

Problem A

Mex Prize

Time Limit: 2 seconds

あるコンテストには 10^5 チームが参加した。その結果、各チームには 1 位から 10^5 位のいずれかの順位が一意に与えられた。

このコンテストにはチームの順位に応じた特別賞が設けられている。すでに a_1, a_2, \dots, a_N 位の計 N チームが特別賞を受け取った。

今回、特別賞が 1 枠追加されることになった。この特別賞は、まだ特別賞を受け取っていないチームのうち、最上位のチームが受け取る。ここで「最上位」とは、順位が最も小さいことを意味する。なお、本問題の制約上、まだ特別賞を受け取っていないチームが存在することが保証される。

新たに特別賞を受け取るのは何位のチームか？

Input

入力は複数のデータセットからなる。データセットの個数は 100 を超えない。各データセットは次の形式で表される。

$$\begin{array}{c} N \\ a_1 \ a_2 \ \dots \ a_N \end{array}$$

N はすでに特別賞を受け取ったチームの数を表す、1 以上 100 以下の整数である。各 a_i ($i = 1, \dots, N$) は 1 以上 100 以下の整数であり、 a_i 位のチームが特別賞を受け取ったことを表す。与えられる順位は昇順に整列されており、 $a_1 < a_2 < \dots < a_N$ が成り立つ。

入力の終わりは、ゼロ 1 つだけからなる行で表される。

Output

各データセットに対して、新たに特別賞を受け取るチームの順位を出力せよ。

Sample Input

```
5
1 2 4 5 7
3
1 2 3
6
2 3 4 5 6 8
0
```

Sample Output

```
3
4
1
```

Problem B

Flight Planning

Time Limit: 2 seconds

JAG 国には空港が N 個あり, M 本のフライトが予定されている. i ($i = 1, \dots, M$) 番目のフライトでは, 飛行機が時刻 i に空港 a_i を出発し, 時刻 $i + 0.5$ に空港 b_i に到着する.

各空港に任意の機数の飛行機を配置して全てのフライトを予定通りに遂行したい. 飛行機の配置とフライトの遂行には次の条件がある.

- 各フライトにはちょうど 1 機の飛行機が必要である.
- はじめ各飛行機をどの空港に配置するかは自由に決めてよい.
- 空港に到着した飛行機は, ただちにその空港から出発する別のフライトに利用してよい.
- 各空港では任意の機数の飛行機を, 任意の時間待機させておいてよい.
- 複数の飛行機を同じフライトに同時に割り当てることはできない.
- 予定されたフライト以外の手段で飛行機を移動させることはできない.

全てのフライトを予定通りに遂行するために必要な飛行機の最小機数を求めよ.

Input

入力は複数のデータセットからなる. データセットの個数は 100 を超えない. 各データセットは次の形式で表される.

$$\begin{array}{ll} N & M \\ a_1 & b_1 \\ \vdots & \\ a_M & b_M \end{array}$$

N は空港の数を表す, 2 以上 100 以下の整数である. M は予定されたフライトの本数を表す, 1 以上 100 以下の整数である. a_i と b_i ($i = 1, \dots, M$) はそれぞれ i 番目のフライトにおける出発地と到着地を表す, 1 以上 N 以下の整数である. ここで $a_i \neq b_i$ が保証される.

入力の終わりは, ゼロ 2 つからなる行で表される.

Output

各データセットに対して, 全てのフライトを予定通りに遂行するために必要な飛行機の最小機数を出力せよ.

| Sample Input | Sample Output |
|--|---------------|
| 5 6 1 3 3 2 3 5 2 4 5 1 5 1 2 5 1 2 1 2 1 2 1 2 1 2 0 0 | 3 5 |

Problem C

今日はコンテスト

Time Limit: 2 seconds

太郎くんはコンテストが開かれる 6 月 28 日を楽しみに待っている。

手帳に書かれた日付「6 / 28」が分数に見えた太郎くんは、次の事実に気が付いた：整数 28 の正の約数全体は 1, 2, 4, 7, 14, 28 であるが、これらの逆数に 6 を掛けた和

$$\frac{6}{1} + \frac{6}{2} + \frac{6}{4} + \frac{6}{7} + \frac{6}{14} + \frac{6}{28} = 12$$

がちょうど整数になる。

このような組み合わせが他にもないか気になった太郎くんのために、与えられた 2 つの数からなる「 K / N 」が同様の性質を満たすか判定するプログラムを作成してほしい。

Input

入力は複数のデータセットからなる。各データセットは次の形式で表される。

$N \ K$

N は約数をとる数を表す、1 以上 200,000 以下の整数である。 K は逆数に掛ける数を表す、1 以上 200,000 以下の整数である。

入力の終わりは 2 つのゼロからなる行で表される。入力に含まれるデータセットは 100,000 個以内である。

Output

各データセットに対して、 N, K が次の性質を満たすならば “Yes” を、そうでないならば “No” を出力せよ。

- N の正の約数全体を a_1, a_2, \dots, a_n とする。このとき、 $\frac{K}{a_1} + \frac{K}{a_2} + \dots + \frac{K}{a_n}$ が整数である。

Sample Input

Sample Output

| | |
|-----------|-----|
| 28 6 | Yes |
| 29 6 | No |
| 672 2 | Yes |
| 32760 1 | Yes |
| 100000 1 | No |
| 2025 2025 | Yes |
| 0 0 | |

Problem D

Guess the parentheses

Time Limit: 2 seconds

この問題はインタラクティブな問題である。

正しい括弧列とは、以下のいずれかの条件を満たす空でない文字列である：

- $()$ に等しい。
- 正しい括弧列 X が存在し、 $(, X,)$ をこの順に結合した文字列に等しい。
- 正しい括弧列 X, Y が存在し、 X, Y をこの順に結合した文字列に等しい。

あなたの目的は、長さ N の隠された正しい括弧列を特定することである。あなたは指定した連続部分列が正しい括弧列かを、高々 N 回知ることができる。

Interaction

はじめに標準入力から N が与えられる。 N は隠された正しい括弧列 S の長さを表す、2 以上 20,000 以下の整数である。

次の形式で質問する。

? $L R$

ここで L と R は $1 \leq L \leq R \leq N$ を満たす整数である。この質問に対して、 S の L 番目から R 番目の文字からなる連続部分列が正しい括弧列である場合は“1”が、そうでない場合は“0”が標準入力から与えられる。

次の形式で回答する。

! S

ここで S は特定した正しい括弧列である。

質問は N 回まで、回答は 1 回まで行うことができる。ただし、回答後に質問することはできない。

形式の違反や質問・回答回数の超過、余計な出力がある場合、ジャッジ結果は不定である。出力バッファのフラッシュが必要な場合があることに注意されたい。

Sample Interaction

| Read | Write |
|------|----------|
| 6 | ? 2 4 |
| 0 | ? 3 6 |
| 1 | ? 5 6 |
| 0 | ! ()(()) |

Problem E

if-then-else

Time Limit: 2 seconds

以下の BNF で表される文法を考える.

```
⟨expression⟩ ::= ⟨number⟩ | if_⟨variable⟩_then_⟨expression⟩_else_⟨expression⟩
⟨number⟩ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
⟨variable⟩ ::= ⟨letter⟩ | ⟨letter⟩⟨variable⟩
⟨letter⟩ ::= a | b | c | d | e | f | g | h | i | j | k | l | m |
           n | o | p | q | r | s | t | u | v | w | x | y | z
```

⟨expression⟩ で表される要素を式, ⟨number⟩ で表される要素を数値, ⟨variable⟩ で表される要素を変数と呼ぶ. 式の評価値を次のように定義する.

- 式が単一の数値のみからなる場合, この式の評価値はその数値に等しい.
- それ以外の場合, 式は `if_v_then_e1_else_e2` という形を取る. e_1 および e_2 は式である. また, v は変数名であり, 英小文字のみからなる文字列である. ここで, 各変数は真または偽の値を取る. この変数が真のとき, この式全体の評価値は e_1 の評価値に等しい. この変数が偽のとき, この式全体の評価値は e_2 の評価値に等しい.

たとえば, `if_a_then_if_b_then_1_else_2_else_3` という式には, a と b の 2 つの変数が出現する. 変数 a の値が真かつ変数 b の値が偽のとき, この式の評価値は 2 となる. また, 変数 a の値が偽のとき, 変数 b の値にかかわらず, この式の評価値は 3 となる.

式 e が与えられる. e 中に出現する異なる変数名が m 種類であるとき, それら m 個の変数それぞれに真または偽の値を割り当てる場合の数は, ちょうど 2^m 通り存在する. これら 2^m 通りの割り当てそれぞれについて e の評価値を求めたとき, その合計はいくつになるだろうか. 答えは大きくなることがあるので, 素数 998244353 で割った余りを出力せよ.

Input

入力は 1 つのデータセットからなり, 次の形式で表される.

e

e の長さは 10^6 を超えないことが保証される.

Output

すべての変数の割り当て方に対する e の評価値の合計を 998244353 で割った余りを, 1 行に出力せよ.

Sample Input 1

| |
|--|
| <code>if_a_then_if_b_then_1_else_2_else_3</code> |
|--|

Sample Output 1

| |
|---|
| 9 |
|---|

Sample Input 2

| |
|------------------------------------|
| <code>if_else_then_9_else_4</code> |
|------------------------------------|

Sample Output 2

| |
|----|
| 13 |
|----|

Sample Input 3

Sample Output 3

| | |
|--|----|
| if_x_then_if_y_then_2_else_7_else_if_y_then_1_else_8 | 18 |
|--|----|

2 番目の入力では, `else` という名前の変数が出現する. このような入力であっても曖昧性がないことが証明できる.

Problem F

Anti-BCC

Time Limit: 2 seconds

Japan Anti-BCC Group では、所属しているメンバー間でメッセージをやり取りする場合、メッセージは暗号化される。

暗号化されたメッセージは、正整数 n 、非負整数 a, b, c 、および 4 種類の文字 $?, A, B, C$ からなる長さ n の文字列 s 、という 5 つの要素で構成される。

Japan Anti-Japan-Anti-BCC-Group Group に所属するあなたは、暗号化されたメッセージを傍受し元のメッセージへ復号することにした。元のメッセージは、以下の条件をすべて満たす文字列の 1 つであることが分かっている。

- s に含まれるそれぞれの $?$ を、 A, B, C のいずれかに置き換えて得られる。
- A をちょうど a 個、 B をちょうど b 個、 C をちょうど c 個含む。
- 連続部分文字列として BCC を含まない。

元のメッセージとして考えられるものが 1 つ以上存在する場合はそのうち辞書順最小のものを求めよ。そのようなものが存在しない場合はそのことを報告せよ。

Input

入力は複数のデータセットからなる。各データセットは次の形式で表される。

$n\ a\ b\ c$
 s

n, a, b, c, s はあなたが傍受した暗号化されたメッセージを表し、 n は 1 以上 3,000 以下の正整数、 a, b, c は $a + b + c = n$ を満たす非負整数、 s は $?, A, B, C$ からなる長さ n の文字列である。

入力の終わりは 4 つのゼロからなる行で表される。また、データセットに含まれる n の和は 15,000 を超えない。

Output

各データセットに対し、元のメッセージとして考えられるものが 1 つ以上存在する場合はそのうち辞書順最小のものを出力し、そのようなものが存在しない場合は “NO” を出力せよ。

| Sample Input | Sample Output |
|--------------------|----------------|
| 7 2 2 3 A??B??C | AACBCBC ABC |
| 3 1 1 1 ??? | NO NO |
| 6 0 2 4 ?BC?CC | NO |
| 3 0 1 2 BCC | |
| 7 4 2 1 ??C?B?C | |
| 0 0 0 0 | |

Problem G

花子さんの芸術

Time Limit: 2 seconds

二次元平面上に、一辺の長さが n の正方形がある。この正方形の頂点は $(0,0)$, $(0,n)$, (n,n) , $(n,0)$ の 4 点である。また、この正方形の辺上に $2m$ 個の点があり、それぞれ黒または白で塗られている。

芸術家の花子さんは、この平面上に m 本の線分を引くことで、傑作を描くことにした。花子さんの美学によると、傑作とは次の条件をすべて満たすものである。

- いずれの線分も、 $2m$ 個の点のうちの 2 点を端点とする。さらに、この 2 点のうち、一方の点は黒、もう一方の点は白で塗られている。
- いずれの線分も、端点以外に正方形の辺との共有点を持たない。
- どの 2 本の線分も共有点を持たない。特に、どの 2 本の線分も端点を共有しない。

そのような絵が本当に素晴らしいものであるかはさておき、あなたには、 $2m$ 個の点の情報から、花子さんが描ける傑作が何種類あるかを求めてほしい。答えは大きくなるかもしれないので、素数 998244353 で割った余りを出力せよ。

なお、 m 本の線分は互いに区別しないものとする。したがって、線分上の座標全体からなる集合が異なるときかつそのときに限り、2 つの傑作を異なるものとみなす。

Input

入力は 1 つのデータセットからなり、次の形式で表される。

```
n m
x1 y1 c1
⋮
x2m y2m c2m
```

1 行目は、正方形の辺の長さを表す整数 n ($1 \leq n \leq 10^9$) と、花子さんが引くべき線分の本数 m ($1 \leq m \leq 300,000$) からなる。

続く $2m$ 行には正方形の辺上の点の情報が与えられる。 i 番目の点の情報は整数 x_i, y_i, c_i からなる。 (x_i, y_i) がこの点の座標である。したがって、 $0 \leq x_i \leq n, 0 \leq y_i \leq n$ であり、かつ $x_i = 0, x_i = n, y_i = 0, y_i = n$ のうち少なくとも 1 つを満たすことが保証される。 c_i は 0 または 1 であり、 $c_i = 0$ であればこの点が黒で塗られていることを、 $c_i = 1$ であればこの点が白で塗られていることを表す。

また、どの 2 点も同一座標にないことが保証される。すなわち、 $i \neq j$ であれば $(x_i, y_i) \neq (x_j, y_j)$ である。

Output

花子さんが描くことのできる傑作の種類の数 を 998244353 で割った余りを、1 行に出力せよ。

Sample Input 1

```
50 6
0 20 1
0 30 0
0 40 0
10 50 1
20 50 1
30 0 0
30 50 0
40 0 0
40 50 1
50 10 1
50 30 1
50 40 0
```

Sample Output 1

```
2
```

Sample Input 2

```
50 5
0 0 0
10 0 1
20 0 1
30 0 0
40 0 1
10 50 1
20 50 0
30 50 0
40 50 1
50 50 0
```

Sample Output 2

```
1
```

Sample Input 3

```
50 2
0 0 0
50 0 1
50 50 0
0 50 1
```

Sample Output 3

```
0
```

Problem H

順列の積

Time Limit: 2 seconds

整数 r と c が与えられる。 r 行 c 列の行列であって、以下の条件をすべて満たすものが存在するならば、1 つ求めよ。ただし、任意の i, j ($0 \leq i \leq r-1, 0 \leq j \leq c-1$) に対し、この行列の $i+1$ 行目、 $j+1$ 列目の要素を $a_{i,j}$ と表記する。番号が 0 から始まることに注意せよ。

- 各行には、0 以上 $c-1$ 以下の整数が 1 つずつ含まれる。
- 各列の要素の総積を c で割った余りは、その列の番号に等しい。すなわち、任意の j ($0 \leq j \leq c-1$) に対し、 $a_{0,j} \times a_{1,j} \times \cdots \times a_{r-1,j} \equiv j \pmod{c}$ が成り立つ。

Input

入力は複数のデータセットからなる。各データセットは次の形式で表される。

$r \ c$

r および c は整数であり、それぞれ $2 \leq r \leq 100,000$ および $2 \leq c \leq 100,000$ を満たす。

入力の終わりは 2 つのゼロからなる行で表される。すべてのデータセットの $r \times c$ の和は 500,000 を超えない。

Output

各データセットについて、条件を満たす行列が存在しなければ、1 行に “No” を出力せよ。存在すれば、1 行に “Yes” を出力し、続けて条件を満たす行列の 1 つを次の形式で出力せよ。

$a_{0,0} \ a_{0,1} \ \cdots \ a_{0,c-1}$
 $a_{1,0} \ a_{1,1} \ \cdots \ a_{1,c-1}$
 \vdots
 $a_{r-1,0} \ a_{r-1,1} \ \cdots \ a_{r-1,c-1}$

条件を満たす行列が複数存在する場合は、そのうちのどれを出力してもよい。

Sample Input

3 5
6 6
0 0

Sample Output

Yes
0 3 1 4 2
0 2 4 1 3
0 1 3 2 4
No

Problem I

最高のケーキを作ろう

Time Limit: 2 seconds

Japan Alumni Group でパーティーを行うこととなった。

パーティーのためにあなたはケーキを買った。このケーキは二次元平面上の座標 $(-10^{100}, -10^{100})$, $(10^{100}, -10^{100})$, $(10^{100}, 10^{100})$, $(-10^{100}, 10^{100})$ を 4 頂点とする正方形として表される。ケーキにはいくつかのイチゴが乗せられており、各イチゴは二次元平面上の点とみなすことができる。各イチゴには整数値のおいしさが定められている。

パーティーの参加者は皆イチゴが大好きなので、ケーキに乗っているイチゴのおいしさの和がケーキのおいしさを決めると考えている。

あなたはパーティーに出すケーキのおいしさ（ケーキに乗っているイチゴのおいしさの和）を最大化したいが、イチゴを取り除くとイチゴの跡が残って見栄えが良くない。そこで、あなたは以下の操作を 0 回以上繰り返すことにした。

- 直線を 1 つ選び、その直線に沿ってケーキを 2 つに分け、そのうち片方を捨てる。
 - ただし、イチゴを通る直線やケーキの真に内部を通らない直線は選ぶことができない。

操作後のケーキに乗っているイチゴのおいしさの和としてあり得る最大値を求めよ。

Input

入力は 1 つのデータセットからなり、次の形式で表される。

```
n
x1 y1 w1
x2 y2 w2
⋮
xn yn wn
```

1 行目には整数 n が与えられる。 n はイチゴの数であり、 $1 \leq n \leq 300$ を満たす。

続く n 行には、 n 個のイチゴの情報が与えられる。このうち i 番目の行には整数 x_i, y_i, w_i が与えられる。 (x_i, y_i) は i 個目のイチゴの座標であり、 $-100,000 \leq x_i, y_i \leq 100,000$ を満たす。 w_i は i 個目のイチゴのおいしさの値であり、 $-10^9 \leq w_i \leq 10^9$ を満たす。

また、どの 2 個のイチゴも同一座標にないことが保証される。すなわち、 $i \neq j$ であれば $(x_i, y_i) \neq (x_j, y_j)$ である。

Output

操作後のケーキに乗っているイチゴのおいしさの和としてあり得る最大値を 1 行に出力せよ。

Sample Input 1

```
5
0 0 5
10 0 10
5 5 -1
0 10 20
10 10 -10
```

Sample Output 1

```
34
```

Sample Input 2

```
5
1 1 -5
3 3 10
5 5 -4
7 7 7
9 9 -2
```

Sample Output 2

```
13
```

Sample Input 3

```
1
0 0 -100
```

Sample Output 3

```
0
```