

Problem A

2025

Time Limit: 2 seconds

日本の小学校では以下の掛け算表を習います。

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

このような 9×9 の項目を持つ表を「九九」と呼びますが、これは世界標準というわけではありません。 12×12 や 20×20 の表を習う文化もあります。

この九九の 9×9 個の項目の総和は、ちょうど今年の西暦年である 2025 なのです。他にもこのような面白い例はないでしょうか。

あなたの仕事は、与えられた整数 n に対して $n \times n$ の掛け算表の項目の総和を求めることです。つまり、

$$\sum_{a=1}^n \sum_{b=1}^n ab$$

を計算してください。

Input

入力は 100 個以内のテストケースからなる。各テストケースは整数 n ($1 \leq n \leq 100$) ひとつを含む 1 行からなる。

入力の終わりは、1 個のゼロだけからなる行で表される。

Output

各テストケースについて、

$$\sum_{a=1}^n \sum_{b=1}^n ab$$

の値を 1 行に出力しなさい.

サンプル入出力は DOMjudge の Problemset ページからダウンロード可能である.

Sample Input 1

```
9
1
2
100
0
```

Sample Output 1

```
2025
1
9
25502500
```

Problem B

接頭辞と接尾辞を同じに

Time Limit: 2 seconds

この問題では、文字列の接頭辞とは、その先頭文字から連続する部分文字列を意味する。例えば文字列 `icpc` の接頭辞は `i`, `ic`, `icp`, `icpc` である。同様に、文字列の接尾辞とは、その末尾文字まで連続する部分文字列を意味する。例えば文字列 `icpc` の接尾辞は `icpc`, `cpc`, `pc`, `c` である。

与えられた文字列 s に対して、 s が接頭辞にも接尾辞にもなっている s 自身以外の文字列で最短のものを欲している。そのような文字列がひとつしかないことは証明できる。

Input

入力は 50 個以下のテストケースからなる。各テストケースは次の形式で表される。

n
 s

1 行目の n は 1 以上 50 以下の整数である。2 行目には英小文字のみからなる長さ n の文字列 s が与えられる。

入力の終わりは、1 個のゼロだけからなる行で表される。

Output

各テストケースについて、 s が接頭辞にも接尾辞にもなっている s 以外の文字列のうちで最短のものを 1 行に出力せよ。

サンプル入出力は DOMjudge の Problemset ページからダウンロード可能である。

Sample Input 1

```
4
test
4
icpc
8
ccccppcc
1
i
9
strongest
3
ttt
0
```

Sample Output 1

```
testest
icpcicpc
ccccppccccppcc
ii
strongeststrongest
tttt
```

Problem C

働き者のカレンダー

Time Limit: 2 seconds

あなたは仕事大好きな働き者である。これからしばらくの間に働ける日数を知りたい。

土曜日と日曜日は休日である。また追加の休日として国民の祝日や会社独自の休日がある。休日にはあなたは働けない。こうした休日は、重なることもある。たとえば会社の創立記念日が国民の祝日であることもある。これらの休日が重なっても、別の日を休みとするような振替制度はない。

今日は月曜日である。今日から始まる指定期間のうち、働ける日数を求めよ。

Input

入力は 100 個以下のテストケースからなる。各テストケースは次の形式で与えられる。

$$\begin{matrix} n & m \\ a_1 & \cdots & a_n \end{matrix}$$

1 行目に追加の休日の数 n と指定された期間の日数 m が与えられる ($1 \leq n \leq 300$, $1 \leq m \leq 10^{18}$).
2 行目に n 個の整数 $a_1, \dots, a_n$ が与えられる ($1 \leq a_i \leq 10^{18}$). 今日を 1 日目として、各 $i = 1, \dots, n$ に対し、 a_i 日目が追加の休日であることを表す。 $a_1, \dots, a_n$ には同じ値が 2 回以上現れることがある。

入力の終わりは、2 個のゼロからなる行で表される。

Output

各テストケースについて、今日から始まる m 日間のうちで働ける日数を 1 行に出力せよ。

サンプル入出力は DOMjudge の Problemset ページからダウンロード可能である。

Sample Input 1

```
4 8
1 2 3 4
2 10
15 3
3 14
6 7 13
5 1000000000000000000
8 15 1 8 999999999999999994
2 3
1 4
0 0
```

Sample Output 1

```
2
7
10
714285714285714282
2
```

Problem D

古代のゲーム盤

Time Limit: 2 seconds

古代 Icpca 文明では同じ大きさの正方形のタイルを敷き詰めた盤面を用いたゲームが行われていた。各タイルの表面が白または黒のどちらかなのは確かだが、各色のタイルの配置については解明されていない。

近年有力とされる学説では、盤面には同色のタイルを正形状に敷き詰めた同じ大きさの領域が、チェス盤のように前後にも左右にも白黒交互に隙間なく並んでいたのだろうとされている。

今日、このゲームに用いられた盤面の一部の長方形部分が発掘された。この遺物が上記の学説と矛盾しないかを判定してほしい。矛盾しない場合は、各正方形領域の一辺のタイル数としてあり得る値が一意に特定できるかも判定してほしい。

Sample Input 1 の最初のテストケースの遺物を図 D.1 の左図に示す。これは右図の赤枠に示すように、正方形領域の一辺が 2 タイルである盤面の一部と考えることができ、遺物は学説と矛盾しない。また、それ以外の大きさの正方形領域を持つ盤面の一部にはなり得ないため、一辺は 2 タイルと一意に特定できる。

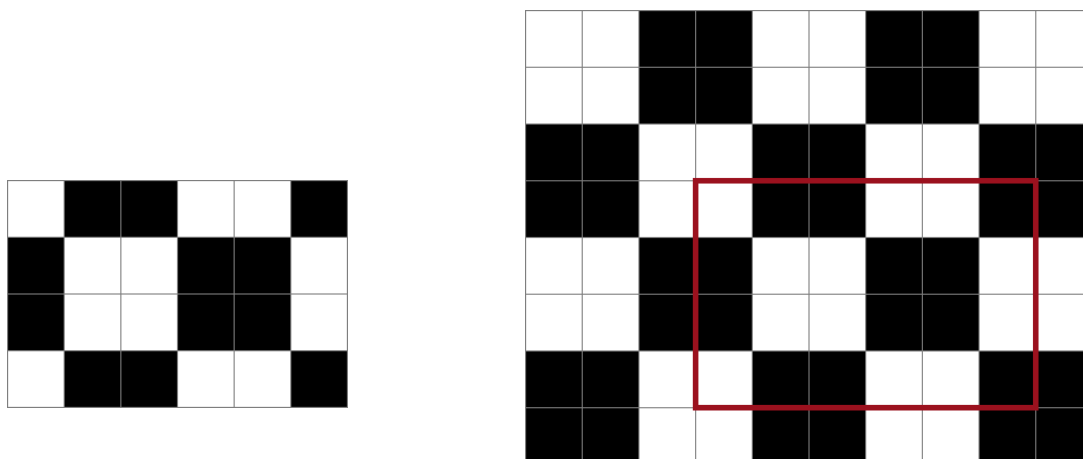


図 D.1. Sample Input 1 の最初のテストケース

Input

入力 は 30 個以下のテストケースからなる。各テストケースは次の形式で表される。

```

n m
c1,1 ⋯ c1,m
⋮
cn,1 ⋯ cn,m

```

テストケースの最初の行の整数 n と m ($1 \leq n \leq 100, 1 \leq m \leq 100$) は、それぞれ遺物（すなわち盤面の残存部分）のタイルの行数と列数を表す。続く n 行にはそれぞれ、遺物の各タイルの色を表す ‘.’ と ‘#’ のみからなる長さ m の文字列が与えられる。その i 行目の j 文字目 $c_{i,j}$ ($1 \leq i \leq n, 1 \leq j \leq m$) が ‘.’ ならば遺物の行 i 列 j のタイルは白、‘#’ ならば黒である。

入力の終わりは、2 個のゼロだけからなる行で表される。

Output

各テストケースについて、次の整数いずれか 1 つを出力せよ。

- 遺物が学説と矛盾する場合は -1
- 遺物が学説と矛盾せず、各正方形領域の一辺のタイル数を一意に特定できる場合はその値
- 遺物が学説と矛盾しないが、各正方形領域の一辺のタイル数を一意に特定できない場合は 0

サンプル入出力は DOMjudge の Problemset ページからダウンロード可能である。

Sample Input 1

```
4 6
.##..#
#...##.
#...##.
.##..#
3 3
###
#.#
###
2 3
#..
.##
4 3
#.#
#.#
#.#
#.#
4 6
.####.
#....#
#....#
#....#
3 3
#.#
.#.
#.#
0 0
```

Sample Output 1

```
2
-1
0
-1
4
1
```

Problem E

航路廃止

Time Limit: 2 seconds

時は西暦 3000 年，張り巡らされた宇宙船航路ネットワークによって，人類は宇宙の星々を旅していた．ところが，近年のワープ技術の発展に伴い，従来の宇宙船航路は順次廃止されることになった．あなたは，古き良き宇宙船による旅を惜しみ，今残っている全ての航路を廃止前に乗り尽くそうと心に決めた．

現在，星々間の航路は， n 個の星をつなぐツリー型のネットワークを構成している．すなわち，星と星をつなぐ双方向の航路が $n - 1$ 本存在し，このような航路を何度かたどることで，どの星とどの星の間も移動することができる．ただし，航路それぞれには今や最終搭乗期限が設定されている．ある星に時刻 t にいる場合，その星からの航路の最終搭乗期限 e 以前（すなわち $t \leq e$ ）であれば，その航路の先の星に時刻 $t + 1$ に到着できる．最終搭乗期限を過ぎると，その航路は使用できない．

また，従来の宇宙船航路に加えて，あなたは新しいワープ技術を使うこともできる．すでに一度訪れたことがある星へは，どの星からでもワープによって時間を使わず瞬時に移動できるのだ！あなたは今， n 個の星のひとつにいる．あなたはこの旅に先立ってその星以外を訪れたことはないので，他の星へのワープを使うには，目的の星にまず一度は宇宙船航路で到達する必要がある．

さて，時刻 0 にこの旅を始めるあなたは，すべての航路をそれぞれの最終搭乗期限内に乗り尽くすことができるだろうか？ ネットワークはツリー型であるため，この質問は，他の星すべてを訪れることができるか否かに等しい．それが可能な場合は，どの順番で星々を訪ればよいかも求めて欲しい．

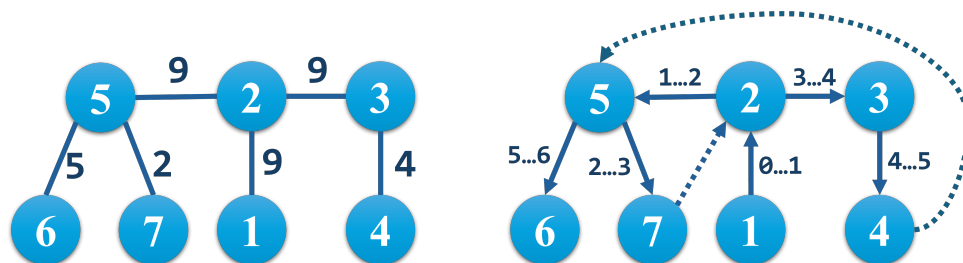


図 E.1. Sample Input 1 の最初のテストケース

図 E.1 は Sample Input 1 の最初のテストケースを図示している．左図は宇宙船航路ネットワークを表し，線分に添えられた数は航路の最終搭乗期限を表す．右図はすべての航路を乗り尽くす方法を表す．右図において点線はワープを表し，“ $t \dots t + 1$ ”は時刻 t に搭乗し $t + 1$ に到着する宇宙船航路の利用を表す．

Input

入力 は 50 個以下のテストケースからなり，各テストケースは次の形式で表される．

```

n
p2 e2
⋮
pn en

```

1 行目の整数 n ($2 \leq n \leq 1000$) は宇宙船航路ネットワークが結ぶ星の数を表す。星には 1 から n までの番号が付いていて、あなたが最初にいるのは星 1 である。続く 2 行目から n 行目は宇宙船航路の情報を表し、整数 p_i ($1 \leq p_i < i$) と e_i ($0 \leq e_i \leq 1000$) は、星 i と星 p_i との間に最終搭乗期限 e_i の航路があることを意味する ($i = 2, \dots, n$)。

入力の終わりは、1 個のゼロだけからなる行で表される。

Output

各テストケースについて、すべての航路をそれぞれの最終搭乗期限内にたどることができるならば、1 行目に yes と出力し、そうでなければ no と出力せよ。

もしそれが可能なら、次の行に星 2 から星 n の番号を初めて訪れる順に 1 行にスペース区切りで出力せよ。複数の方法がある場合は、どれを出力しても良い。

サンプル入出力は DOMjudge の Problemset ページからダウンロード可能である。

Sample Input 1

```
7
1 9
2 9
3 4
2 9
5 5
5 2
4
1 1
1 1
1 1
0
```

Sample Output 1

```
yes
2 5 7 3 4 6
no
```

Problem F

犬の芸

Time Limit: 2 seconds

あなたの2匹の飼い犬 Amy と Bessie が行う多彩な芸の動画は、いまや Icpca 王国の人気コンテンツである。さらなる動画の可能性を探るため、あなたは犬たちに新しい芸を覚えさせた。

新しい芸は、左右に一列に並んだ皿を用いて行う。各皿にはリンゴまたはバナナのうちのどちらかが置かれている。

あなたが Amy に指示を出すと、Amy はバナナをくわえて、左端から順に皿を見ていく。バナナの皿は素通りし、初めてリンゴの皿まで来たら、くわえていたバナナはその皿に置き、皿にあったリンゴをくわえて右に進む。今度はリンゴの皿は素通りして、初めてバナナの皿まで来たら、くわえていたリンゴはその皿に置いて、皿にあったバナナをくわえてあなたの元に帰り、芸は成功に終わる。これができるように果物が皿に並んでいなかったら、芸は失敗に終わる。

Bessie に指示を出すと、Bessie はリンゴをくわえて、左端から順に皿を見ていく。リンゴの皿は素通りし、初めてバナナの皿まで来たら、くわえていたリンゴはその皿に置き、皿にあったバナナをくわえて右に進む。今度はバナナの皿は素通りして、初めてリンゴの皿まで来たら、くわえていたバナナはその皿に置いて、皿にあったリンゴをくわえてあなたの元に帰り、芸は成功に終わる。これができるように果物が皿に並んでいなかったら、芸は失敗に終わる。

2匹の犬に同時に指示を出したり、一方の犬が芸をしている間にもう一方の犬に指示を出すことはできない。

次の動画では犬たちに成功する芸を繰り返し行わせて、皿の上の果物を目標の並びにしたい。皿の上の果物の初期状態と目標状態が与えられる。初期状態からはじめて、Amy と Bessie に定められた回数以内の指示を適切な順で出し、果物の並びを目標状態にできるだろうか。できるのならそのような指示の列を具体的に求めて欲しい。

Input

入力は100個以下のテストケースからなる。各テストケースは次の形式で表される。

$$\begin{array}{l} n \\ s \\ t \end{array}$$

テストケースは3行からなる。1行目には皿の数を表す1以上100以下の整数 n が与えられる。2行目には初期状態を表す文字列 s が与えられる。初期状態で左から i 番目の皿の上にある果物は、 s の i 番目の文字が a のときリンゴであり、 b のときバナナである。3行目には目標状態を表す文字列 t が与えられる。目標状態で左から i 番目の皿の上にあるべき果物は、 t の i 番目の文字が a のときリンゴであり、 b のときバナナである。 s と t は a と b だけからなり、長さはともに n である。 s と t は異なる文字列である。

入力の終わりは、1個のゼロだけからなる行で表される。

Output

各テストケースについて、10 000 回以内の犬への指示の列で初期状態から目標状態にするものがあれば `yes` を、なければ `no` を 1 行に出力せよ。さらに、そのような指示の列がある場合は、その列を表す文字列を次の行に出力せよ。この文字列は文字 `A` と `B` からなるもので、 i 番目の指示が Amy に対するものなら文字列の i 番目は `A`、Bessie に対するものならば `B` とせよ。適切な指示の列が複数ある場合はどれを出力してもよい。

サンプル入出力は DOMjudge の Problemset ページからダウンロード可能である。

Sample Input 1

```
2
aa
bb
3
bba
bab
4
aaba
abaa
0
```

Sample Output 1

```
no
yes
BA
yes
AB
```

Sample Input 1 の 2 番目のテストケースでは、初期状態 `bba` に対して、まず Bessie に指示を出すと、左端のバナナをリングに置き換えて `aba` となり、次に右端のリングをバナナに置き換えて `abb` となる。次いで Amy に指示を出すと、左端のリングをバナナに置き換えて `bbb` となり、さらに隣のバナナをリングに置き換えて `bab` となる。これは目標状態である。

Problem G

面の数

Time Limit: 2 seconds

x, y, z 軸を持つ 3 次元ユークリッド空間中に 2 つの平面 $H_1: z = 1$ と $H_2: z = 2$ がある。

n 個の実数 $d_1, \dots, d_n$ と m 個の実数 $d'_1, \dots, d'_m$ が与えられる。これらの実数は正で、180 未満である。平面 H_1 と H_2 のそれぞれに以下のような凸多角形を描くことを考える。

- H_1 には n 角形を描く。この多角形の頂点の内角は、原点から見て反時計回り順に $d_1, \dots, d_n$ 度である。
- 同様に、 H_2 には m 角形を描く。この多角形の頂点の内角は、原点から見て反時計回り順に $d'_1, \dots, d'_m$ 度である。

ここで指定されているのは内角の大きさだけであり、凸多角形の辺の長さや頂点の位置は指定されていない。

2 つの凸多角形の各頂点の位置が定まると、それら $n + m$ 個の頂点を頂点集合とする凸多面体は一意に決まる。このような凸多面体を持ちうる面の数を列挙するプログラムを書いてほしい。

ここでは、凸多面体において二面角（隣り合う二面のなす角）はすべて 180 度未満でなければならない。

Sample Input 1 の最初のテストケースでは、 H_1 と H_2 に内角がすべて 90 度である四角形を描く。例えば、図 G.1 (a) のような直方体を構成でき、これは 6 面を持つ。片方の四角形を回転すれば、図 G.1 (b) のような 10 面を持つ凸多面体も構成できる。持ちうる面の数は 6 と 10 である。

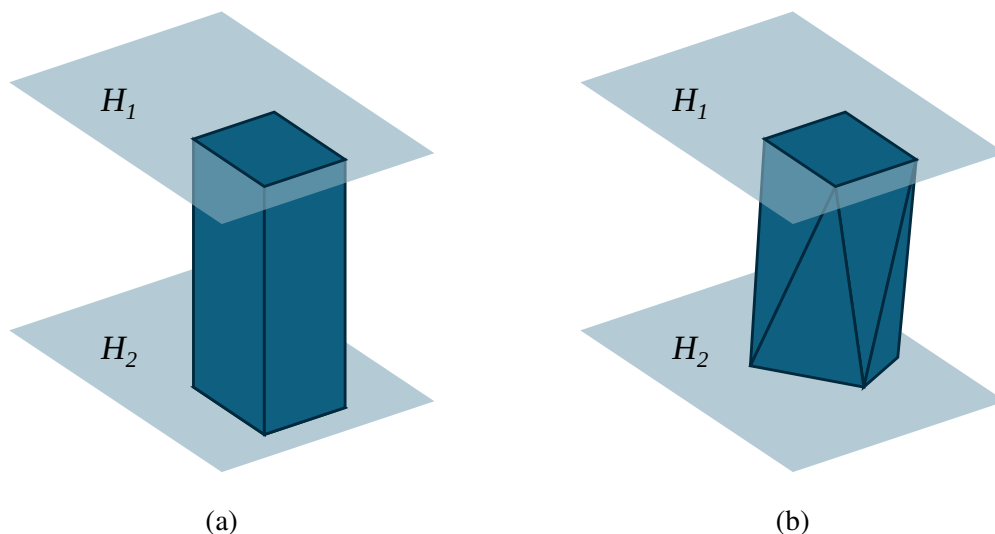


図 G.1. Sample Input 1 の最初のテストケース

Input

入力は 50 個以下のテストケースからなる。各テストケースは次の形式で表される。

$$\begin{array}{l} n \\ d_1 \\ \vdots \\ d_n \\ m \\ d'_1 \\ \vdots \\ d'_m \end{array}$$

整数 n は H_1 に描く多角形の頂点数を表す ($3 \leq n \leq 50$)。実数 $d_1, \dots, d_n$ は、内角の大きさを表し、それぞれ 10^{-9} 以上 180 未満であり、小数点以下ちょうど 9 桁まで与えられる。これらは $d_1 + \dots + d_n = (n - 2) \times 180$ を満たす。

同様に、整数 m は H_2 に描く多角形の頂点数を表す ($3 \leq m \leq 50$)。実数 $d'_1, \dots, d'_m$ は、内角の大きさを表し、それぞれ 10^{-9} 以上 180 未満であり、小数点以下ちょうど 9 桁まで与えられる。これらは $d'_1 + \dots + d'_m = (m - 2) \times 180$ を満たす。

入力の終わりは、1 個のゼロだけからなる行で表される。

Output

各テストケースについて、凸多面体が持ちうる面の数をスペース区切りで昇順に 1 行で出力せよ。

サンプル入出力は [DOMjudge の Problemset ページ](#) からダウンロード可能である。

Sample Input 1

```
4
90.0000000000
90.0000000000
90.0000000000
90.0000000000
4
90.0000000000
90.0000000000
90.0000000000
90.0000000000
3
33.3333333333
66.6666666666
80.0000000001
3
80.0000000001
66.6666666666
33.3333333333
3
59.165980540
68.504848124
52.329171336
5
87.702342452
144.626828884
97.879972796
169.296126888
40.494728980
0
```

Sample Output 1

```
6 10
6 7 8
8 9 10
```

Problem H

カッコ

Time Limit: 2 seconds

あなたの前には、開きカッコ '(' または閉じカッコ ')' のいずれか 1 文字が彫られたスタンプが左から右に 1 列に並んでいる。あなたはまず、どれかひとつのスタンプを選んで取り出し、それを紙に印字する。印字したら、そのスタンプは元の位置に戻す。次に、戻したスタンプの右隣または左隣のいずれかのスタンプを選び、それを直前の印字のすぐ後に印字する。その後も、こうした左右のスタンプの選択と印字を繰り返す。繰り返しにおいて、同じ位置のスタンプを何度でも選ぶことができるが、続けて選ぶことはできない。

この結果、紙上には開きカッコと閉じカッコだけからなる行が印字されることになる。カッコ列が正しいものであるためには、最初の印字箇所から始めて行中の文字をひとつずつ追って、開きカッコと閉じカッコの個数を数えると、途中まで数えたところでは常に開きカッコが閉じカッコと同数以上であり、最後の文字まで数えたときに両者が同数でなければならない。最初を選んで押すスタンプと最後にを選んで押すスタンプの組で、途中で適切に左右のスタンプを選ぶことで正しいカッコ列を印字できるものがいくつあるかを求めよ。

Input

入力は 10^5 個以下のテストケースからなる。各テストケースは次の形式で表される。

$$\begin{array}{c} n \\ s \end{array}$$

テストケースはそれぞれ 2 行で与えられる。1 行目はスタンプの数を表す整数 n を含む ($2 \leq n \leq 2 \times 10^5$)。2 行目は '(' と ')' からなる長さ n の文字列 s を含む。 s の i 文字目は左から i 番目のスタンプに彫られた文字を表す ($i = 1, \dots, n$)。

入力の終わりは、1 個のゼロだけからなる行で表される。全てのテストケースの n の合計は 2×10^5 を超えない。

Output

各テストケースについて、条件を満たす組がいくつあるかを 1 行に出力しなさい。

サンプル入出力は DOMjudge の Problemset ページからダウンロード可能である.

Sample Input 1	Sample Output 1
5))) ((10) ()) (()) (20 (() (())))) () () ((0	3 9 46

Sample Input 1 の最初のテストケースで，スタンプを左端から順に $1, 2, \dots$ と番号付けることにする．最初と最後のスタンプの組のうち条件を満たすものは $(4, 1), (4, 3), (5, 2)$ の 3 つである．例えば，組 $(4, 1)$ に対して $4, 5, 4, 3, 2, 1$ の順にスタンプを選ぶと，印字される行は “((()))” になる．

Problem I

昼食会の準備

Time Limit: 2 seconds

ICPC に出場する学生たちが円卓を等間隔に囲んで昼食会を開こうとしている。各学生は一つずつ仕出し弁当を持っている。弁当にはいくつか種類があり、必ずしも全員が同じ種類の弁当を持っているとは限らない。学生の人数は偶数である。したがって、それぞれの席の学生から見て、円卓の真向かいの席にちょうど一人の学生がいる。各種類の弁当の個数は偶数である。

真向かいの席の弁当はすぐ目に入るの、それが自分のものと違う種類だとうらやましくなるかもしれない。そこで、向かい合う学生同士の弁当は同じ種類になるようにしたい。どの種類の弁当も偶数個なので、このような再配置は必ず実現できる。「隣り合う学生のペアを任意に選び、この二人が持っている弁当を交換する」という操作を何度か繰り返すことでこの目標を達成するとき、全体で必要な最小の操作回数を求めよ。

Input

入力は 5×10^4 個以下のテストケースからなる。各テストケースは次の形式で表される。

$$n$$

$$a_1 \cdots a_{2n}$$

テストケースは 2 行からなる。1 行目には 2 以上 2×10^5 以下の正の整数 n が与えられる。これは食事会に $2n$ 人の学生がいることを表す。2 行目には $2n$ 個の整数 $a_1, \dots, a_{2n}$ が与えられる。 i 番目の整数 a_i は、円卓の特定の位置から数えて時計回りに i 番目の学生が最初に持っている弁当の種類を表す ($i = 1, \dots, 2n$)。 a_i は 1 以上 n 以下の整数である。なお、任意の整数 j に対して $a_i = j$ である i は偶数個であることが保証されている。

入力の終わりは、1 個のゼロだけからなる行で表される。全てのテストケースにわたる n の総和は 2×10^5 を超えない。

Output

各テストケースについて、目標を達成するための最小の操作回数を 1 行に出力しなさい。

サンプル入出力は DOMjudge の Problemset ページからダウンロード可能である。

Sample Input 1

```
4
4 2 3 4 1 2 1 3
4
1 2 1 3 2 2 2 3
3
3 3 3 3 3 3
0
```

Sample Output 1

```
2
3
0
```

最初のテストケースについては、7番目にある1と次の3を交換して42341231とし、続いて8番目の1と1番目の4を交換すれば、弁当の並びは12341234となる。これで真向かいの席には同じ種類の弁当が来ることになるので、2回の交換で済む。1回の交換では目的は達成できないので、答えは2である。