X
120            CMP '* ;*(BITIS ISARETI)
130            BEQ BITTI
140            JSR CHROUT ;KERNAL RUTINI
150            INX ;POINTER BIR ARTTI
160            JMP DEVAM ;ISLEME DEVAM ET
170 BITTI      RTS ; CIKIS
180 MESAJ      DFB 'MERHABA KERNAL*'
```

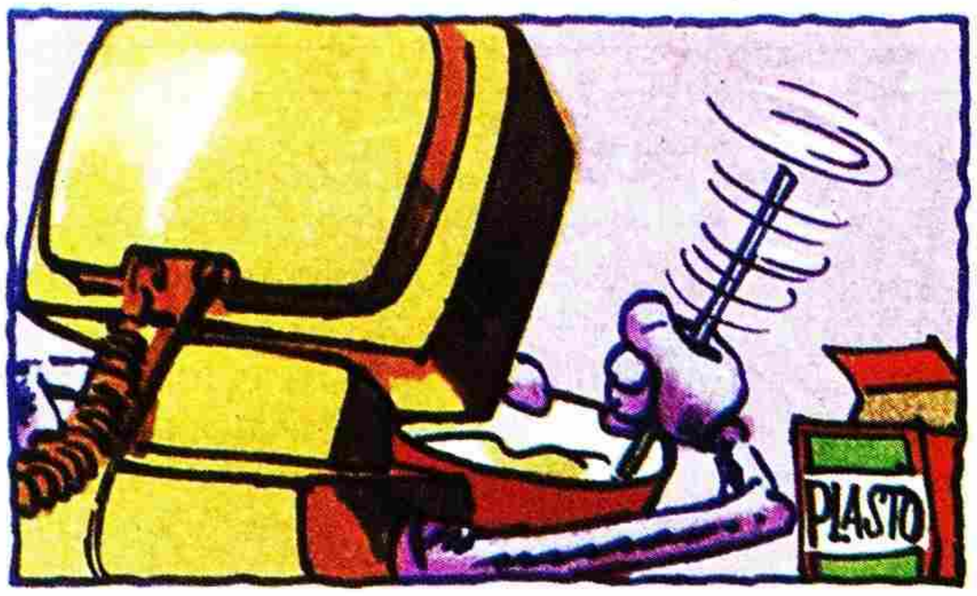
Bu programda carry=0 yapılarak konum belirleme moduna geçilmekte X ve Y kütükleri konumlara ilişkin bilgiyi taşımaktadır. PLOT işleminden sonra da bir önceki örnekteki mesaj ekrana yazılmaktadır.



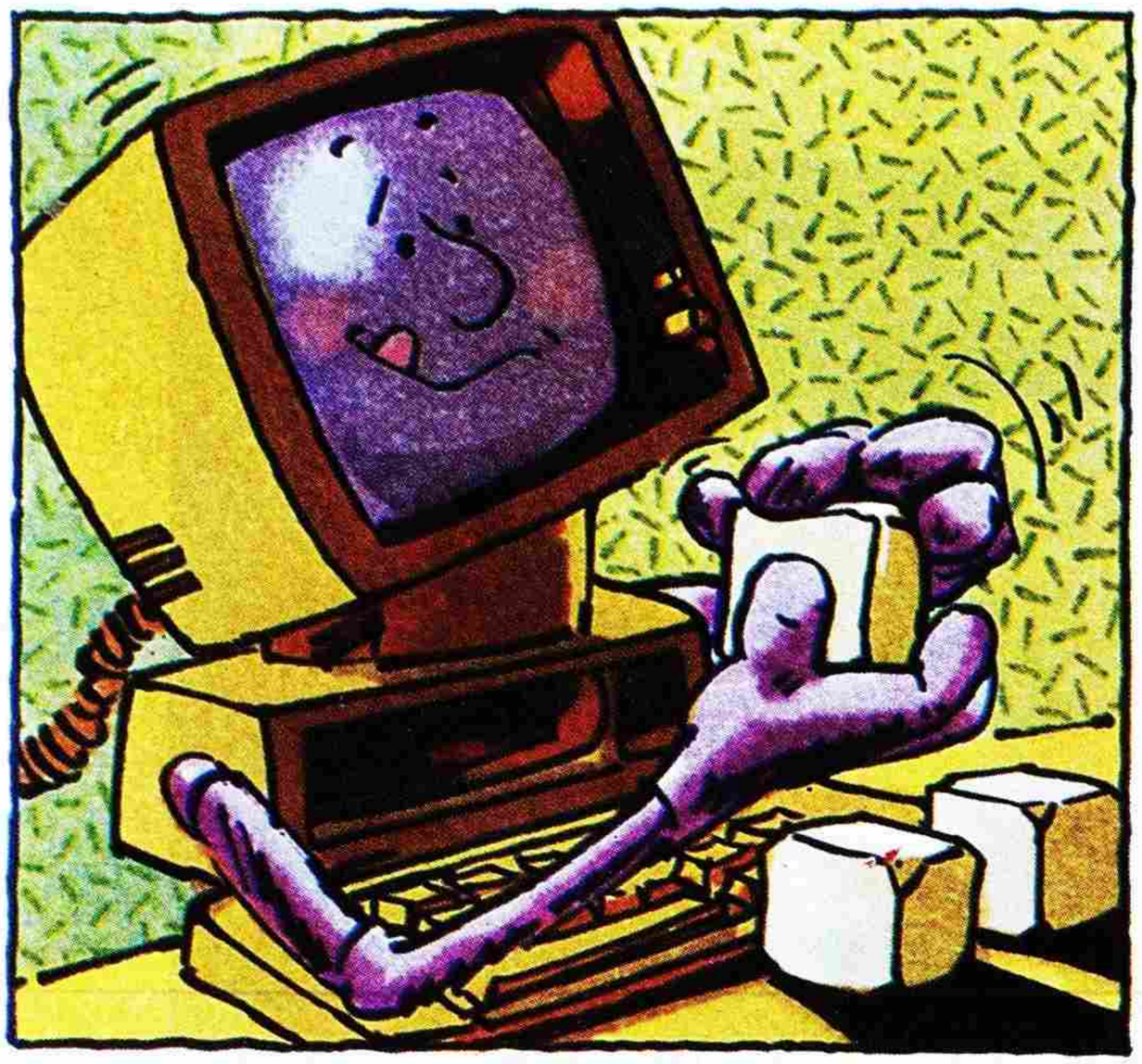


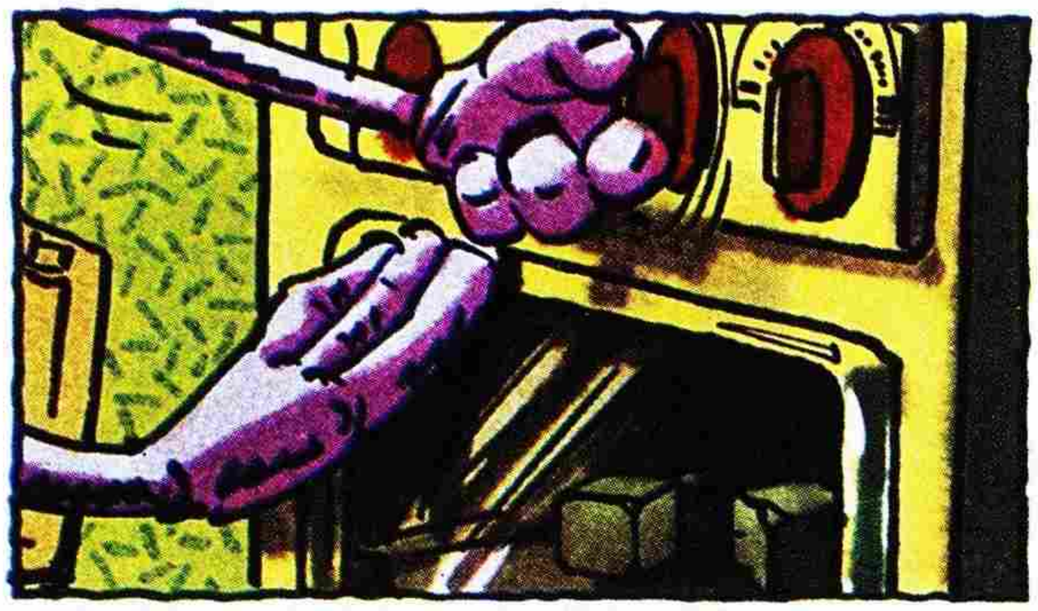

# 1801 MONİTÖR

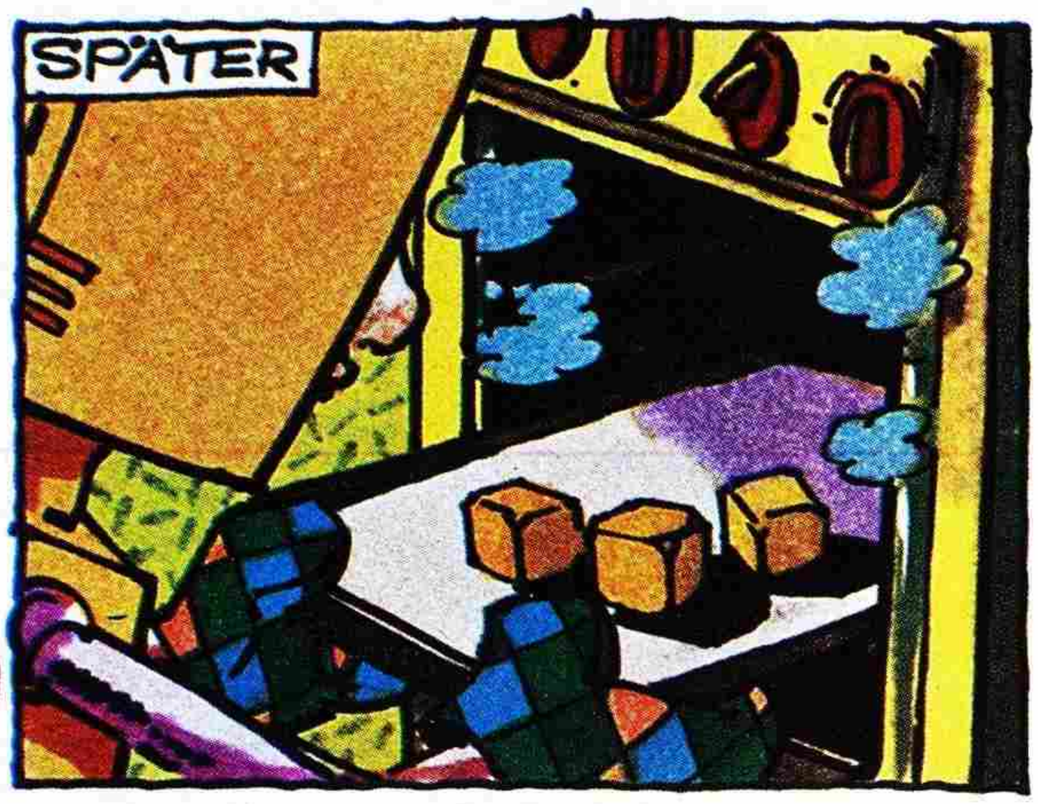
## TÜM BAYİLERDE

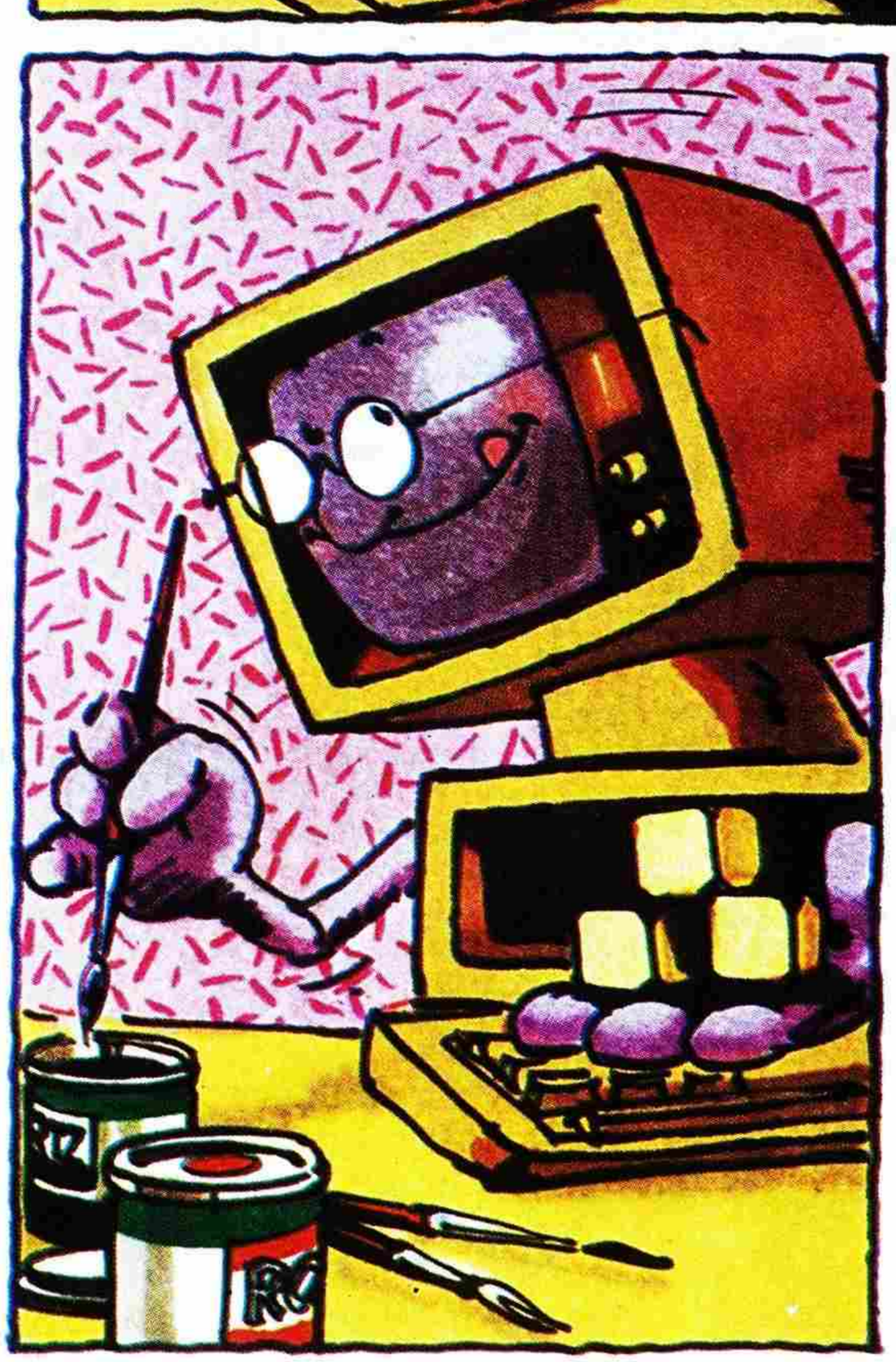

PLASTO








SPÄTER

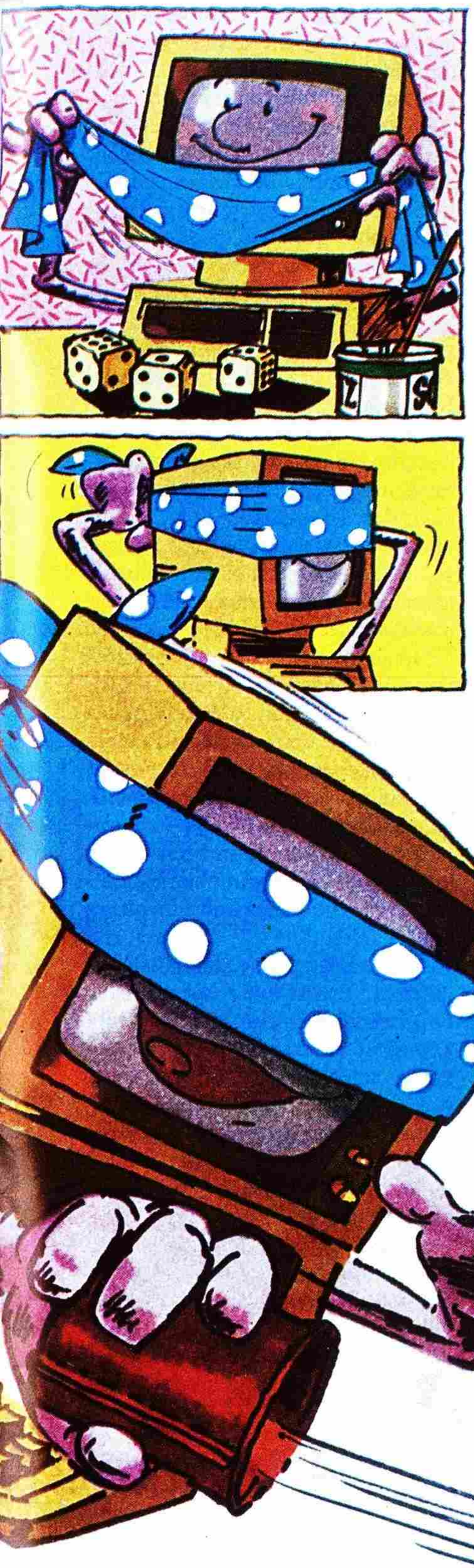





SCH



# Rastlantısal Sayıların Önemi




**Gernot SCHARMELI**

*Bilgisayar gibi böylesine kesinlik içinde çalışan bir makine herhangi bir şeyi rastlantısal olarak yapabilir mi? Bilgisayarların da rastlantısallığı sağlamak için zar attığını okuyunca şaşıracaksınız.*

Başka hiçbir olgu bizi rastlantı kadar şaşkınlık içersinde bırakamaz. Filozoflar ve müneccimler evrende rastlantıya yer olup olmadığını, her bir şeyin önceden belirlenip belirlenmediğini tartışadursunlar, diğer birtakım insanlar ise bu konuyu düşünmekten bile rahatsız oluyorlar.

Bir dizi rastlantının ardında yatan yasallıkları keşfetmeye çalışan matematikçi ve istatistikçiler ürkütücü bir kavram ve formüller dağarcığı geliştirmiş durumdalar. O kadar ki, en parlak ekonomi ve mühendislik öğrencileri bile, haftalık ders çizelgelerinde istatistik ya da olasılık öğretisi derslerini görünce cin çarpmışa dönmektedirler.

Yine de, pratikte bu karmaşıklığı dert etmeyiz, ister seçim sonuçlarının tahmini sözkonusu olsun, ister bir pazar araştırması, istatistikçilerden bilinçsiz bir kayıtsızlıkla bilgi edinmeye devam ediyoruz. Ama bu arada herhalde hiçbirimiz, istatistikçilerin bu başarılarının temelinde bilgisayarların "attığı zarların" yattığını aklımıza getirmemişizdir. Bu kısaca şöyle olmaktadır:

İstatikçilerin RND komutuna bağlı olarak, "rastlantısal sayı üreticisi" adı verilen bir aygıt, birtakım sayılar kurmaya başlar. İstatikçiler de bu sayıları bir tür örneklem olarak kabul ederler.

Ama bu saydıklarımız, buzdağının görünen ucundan ibaret. Örneğin, kendileriyle "rulet" oynanabilen bilgisayarlar aracılığıyla "rastlantı" olgusuna hesaplanabilirlik kazandırılması, parçacık fizikçileri gibi dakiklik peşinde koşan bilim adamlarının dev adımlar atmalarını sağlamıştır. Gene, bugün şehir trafiğine ilişkin bir planlama sözkonusu olduğunda, açılacak yeni bir hattın trafik akışını hızlandırıp hızlandırmayacağı "rastlantı bilgisayarları" sayesinde çabucak değerlendirilebilmektedir.

Aynı şekilde, işletmeler de, bünyelerinde bilgisayarlar ve terminallerden

oluşan bir sistem kurmadan önce, tasarladıkları sistemin boyutlarının elverişli olup olmadığını saptamak için bir bilgisayarın "zar oyunu"na başvurabilirler. En başarılı uygulamalardan biri ise şudur: Eğer bir bilgisayar şebekesine aktarılan veriler, daha önceden bir rastlantısal sayılar üreticisinin belirlediği şifrelerle kodlanırsa, bilgisayar korsanlarının işi bitmiş demektir.

Bunca işe yarayan rastlantılar alemine şöyle bir göz atmak bile büyüleyici olabilir. O halde gelin bu alemde bir gezintiye çıkalım. Okurlar bir yandan yazıları okurken bir yandan da rastlantısal sayıları yoklamak için kendi bilgisayarlarından yararlanabilirler. Yalnız uyaralım: Genellikle el kitaplarında RND fonksiyonuna ilişkin yapılan açıklamalar yetersizdir. Hepsinin ortak özelliği, şu öğüdü vermeleridir: "Kendinizi sürprizlere terkedin." Bu öğütle yapılabilecekler, önceden kestirilemeyen ve şaşırtıcı bir estetiğe sahip olan grafik çizimlerle, -eğer dilbilgisi kurallarını biraz biliyorsanız- çapraşıklığı insanı güldüren tuhaf cümlelerdir. Ama bütün bu uygulamalarda, birkaç denemeden sonra işin çekici bir yanı kalmaz.

Oysa, rastlantısal sayılarla yapabilecekleriniz, bu sayılarla ilgili bilgilerinizle doğru orantılıdır. Rastlantı ile de düzensizlik demek değildir. Eğer rastlantı her zaman "içsel bir düzenden yoksunluk" anlamına gelseydi, dünyada tek bir istatistikçi bile kalmazdı.

Aslında hepimizin bildiği şu kemik zar taneleri, rastlantının yasallıkla ilişkisi için gayet güzel bir örnek oluştururlar. İstatistik şunu göstermekte ve aynı zamanda kabul etmektedir: Onbinlerce zar atışı gözlemlendiğinde, eğer işin içinde bir hile yoksa, zarın altı yüzündeki sayılar da, uygulamada aynı sıklıkta düşeceklerdir. Ancak, uzun vadede geçerli olan bu yasanın tersine, tek bir zar atışı sözkonusu olduğunda, işe "bağımsızlık yasası" karışmaktadır. Şöyle ki, her tek zar atışına ondan önceki bütün atışların hiçbir etkisi olmaz. Altı yüzden herhangi birinin düşme olasılığı, her tek atış için her zaman altıda bir olarak kalmaktadır.

Ancak, bilgisayardaki rastlantısal sayılar üreticisi, bir zarı taklit etmeye çalışıyor da olsa, buradaki durum biraz farklıdır. Buradaki işlem, her biri farklı sayılar taşıyan, sözgelimi 999,999 bilyalı bir lotto oyununa benzer. "Üretici" (**generator**) bir kere çekmiş olduğu bilyayı, bu ikinci kere çekme şansına sahip değildir. Ancak, bilyalar üzerindeki sayıların 1,2,3,... şeklindeki tamsayılar değil, 0 ila 1 arasında yeralan ondalık sayılar olduğunu unutmayalım. Bu tür bir tercihin yararı açıktır: Böylelikle kullanıcı kendi gereksinmelerine bağlı olarak, rastlantısal sayıyı bir çarpan aracılığıyla dilediğince yükseltebilir.

El kitaplarında da verilen buraya kadarki bilgilerle bile bir şeyler yapılabilir. Rastlantısal sayılardan birincisinin X ekseni (yani yatay eksen) üzerindeki bir koordinatın değeri olduğunu düşünelim; ikincisi ise bir Y koordinatı değeri olsun. Bu durumda, her çift atış $1 \times 1$'lik bir düzlem üzerinde bir noktaya rastlantısal bir konum kazandıracaktır. Bunu bir "atış" olarak düşündüğümüzde, üretici (generator) tıpkı dikdörtgen bir hedef tahtasına rastgele atış yapan bir tüfek gibi davranıyor olacaktır.

Bu tür bir alıştırmanın pratik yararına gelincei: Uzun vade dikkate alındığında, diyelim ki bir milyon atıştan sonra, hedef tahtasının her tarafı aynı oranda isabet alacağından, örneğin 1 yarıçaplı dörtte bir daireye düşen isabetlerinin payının toplanması yeterli olacak ve böylece de II sayısının değeri zar atışı ile belirlenebilecektir; çünkü dörtte bir dairenin alanı II/4 $(1 \times 1 \times 3,14:4)$'tür.

"Zarların rastlantısı" anlaşılıyor anlaşılmasına da, bu rastlantı bilgisayara nasıl girmektedir acaba? Eğer bu sorunun yanıtını henüz bilmiyorsanız, o halde sıkı durmalısınız: Tüm milyon atıştan ancak ilki gerçekten rastlantısaldır. Gerisi ise, rastlantı gibi gözüküyorsa da, göz boyacılığından başka bir şey değildir.

"Üretici"ye ilk atış yaptırıldığında, üretici belli bir yazmaçtan (Register) ve o anda rastlantısal olarak orada bulunan bir sayıyı çeker. Bu, içinde zamanın mikrosaniye sayısı bulunan bir yazmaç olabileceği gibi, bazı bilgisayarlarda görüldüğü üzere program yapabilen bir yazmaç da olabilir. Daha sonra bilgisayar gerçekten rastlantısal olan bu sayıyı başlangıç değeri olarak, programlanmış bir rutine yerleştirir, ikinci "rastlantı"nın sayısını hesaplar, hesapladığı sayıyı alır ve bunu yeniden rutine yerleştirir; bu işlem, istenirse milyon defa yinelenebilir.

Sayılar bir algoritma aracılığıyla üretildiklerinden, ardarda yapılan atışlar birbirlerinden bağımsız değillerdir. Bundan dolayı, zarlar ya da rulet bilyaları gibi fiziksel üreticilerin tersine, bu sayılar "sözde rastlantısal sayılar" olarak adlandırılmışlardır.

Pek çok BASIC versiyonunda bilgisayarın başlangıç değerini kaydetmesi bir komutla sağlanabilmektedir. Bu durumda bilgisayar, bütünüyle aynı sayılardan oluşan ve sıra ve düzen bakımından da birincisine tam tamına denk düşen ikinci bir zar serisi oluşturabilmektedir. Buna, somut olarak, her başlangıçta AND-fonksiyonunun aynı ve eksi bir parametre ile harekete geçirilmesi sayesinde ulaşılabilmektedir.

Hangi "rastlantı" sonuçlarının ortaya çıkacağının saptanabiliyor olması rastlantı kavramının anlamı düşünüldüğünde, ne denli hayal kırıklığına uğratıcı ise de, pratik bir değer taşımaktadır. Yer darlığı nedeniyle milyonlarca sayının RAM-deposunda depolanma imkânı bulunmamaktadır; bunların algoritmanın "zekâsında" saklanması çok daha kolay bir depolama yoludur.

Atışlar bağımsız ve rastlantısal değillerse, "üretici"nin kullanılmasına ne gerek vardır? diye sorulabilir. Bu konuda sözü saat zamanına bırakmak her zaman için daha doğru olmaz mıydı? Bu elbette mümkündür. Bunun için RND-fonksiyonu harekete geçirildiğinde "Argument"i 0 yapmak yeterlidir. Fakat gene de, bu yöntem birçok durumda soruna bir çözüm getirmeyecektir. Bir düşünülecek olursa: Bir program şeridi sık sık akıp geçecek ve her defasında bir rastlantısal sayı buna eşlik edecektir. Ama bilgisayar, belki bir program şeridini işlerken hep aynı zamana gereksinme duy-



duğundan, bu durum, sayıların sistematik olarak farklılık göstermelerine ya da tümünün de eşit olmasına yol açacaktır ki, bu da hiç katlanılacak bir durum değildir.

Algoritmalarla verilen rastlantısal sayılara gelince, burada durum daha başkadır. Gerçi bunlar bağımsız değillerdir, ama özelliklerinin sınanması amacıyla bir istatistikçiye sunulurlarsa ve bu arada sayıların nereden çıktıkları kendisine bildirilmediği takdirde, istatistikçi buradaki rastlantısal olmama durumunu kanıtlayamayacaktır. Daha önce sözü geçen hedef tahtası örneğinde, hedef tahtası $10 \times 10$ alana bölünmüştür. Tek tek her alandaki isabetli vuruş sayıları aracılığıyla buradaki eşit dağılımın yetkinliği konusunda ilk görsel izlenimi edinmek mümkündür.

İstatistikçiler burada anlamlı isabet sıklıklarının hangi alanlarda toplanmış olduklarını bulmak için bir dizi sınama yöntemi uygulayabilirler. Alanları ikişer ikişer, üçer üçer ya da herhangi bir gruplama tarzında gözlemleyebilirler. Sonuçta ise, sayı dizilerinde bir sistematiklik kanıtlanamamaktadır.

Elbette "rastlantı" sayılarının bu düzeyde olanlarını sıradan bir algoritma ile elde etmek mümkün değildir. Bunun için bilgisayarın ortaya çıkışından bu yana çok değişik hesaplama yöntemleri düşünülmüş ve denenmiştir. Uzun bir süre için, bir sayının karesini almak ve bu işlem sonunda çıkan sayının orta kısmını oluşturan sayıyı "yeni sayı" olarak kabul etmek için yaygın bir yöntem olmuştu (Örneğin, $45086273 \times 45086273 = 2032$ **7720130303** 0529. Buradaki altı çizili sayı elde edilen yeni sayıdır). Ama bu yöntemin istatistiksel açıdan yetersizlikleri vardır.

Bugün kendini kabul ettirmiş olan algoritma "Modulo" yöntemidir. Bu yöntemin özü şudur: A ve P gibi iki tamsayı değişmez alınır, tamsayı bir başlangıç değeri seçilir, bu değer A ile çarpılır, çıkan sonuç da P'ye bölünür; bölme işlemi sonucunda arta kalan Z sayısı "yeni sayı" olarak kabul edilmektedir. Bununla da yeni bir süreç başlatılabilir. "Sözde rastlantı"nın gizi işte bu denli açıktır.

RND-fonksiyonunu BASIC'te uygulamak istediğimizde, bütün bu sayıların seçimiyle uğraşmak zorunda değiliz. A ve P değişmezlerinin "üretici" üzerine etkisi konusunda pek çok bilimsel yapıt mevcuttur. Burada özellikle ilginç olan nokta şudur: A ve P, her ikisi de asal sayı olmak zorundadır, yani örneğin 7 sayısı gibi, yalnızca kendi kendisiyle bölünebilen birer sayı. (Ayrıca da, A P'den daha küçük olmalıdır: $A < P$) O zaman "üretici" kendini tekrara kaptırmaksızın miktar olarak en çok düzeyde rastlantısal sayıya ulaşabilir. P'ye bölünme işlemi sonucunda arta kalan sayı ancak 1,2,3,...p-1 olabileceğinden, üretici de, kendi kendini tekrarlamadan önce, ancak P-1 sayılarını fırlatıp atabilmektedir.

O halde, "üretici"ye sokulabilecek P sayısı ne denli büyükse, onun "period"u da o denli büyük olmaktadır.

BASIC-versiyonlarında "period"lar yaklaşık 1 milyon kadardır. Basit kullanımlar için bu sayı tam anlamıyla yeterlidir. Daha masraflı projeler için, IBM'in büyük bilgisayarları 64 Bit'lik veri sözcükleri çerçevesinde $2^{31}$ üstü bir P sayısını barındıracak bir mekan sunmaktadır. Bunun anlamıysa şudur: Bir "period"da 2 milyar rastlantısal sayı bulunmaktadır.

Çok üst düzeyde teknik-bilimsel problemlerin ele alındığı kurumlarda kullanılan ve büyük bilgisayarların kralları sayılan Cray ya da Cyber 205 ise, P-değerleri için 46 Bit kullanabilirler ki, bu da $P = 2^{46}$ olarak tam sayımına 70 trilyon rastlantısal sayılı bir "period" demektir.

İnsanın aklına bunun yeterli olduğu geliyor değil mi? Jülich'deki Nükleer Araştırma Kurumu'nun (KFA) fizikçileri aynı görüşte değiller. Bu muazzam sayılara rağmen, "üretici"nin verim yeteneğinin sınırlarına çarpmaktadır bu kişiler. Orada sözkonusu olan şey nedir? Fizikçiler modelleştirmeler yoluyla bilgisayara, örneğin belli bir derecenin üstünde ısıtıldığında sürekli manyetik gücünü birdenbire yitiren demir gibi bir cevherin hareketini taklit ettirirler (simulation). "Bir evreden başka birine geçiş" olgusu fiziğin en önemli problemlerinden biridir.

Atomların birbirlerine etkileri konusunda tasarımlarımız ancak bir model çerçevesinde mümkün olabilirken, tek tek atomların istatistiksel hareketlerine ilişkin olarak bildiklerimiz daha ayrıntılıdır. Bilgisayarın bu konuda yapması gereken şeye gelince, şu söylenebilir: Bilgisayar, bu atomların rastlantısal bir davranışını zar atarak belirler. Bilim adamları için de atomların birbirlerine etkileri konusunda geliştirdikleri kendi model kuramlarını güvenilir bir tarzda hesaplamak ve sınamak ancak bu sayede mümkün olmuştur.

"Rastlantı üreticisi"nin gücünün çabuk tükenmesi ise şundan dolayı olmaktadır: "Üretici" on binlerce atom için binlerce enerji ve salınım durumunu zar atarak belirlemek ve bütün bunları binlerce zaman aralığı içinde ardarda yapmak zorundadır. Buysa, hedef tahtamızın iki değil, milyarlarca boyutu varmış gibi bir durum yaratır. Ve nihayet tek bir zar atımıyla yaratılan durum ile de yetinmek mümkün değildir. Rastlantısallıkların açığa çıkmaları için bu oyunun yüzlerce ve binlerce kez yinelenmesi gerekmektedir. Bütün bunlar ise, tabiî ki "üretici"yi ikinci bir kez çalıştırmaksızın yapılmalıdır.

Bilim adamlarının burada bir sınıra gelip dayanmaları sadece "period" konusunda olmamaktadır. Onlar istatistiksel elverişlilik açısından da eksiklikler keşfetmişlerdir. Bizler gibi sıradan tüketiciler için kaçınılmaz olsa



da tahammül edilebilir, bilim adamlarının gözünde ise tatsız eksikliklerdir bunlar. Aşırı büyük boyutlu "hedef tahtası"ndaki rastgele sayıların durumu şunu göstermektedir ki, "üretici" bazı alanları, hatta bazı dizileri tümden kendi "atış"larının dışında tutmaktadır.

Bu nedenle, bilim adamları görünüşte rastlantıyı daha yetkin bir tarzda üreten ve daha güçlü "period"lara imkân veren algoritmaların oluşturulması için gerçekten büyük bir tutkuyla çaba göstermektedirler. Bugün sayıları çok fazla olan "üretici"ler arasında "Kirkpatrickstoll" ya da "Tansworthe" gibi adlara rastladıklarında bu kelimeler bilim adamlarının kulaklarına tıpkı bir müzik gibi gelmektedir. Bu üreticilerin ardında en akla hayale gelmez yöntemler ve oyunlar saklıdır. Bunların sonucunda $2^{250}$ üstü bir "period", yani 60 haneli bir sayı doğmuştur.

Bu arada, zarların standart bir yöntem olan "Modulo" ile atılması sırasında da rastlantısal sayılarla ilgili bazı şeylerin manipülasyonu sözkonusu olabilmektedir. Eşit olarak dağılmış sayılara çok ender durumlarda gereksinme duyulmaktadır. Eğer bilgisayar hedef tahtasına eşit oranda "atış" yaptırılsaydı, bilgisayar böylelikle tuhaf bir atıcıyı taklit etmiş olurdu. İyi bir atıcının isabetleri merkezi alanlarda toplanır. Uzmanların deyişiyle, isabetler merkez yöresindeki "normal dağılımlar"dır.

Gene de, eşit dağılımlı RND-fonksiyonu, rastlantı için temel olarak kullanılabilmektedir. Bunu izleyen bir dönüştürüm programında -hedef tahtasını örnek almaya devam edersek- asıl boyutları $10 \times 10$ büyüklüğündeki alanların biçimleri değişmiş olacaktır: Merkezdekiler büyüyecek, kenardakiler ise küçüleceklerdir. Sonuçta alanlar tüm "atış"larla birlikte yeniden biçim değiştirecekler (yani rastlantısal sayılar değişmiş olacaklar) ve her şey son derece "normal dağılımlı" bir görünüm kazanacaktır.

Münih'teki Max-Planck Fizik Enstitüsü'nde elemantar parçacıkların avlanmasında yardımcı olan bir bilgisayar, değişik rastlantı dağılımlarının tüm bir çeşitlemesini depolamış durumdadır. Protonları nötronlarla bombardıman ettikten sonra araştırmacıları bu konuda en çok zorlayan soru "Hangi parçacıkların ne tür bir kesinlikle gözlemlendikleri" sorusudur.

Burada zar atarak belirlenen şeyler şunlardır: Parçacığın ölçümünde yapılacak bir hata, mümkün birçok çarpma reaksiyonundan hangisinin gerçekleşeceği; yani tek bir durumda şu ya da bu parçacığın ortaya çıkıp çıkmayacağı ve son olarak da parçacıkların çarpmadan sonra hangi açı altında dağılacakları- çünkü burada, örneğin bilardo oyununda olduğu gibi, hiçbir şeyi önceden hesaplamak mümkün değildir.

"Monte-Carlo" yöntemi ile rastlantısallıkları çoğu kez taklit eden bilgisayarın yaptığı iş fizikçiler için paha biçilmez bir değer taşır. Onlar ancak bilgisayar sayesinde, kendi yarattıkları karmaşık mekanizmanın ifade gücü hakkında güvenilir bir bilgi edinme imkânına kavuşurlar.

Öyleyse rastlantısal sayılar "üreticisi" daha çok doğa bilimlerinde ortaya çıkan problemlere mi çözüm getirmektedir? Hayır. Durum hızla değişmektedir. Bilgisayar gitgide ekonomide -özellikle de projelerin optimal planlanması alanında- büyük işler görmektedir.

Bankalar, büyük satış mağazaları ya da başka büyük işletmeler eğer bir bilgisayar şebekesi kurmak istiyorlarsa, buralarda çözüm aranılan sorunun "bekleme kuyrukları" olduğu söylenebilir. Burada sözkonusu olan sisteme bir merkezi bilgisayara bağlı uydu bilgisayarlardır ki, bunlar da kendi başlarına pek çok terminal kullanmaktadırlar. Soru şudur: Sistem, terminaldaki kullanıcının yanıt için normal durumda tahmin edilen iki saniyelik bekleme süresinden asla daha fazla beklemeyeceği şekilde mi kurulmuştur?

Acaba hangi terminale belirli bir zaman ayarı çerçevesinde el atılabilmektedir? Hangi sıklıkta ve hangi işler bu terminalde işlenmek durumundadır? Terminalin bilgisayara sadece bir veri yığını yolladığı ve kısa bir yanıt aldığı küçük sorular mı işlenmektedir? Yoksa birçok defalık bir bilgi alışverişini gerektiren daha büyük sorular mı? Yoksa işlemciyi (Processor) uzun süre işgal eden programlama işleri mi? Veyahut da bir disk sürücüsünü tekeline alan, ama işlemciyi bloke etmeyen büyük boyutlardaki veri yığınlarının nakli mi?

İşletme, tecrübî değerlere ve tahminlere dayanarak her terminal için anılan olayların olasılık derecelerini bilir. Veriler bilgisayara gelirler ve ona, eğer olayları zar atarak belirlemek istiyorsa, hangi tasarımlara göre rastlantıyı "yönlendirme"si gerektiğini bildirirler.

Böylece, her terminal için, ne zaman müdahale edileceği, bunun ne kadar süreceği, bir haberin kaç Byte kapsadığı, işlemcinin ne kadar hesaplama süresine gerek duyduğu, işin bir diskin müdahalesini zorunlu kılıp kılmadığı gibi şeyler bilgisayarın rastlantısal sayıları tarafından saptanmaktadır. Bu verilerin ışığında bilgisayar sistemin dayanıp dayanmayacağını çabucak araştırmaya koyulur. Çünkü bilgisayar tek tek bileşenlerin neler yapabileceklerini tabiî ki bilmektedir (işlemcinin hızını, diskin müdahale zamanını, hatların nakil kapasitesini). Rastlantısal yükleme durumlarına ilişkin tüm bu oyun her yönüyle araştırıldıktan sonra, bilgisayar en sonunda durum hakkında net bir tablo çizer: Planlama doğrudur ya da doğru değildir.

Eğer bugünlerde Frankfurt Havaalanı'ndaki akar banttan bavulunuzu alacak olursanız şunu da bir kere düşünün: Bu bant yapılmadan önce, uçakların kalkış saatlerinden önce ya da iniş saatlerinden sonra ne kadar yolcunun hangi zamanlarda kalabalıklar oluşturabileceğini bir bilgisayar zar atarak belirlemişti.

Eğer bugün veya yarın otoyollarda 100/Km-hız sınırı getirilmesinin anlamı ve etkisi kanıtlanmak istenirse, bir yığın rastlantı üretebilecek bir bilgisayar işin içinde olacak demektir. Bilgisayar bu konuda ve belli bir yol uzaklığını dikkate alarak şunları belirleyecektir: Şoförlerin tepkisel davranışları (aşırı yorgunluk), bir aracın bozuk frenlere sahip olup olmayacağı; bir şoförün kaza yapma eğilimine kapılıp kapılmayacağı, arabasını sürekli geçiş çizgisinde sürüp sürmeyeceği ya da lastiklerinden birinin patlayıp patlamayacağı...

Bütün bu rastlantısal durumlar hesaba katıldığında, önü kesilen bir akarsuyun nasıl dalga dalga kabardığı, belirli olaylardaki tehlike durumuna nasıl dramatik bir boyut aldığı çok daha gerçekçi bir biçimde taklit edilebilir (simulation) -tıpkı 100 Km/hız'ın etkisi gibi-.

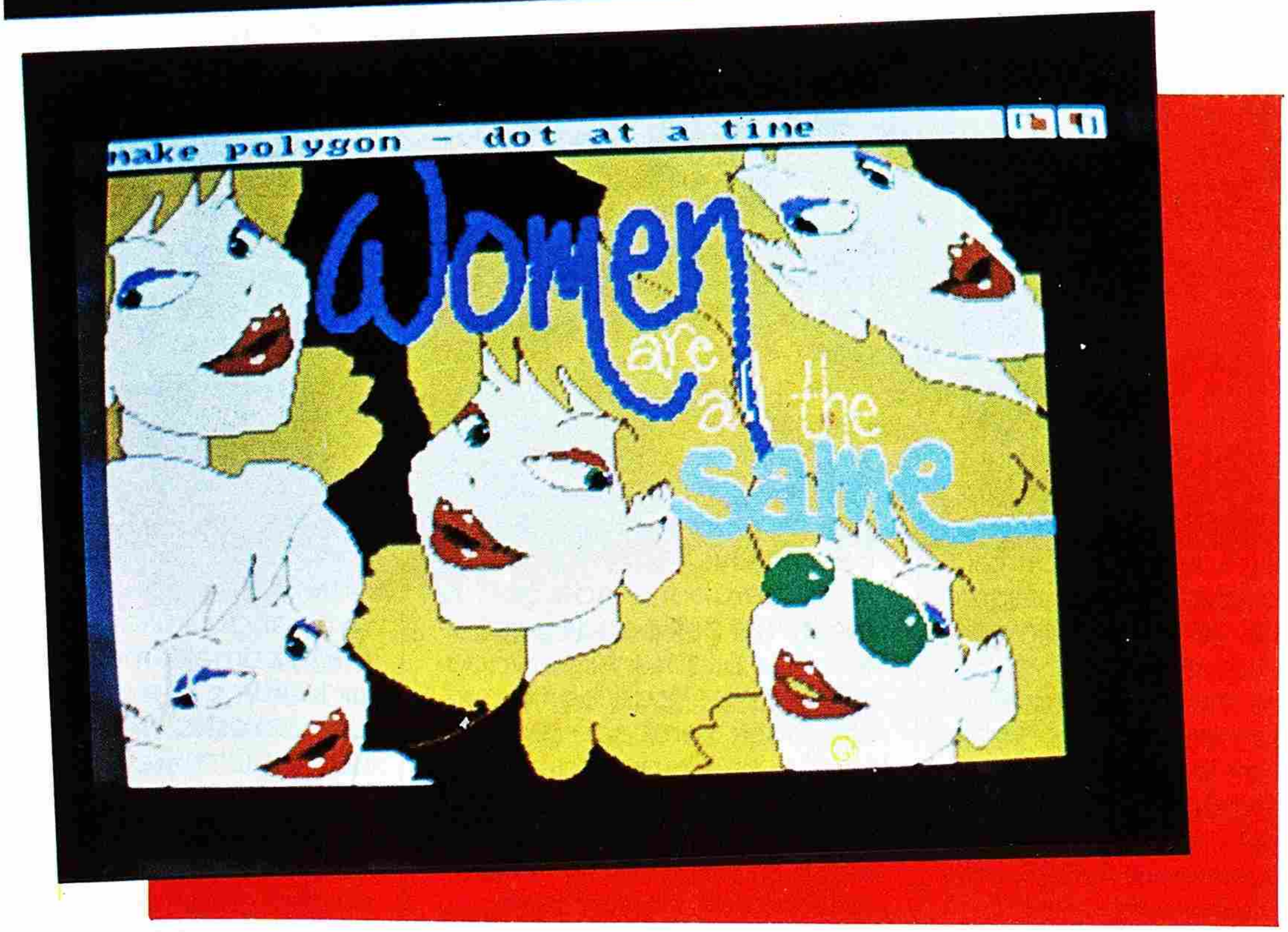
**Çev. Güven Savaş KIZILTAN**

# Yaratıcı Bilgisayar: AMİGA

Ali Murat ERKORKMAZ

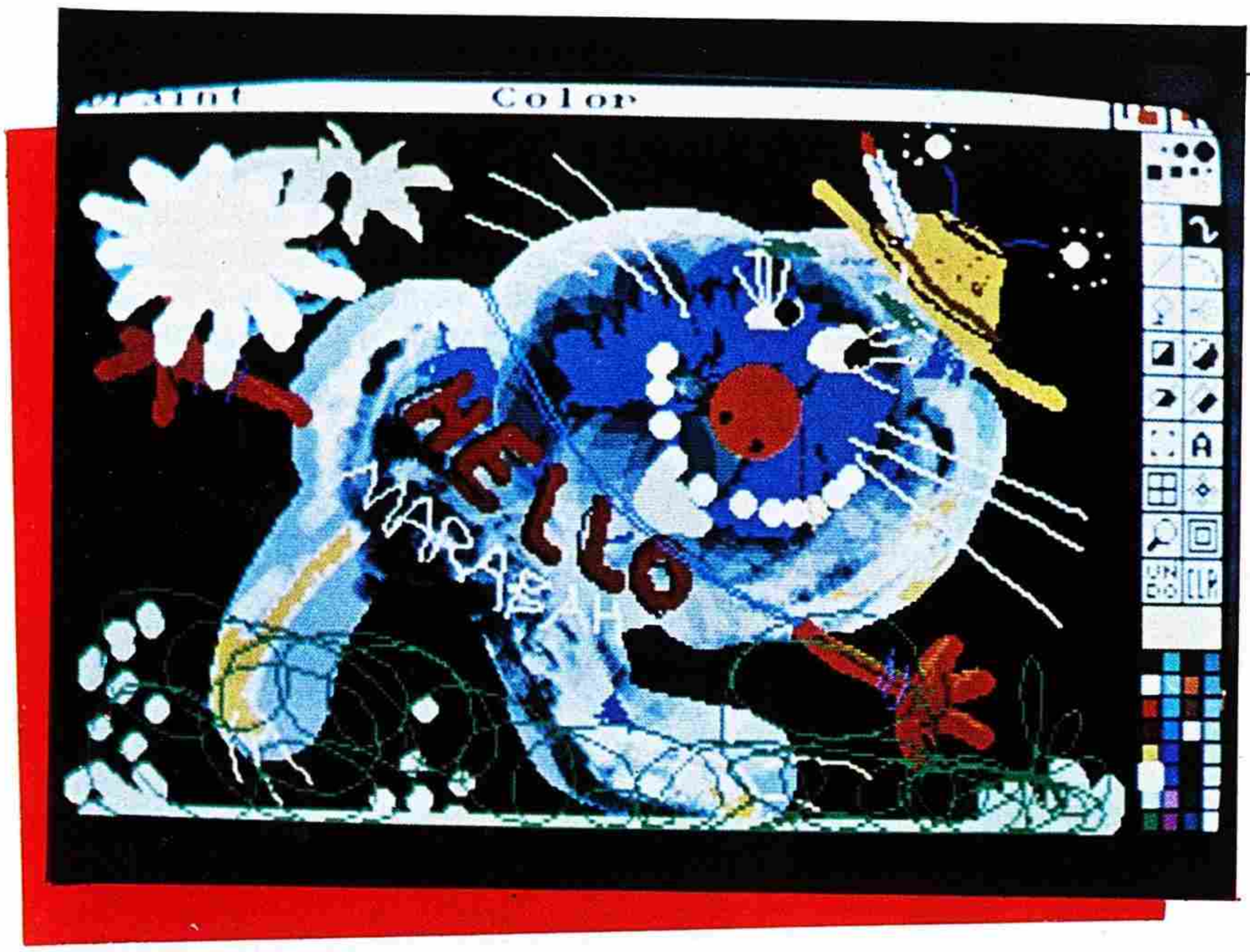


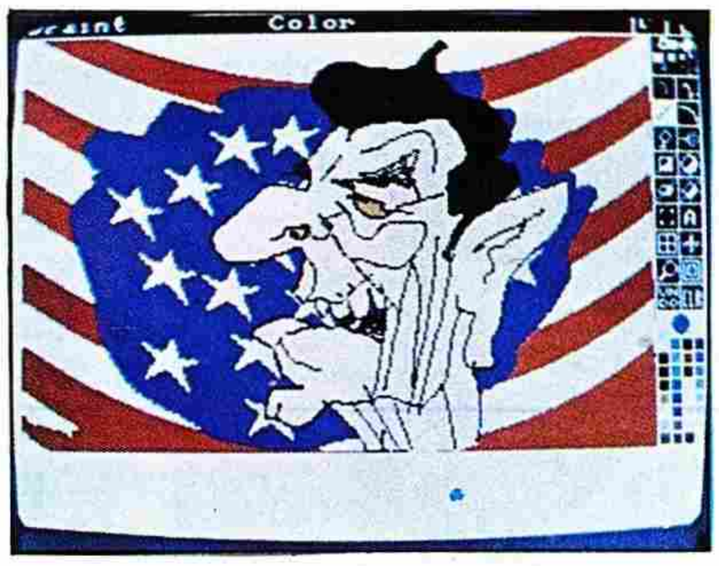





Daha beş altı yıl evvel bilgisayarlar büyük işyerlerine has pahalı anlaşılması zor, çalıştırılması güç kopleks makinelerdi. Yaptıkları işler açısından da hemen hepsi "İş"e yönelik büyük birer hesap ve arşiv depo aygıtı idiler. Gün gelip bunların evlerde kullanılabileceği hem işlev hem fonksiyon açısından akla gelmezdi.

Derken mikro çipin evrimi ile birlikte 1980 başlarında 2 Kbyte, 5 Kbyte gibi hafızalarla ve minik fiyat etiketleri ile bu akıllı makineler ev kapılarını aralamaya başladılar. Herkes bu aletlerin evde nasıl bir ihtiyacı karşılayabileceğini merak ediyordu. Elbette evdeki patates ve soğan stokunu tutmak için veya bakkal hesabını düzene sokmaya




böyle bir makineye yüzbinlerce lirayı bağlamaya gerek yoktu. Yeni bir alana gerek vardı. Çalışırken olmadığına göre dinlenirken olabilirdi. Ortaya bilgisayar oyunları çıktı. İlk defa bilgisayar iş dışında bir grafik ve animasyon ortamında kullanılıyordu.

İlk oyunları hatırlarsınız. Bir iki pikselden oluşan garip "uzay"lılar ve bir iki çizgiden oluşan sözde dünyanın istilasının önlenmesi. Bir anda etrafta yüzlerce bu tip oyun türedi. İsimleri değiştirip aynı oyunları piyasaya süren yazılım şirketleri vurguna başladılar. Ve bu sayede bilgisayar evlere girmeyi başardı. Bu arada hardware şirketleri oyun programlarını daha cazip kılabilmek ve daha görkemli sahneler sergileyebilmek amacı ile renklerini artırıp sese de önem vermeye başladılar. Çiplerin de hızla gelişmesi bilgisayarların kapasitelerini bir anda iş bilgisayarları standardına çıkartmalarını sağladı. Hafıza yükselince resolusyon ve işlevler arttı. 8 bit'ler yerlerini 16 bit'e ve 32 bit'e bırakırken bilgisayarlar arasındaki renk grafik ve ses savaşı da son hızla artmaya devam etti.

Ve devreye "Personal Computers" dediğimiz iş amaçlı kullanılan fakat çok yönlü gerçekte ev ve iş bilgisayarları arasında kalmış bir sınıf girdi. Bu bilgisayarlar yeni bir sahayı da beraberlerinde getirdiler. Bu da çok pahalı ve sofistike bazı bilgisayarların yapabildikleri özel bazı işleri ki bunlar yüksek derecede grafik çizimleri müzik programlarının hazırlanması ses örneklemesi gibi yüz milyonlarca liralık makinelerin yapabildiği şeylerdi, yapabilip küçük firmaların bu sahada çalışmalarını özendirmekti.

İşte Amiga burada devreye girdi ve küçük firmaların verebilecekleri bir ücret karşılığında çok yönlü bir iş makinesini piyasaya sürdü. Bu minik cihaz hem ünlü müzik makinelerinin yapabildiği çok sesli müzik ve ses efektlerini yapabilmekte hem de video grafikleri konusunda hiç de amatör sayılmayacak boyutta hizmet verebilmekte idi.

Amiga'nın çok yönlülüğünü araştırmak istersek gelişmeleri de gözönünde tutarak şu özelliklerini sıralayabiliriz:

a) Renk ve grafik boyutunda Amiga 4096 rengi ve 640*400 çizgi çözümlemesi ile video standardını yakalamış durumdadır. Genlock imkânı ile profesyonel cihazlara bağlanabilmektedir. Animasyon ve çizim programları her ne kadar düşük çözümlemeye göre hazırlanmış olsalar da Amiga'nın Ram hafızasının artması ile bu program çözümlemeleri de istenilen boyutlara çıkabilecektir. Birçok bilgisayarın grafik kartı kullanarak hazırlayabildiği grafik programları Amiga özel grafik ve Animasyon çipi ile kendi hardware'i içinde halledebilmektedir. Bu da ikinci bir grafik kartı yatırımını ortadan kaldırmaktadır. IFF olarak saklanan resimler değişik programlara aktarılabilmektedir ve burada işlenebilmektedir.

b) Ses konusunda örnek olarak Fairlight CMI müzik kompüterinin iç devreleri örnek alınmış ve 4 kanal digital ses imkânı sağlanmıştır. Halen Amerika'da 16 kanal ses ve aynı anda kullanılabilen 16 değişik örnekleme ses (ki bunlar instruman veya insan sesinden orkestral seslere kadar uzanmaktadırlar) işlenebilmektedir. Çıkış portlarının stereo olması ve RCA'disi fişleri ile çıkış alınabilmesi cihazın hem ev hem de profesyonel makinelere bağlantısını kolaylaştırmaktadır. Midi cihazlara bağlantı için bir midi giriş çıkışı olup yine midi interface'li başka müzik enstrümanlarını control edebilmektedir.

c) Robot kontrol ve proses kontrolü için ise Amiga hem seri hem paralel portları ile










açık bir mimari üzerinde çok geniş anlamda kullanma imkânı vermektedir. Bugün artık vazgeçilmez hale gelen elektrikli ve elektronik aygıtların ve iş robotlarının kontrolünde Amiga çok iyi netice vermektedir.

d) Frame Grabber, yani video veya kameradan görüntü kapma imkânı sayesinde hem grafik programları için hem de arşivler için Amiga çok ucuz bir sistem oluşturmaktadır çünkü bugün diğer bilgisayarların yalnızca frabe Grabber üniteleri Amiga'nın tüm fiyatından fazladır.

e) Amerika'da Amiga için yapılan bazı ek cihazlarla "speech Recognition" yani söylenileni anlama %98.6 oranında gerçekleştirilmiştir. Zaten kendi içinde bulunan konuşma ünitesi ile bu sayede Amiga konuşup dinleyep bir "canlı" makine oluşturmaktadır. Bu da makineden androide geçişte gerçek bir köprüdür. Yakın bir gelecekte bugün mouse ile kumanda edilen Amiga yalnızca konuşarak kontrol edilebilecektir.

f) Disk işletme sistemleri ve ana operasyonu artık herhangi bir lisana gerek duymadan rahatça çalıştırılabilinecektir. Bu da cihazın daha yaygın kitlelerce kullanılımını sağlayacaktır.

g) Hafızanın 8.8 megabyte ve depolamanın 1/2 gigabyte'a çıkabilmesi piyasanın dev bilgisayarlarına karşı Amiga'nın üstünlüğü olmuştur.

Amiga'nın tarifi bugün artık zorlaşmıştır. Her isteyen Amigaı'sında kendi işi için mutlaka süper bir yön bulacaktır. Bütün mesele doğru seçimi yapıp masanın üzerine bu "canlı" makineden bir adet koymaktır. □





# KALEPORSELEN'DEN KESİNTİSİZ GÜÇ KAYNAĞI

- Dünyanın en büyük Kesintisiz Güç Kaynağı üreticisi MERLIN GERIN şimdi Türkiye'de Kaleporselen ile
- 125 VA'dan 3600 kVA'ya kadar eksiksiz tüm seri.
- Küçültülmüş ölçüler ve gürültüden arındırılmış ses.
- 20 yıllık üretim tecrübesi ve en gelişmiş üretim teknolojisi.
- Tecrübeli ve bilgili teknik kadro ile satış sonrası teknik servis güvencesi.
- Bankacılık, sigortacılık, haberleşme, sağlık, eğitim.... vb. sektörleri ile askeri tesisler için uygun modeller ve frekans çeviricileri.

*Broşür isteyiniz*

**KALEPORSELEN**
*"Elektroteknikte Emniyet"*

BARAJANS

**MERKEZ VE FABRİKA**
Halkalı Asfaltı No:170 Sefaköy-İstanbul
Tel: 579 12 10 (2 Hat) 580 13 35 (3 Hat)
Pazarlama Md: 579 02 29 - 579 82 88
Telex: 28771 KASE TR.
Telefax: (9-1) 579 09 80

**MARMARA BÖLGESİ**
Okçumusa Cad. No:82
Karaköy-İstanbul
Tel: 155 35 51 - 155 31 96
Telefax: (9-1) 155 31 96

**İÇ ANADOLU BÖLGESİ**
Ambarlar Cad. Beyaz Saray Apt.
No:10/1 Sıhhiye-Ankara
Tel: 29 36 11
Telefax: (9-41) 29 14 05

**EGE VE BATI AKDNZ. BÖL.**
Lozan Meydanı No:3
Alsancak-İzmir
Tel: 21 55 77 (3 Hat)
Telefax: (9-51) 21 14 28

**DOĞU AKDNZ. VE GÜNEYDOĞU ANADOLU BÖL.**
T.Cemal Beriker Bulvarı
Kalaoğlu İş Hanı
Kat:3 No:7-8 Adana
Tel: 19 765
Telefax: (9-711) 35 047

**KARADENİZ BÖLGESİ**
Kâzımpaşa Cad. Alemdağı İş Hanı
No:5 Samsun
Tel: 13 510
Telefax: (9-361) 20 862

# Ana okulundan üniversiteye büyük okul Commodore.

Commodore'un zengin bir dizi oluşturan hazır eğitim programlarından ve eğitim kitaplarından çocuğunuza siz de alın. Büyük okul Commodore'un "dahi"leri arasındaki yerini hemen alsın!

## EĞİTİM PROGRAMLARI DİZİSİ

| | |
|---|---|
| İLKOKUL | |
| İlkokul | 1. 2. 3. sınıf Matematik |
| İlkokul | 4. 5. sınıf Matematik |
| İlkokul | 4. 5. sınıf Genel Yetenek Test. |
| Playful Prof. | |
| ANADOLU LİSELERİNE HAZIRLIK | |
| İlkokul | 4. sınıf Fen |
| | 4. sınıf Türkçe |
| | 4. sınıf Sosyal |
| | 5. sınıf Matematik |
| | 5. sınıf Fen |
| | 5. sınıf Türkçe |
| | 5. sınıf Sosyal |
| ORTAOKUL | |
| Ortaokul | Mat. Denklemler. |
| | 1. sınıf Matematik |
| | 1. sınıf Fen |
| | 2. sınıf Matematik |
| | 2. sınıf Fen |
| | 3. sınıf Matematik |
| | 3. sınıf Fen |
| Bilçağ | 1. sınıf Matematik |
| | 2. sınıf Matematik |
| | 3. sınıf Matematik |
| LİSE | |
| Bilçağ | 1. sınıf Fizik |
| | 2. sınıf Fizik |
| | Biyoloji |
| | Trigonometri |
| | Elektr ÜSYS-64 |
| Lise | Mekanik |
| | Matematik |
| Bilçağ | 1. sınıf Türkçemiz |
| | 2. sınıf Türkçemiz |
| İNGİLİZCE | |
| | Dil Eğitim Paketi |
| | Ortaokul İngilizce |
| | Cave of the World Wizard |
| | Talking Teacher |

OKUL ÖNCESİ VE OKULA YENİ BAŞLAYANLAR İÇİN:

- Hide And Seek
- Garfield
- People
- Mimi (Disket)
- Get Ready To Read
- Games of Logic
- Know Your Child's IQ

## COMMODORE EĞİTİM KİTAPLARI

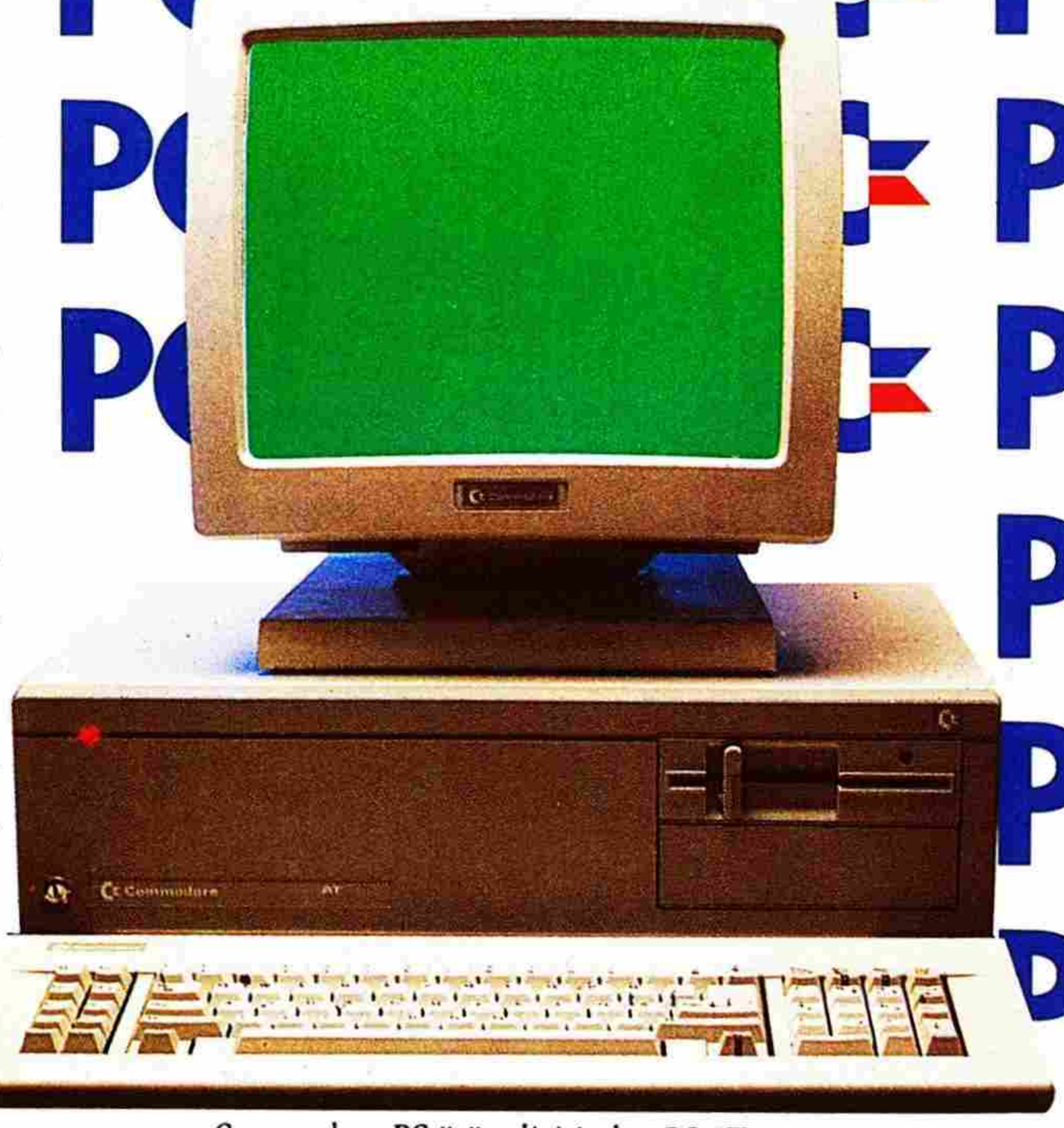
- Programcının El Kitabı
- Basic Programlama Kitabı
- Bilgisayar Ansiklopedisi
- C-64 Türkçe Kullanım Kılavuzu
- 1541 Disk Sürücü Kullanım Kılavuzu
- Easy Script C-64 Kelime İşlem Prog.Kılavuzu
- Peeks & Pokes on the C-64
- Machine Language Book for the C-64
- Advanced Machine Language Book for the C-64
- Idea Book C-64
- Graphics Book for the C-64
- The Anatomy of a C-64
- Tricks & Tips for your C-64
- The Anatomy Of C-128
- The Anatomy Of 1571
- Tricks & Tips For The C-128
- Artifical Intelligence on the C-64
- Business Applications for the C-64
- The Working Commodore 64
- Beginning Machine Code on the C-64
- Graphics art on the C-64
- Commodore 64 Adventures

MARKOM

Commodore'un hazır eğitim programları ve eğitim kitapları Etap Marmara'da, Commodore Show'da.

*Teleteknik bir Kavala Grubu kuruluşudur.*

Teleteknik



Commodore PC ürün dizisinden PC AT

# Commodore PC ile bir tanışın, başka iş bilgisayarı tanımazsınız.

Commodore PC ihtiyacınızı mükemmel karşılayacak iş bilgisayarıdır:
İleri teknolojisiyle... Yaygın ve sürekli servisi, geniş yazılım desteğiyle...
Uygun fiyatı, ödeme koşullarıyla...
Teleteknik'te ya da Yetkili Satıcıları'nda Commodore PC'leri görün.
İnceleyin. Tercih edin.

**Commodore PC'de Profesyonellerin Aradığı Standart Üstünlükler:**

| | | |
|---|---|---|
| **Bellek** | PC 10 - II | : 640 KB |
| | PC 20 - II | : 640 KB |
| | PC AT | : 1 MB |
| **Disk Sürücü** | PC 10 - II | : 2 x 360 KB Disket |
| | PC 20 - II | : 20 MB Harddisk + 360 KB Disket |
| | PC AT | : 20 MB Harddisk + 1.2 MB Disket<br>30-80 MB (optional) |
| **Interface'ler** | | : RS 232C Seri-Centronics Paralel |
| **Ekran** | | : Renkli veya Monochrome |
| **AGA Görüntü Kartı** | | : Hercules Monochrome Grafik Uyumlu<br>Renkli Grafik Emilasyon<br>Renkli Grafik Uyumlu<br>Plantronics Color Plus Uyumlu<br>16 Renk, 640x200 Çözümleme Modu<br>132 Kolon Çalışma Özelliği |
| **Merkezi İşlem Birimi** | PC 10 - II | : Intel 8088 (4.77 MHz) |
| | PC 20 - II | : Intel 8088 (4.77 MHz) |
| | PC AT | : Intel 80286 (6/10 MHz, 16/32 Bit) |
| **Software** | | : MS-DOS, GWBasic |
| **Standart Türkçe Karakter Seti** | | |

**Teleteknik**

Commodore'un Türkiye ve K.Kıbrıs Türk Cumhuriyeti Tam ve Tek Yetkili Distiribütörü

Merkez: Rıhtım Cad. Nesli Han 207/4, 80030 Karaköy, İstanbul 145 13 30 (4 hat)
Bilgisayar Danışma Merkezi: Silahhane Cad. 59 80200 Teşvikiye, İstanbul 131 15 50

*COMMODORE, Commodore Electronics Limited'in tescilli markasıdır.*

PC Commodore®

**Commodore PC Yetkili Satıcıları**

**ADANA**
Bilgisaray 35 156
Güney Yayın 30 916
Modem 38 356

**ANKARA**
Akpınar 39 57 87
Akustik 18 43 96
Altınbaşak 33 25 07
Feykem 22 67 92
Ortaş 31 44 14
Önel 33 76 09

**BURSA**
Ramesa 24 321

**ÇORUM**
Elektro Market 21 660

**DENİZLİ**
Denbim 16 935
Dos-Bil 18 769

**HATAY**
Su Ltd. 12 899

**İSTANBUL**
Artnet 337 90 79
Aykom 173 13 51
Aytaç 520 14 06
Bebim 561 24 24
Bentaş 147 58 40
Bilmerk 141 03 37
Bimaks 166 70 72
Biton Ltd. 143 50 20
Bürsan 166 58 77
Bükomak 172 27 86
Ceypa 521 42 04
Çağdaş 337 18 86
Enter 172 71 55
Erkutun 520 53 66
Geta 160 40 39
İstanbul Paz. 167 65 70
Memory 360 01 23
Mikrobahçe 148 19 75
Norsan 151 36 64
Nova 147 28 00
Optimum 172 27 05
Ramesa 358 66 56
Rom 160 13 24
Sameks 160 25 18
Tempo 528 59 00
Tes 145 19 18

**İZMİR**
Boyut 22 02 35
Ege 14 83 41
Yabim 25 86 75

**KOCAELİ**
Vibi 18 116

**KONYA**
Muammer Bişkinler

**KÜTAHYA**
Kütahya Bilim 32 73

**MALATYA**
Bilem 13 211

**MERSİN**
Retap 22 693

**SAMSUN**
Güneş Kardeşler 31 084

MARKOM