

問題文はコンテスト開始時点でのものになります。

Clar等による変更は反映されませんので、コンテストサイトを随時参照してください。

The problem statement is current as of the start of the contest.

Please refer to the contest site from time to time as changes due to clarification, etc. will not be reflected.

どのような投稿がルールに違反するかはこちらの記事 (/posts/262) もお読みください。

A - Overall Winner

実行時間制限: 2 sec / メモリ制限: 1024 MB

配点: 100 点

問題文

高橋くんと青木くんが N 回の試合を行いました。これらの試合の結果を表す長さ N の文字列 S が与えられます。 i 回目の試合の勝者は、 S の i 文字目が T ならば高橋くん、A ならば青木くんです。

高橋くんと青木くんのうち、勝った試合の数が多い方を総合勝者とします。ただし、勝った試合の数が同じである場合は、先にその勝ち数に達した者を総合勝者とします。高橋くんと青木くんのどちらが総合勝者であるか求めてください。

制約

- $1 \leq N \leq 100$
- N は整数
- S は T および A からなる長さ N の文字列

入力

入力は以下の形式で標準入力から与えられる。

```
N
S
```

出力

総合勝者が高橋くんならば T を、青木くんならば A を出力せよ。

入力例 1

```
5
TTAAT
```

出力例 1

T

高橋くんは 3 回の試合に勝ち、青木くんは 2 回の試合に勝ちました。 よって、勝った試合の数が多い高橋くんが総合勝者です。

入力例 2

6
ATTATA

出力例 2

T

高橋くんと青木くんのどちらも 3 回の試合に勝ちました。 また、高橋くんは 5 回目の試合で 3 勝目に達し、青木くんは 6 回目の試合で 3 勝目に達しました。 よって、先に 3 勝目に達した高橋くんが総合勝者です。

入力例 3

1
A

出力例 3

A

B - Fill the Gaps

実行時間制限: 2 sec / メモリ制限: 1024 MB

配点 : 200 点

問題文

正整数からなる長さ N の数列 $A = (A_1, \dots, A_N)$ があります。どの隣接する 2 項の値も相異なります。

この数列に対し、次の操作によりいくつか数を挿入します。

- 数列 A のどの隣接する 2 項の差の絶対値も 1 であるとき、操作を終了する。
- 数列 A の先頭から見て、隣接する 2 項の差の絶対値が 1 でない最初の箇所を A_i, A_{i+1} とする。
 - $A_i < A_{i+1}$ ならば、 A_i と A_{i+1} の間に、 $A_i + 1, A_i + 2, \dots, A_{i+1} - 1$ を挿入する。
 - $A_i > A_{i+1}$ ならば、 A_i と A_{i+1} の間に、 $A_i - 1, A_i - 2, \dots, A_{i+1} + 1$ を挿入する。
- 手順 1 に戻る。

操作が終了したときの数列を出力してください。

制約

- $2 \leq N \leq 100$
- $1 \leq A_i \leq 100$
- $A_i \neq A_{i+1}$
- 入力は全て整数

入力

入力は以下の形式で標準入力から与えられる。

```
N
A_1  A_2  ...  A_N
```

出力

操作が終了したときの数列の各要素を空白区切りで出力せよ。

入力例 1

```
4
2 5 1 2
```

出力例 1

2 3 4 5 4 3 2 1 2

最初、数列は (2, 5, 1, 2) です。操作は以下のように行われます。

- 1 項目の 2 と 2 項目の 5 の間に 3, 4 を挿入する。数列は (2, 3, 4, 5, 1, 2) となる。
- 4 項目の 5 と 5 項目の 1 の間に 4, 3, 2 を挿入する。数列は (2, 3, 4, 5, 4, 3, 2, 1, 2) となる。

入力例 2

6
3 4 5 6 5 4

出力例 2

3 4 5 6 5 4

一度も挿入が行われないこともあります。

C - AtCoder Cards

実行時間制限: 2 sec / メモリ制限: 1024 MB

配点 : 300 点

問題文

AtCoder社ではカードを使った1人ゲームが流行っています。

ゲームで使う各カードには、英小文字1文字または@の文字が書かれており、いずれのカードも十分多く存在します。

ゲームは以下の手順で行います。

1. カードを同じ枚数ずつ2列に並べる。
2. @のカードを、それぞれa, t, c, o, d, e, rのいずれかのカードと置き換える。
3. 2つの列が一致していれば勝ち。そうでなければ負け。

このゲームに勝ちたいあなたは、次のようなイカサマをすることにしました。

- 手順1以降の好きなタイミングで、列内のカードを自由に並び替えてよい。

手順1で並べられた2つの列を表す2つの文字列 S, T が与えられるので、イカサマをしてもよいときゲームに勝てるか判定してください。

制約

- S, T は英小文字と@からなる
- S, T の長さは等しく1以上 2×10^5 以下

入力

入力は以下の形式で標準入力から与えられる。

S
 T

出力

イカサマをしてもよいとき、ゲームに勝てるならYes、勝てないならNoと出力せよ。

入力例 1

ch@ku@ai
choku@@i

出力例 1

Yes

@をうまく置き換えることによって、両方とも chokudai と一致させることが可能です。

入力例 2

ch@kud@i
akidu@ho

出力例 2

Yes

イカサマをし、@をうまく置き換えることによって、両方とも chokudai と一致させることが可能です。

入力例 3

aoki
@ok@

出力例 3

No

イカサマをしても勝つことはできません。

入力例 4

aa
bb

出力例 4

No

D - Bitmask

実行時間制限: 2 sec / メモリ制限: 1024 MB

配点 : 400 点

問題文

$0, 1, ?$ からなる文字列 S および整数 N が与えられます。 S に含まれる $?$ をそれぞれ 0 または 1 に置き換えて 2 進整数とみなしたときに得られる値の集合を T とします。たとえば、 $S = ?0?$ のとき、 $T = \{000_{(2)}, 001_{(2)}, 100_{(2)}, 101_{(2)}\} = \{0, 1, 4, 5\}$ です。

T に含まれる N 以下の値のうち最大のものを (10 進整数として) 出力してください。 N 以下の値が T に含まれない場合は、代わりに -1 を出力してください。

制約

- S は $0, 1, ?$ からなる文字列
- S の長さは 1 以上 60 以下
- $1 \leq N \leq 10^{18}$
- N は整数

入力

入力は以下の形式で標準入力から与えられる。

```
S
N
```

出力

答えを出力せよ。

入力例 1

```
?0?
2
```

出力例 1

1

問題文中で例示したとおり、 $T = \{0, 1, 4, 5\}$ です。 T に含まれる N 以下の値は 0 と 1 なので、そのうちの最大である 1 を出力します。

入力例 2

101
4

出力例 2

-1

$T = \{5\}$ であるため、 N 以下の値は T に含まれません。

入力例 3

?0?
10000000000000000000

出力例 3

5

E - Pac-Takahashi

実行時間制限: 5 sec / メモリ制限: 1024 MB

配点: 475 点

問題文

H 行 W 列のグリッドがあります。上から i 行目、左から j 列目のマス目を (i, j) と表します。グリッドの各マスはスタートマス、ゴールマス、空マス、壁マス、お菓子マスのいずれかです。 (i, j) が何のマスであるかは文字 $A_{i,j}$ によって表され、 $A_{i,j} = S$ のときスタートマス、 $A_{i,j} = G$ のときゴールマス、 $A_{i,j} = .$ のとき空マス、 $A_{i,j} = \#$ のとき壁マス、 $A_{i,j} = o$ のときお菓子マスです。ここで、スタートマスとゴールマスはちょうど 1 つずつあり、お菓子マスは 18 個以下であることが保証されます。

高橋くんは現在スタートマスにいます。高橋くんは、上下左右に隣接するマスであって壁マスでないマスに移動することを繰り返し行えます。高橋くんは今から T 回以下の移動によってゴールマスに到達したいです。そのようなことは可能かどうか判定してください。可能な場合は、最終的にゴールマスにいるという条件のもとで、移動の途中に訪れるお菓子マスの数の最大値を求めてください。ただし、1 つのお菓子マスに複数回訪れた場合でも、カウントするのは 1 回のみです。

制約

- $1 \leq H, W \leq 300$
- $1 \leq T \leq 2 \times 10^6$
- H, W, T は整数
- $A_{i,j}$ は S, G, ., #, o のいずれか
- $A_{i,j} = S$ を満たす (i, j) の組がちょうど 1 つ存在する
- $A_{i,j} = G$ を満たす (i, j) の組がちょうど 1 つ存在する
- $A_{i,j} = o$ を満たす (i, j) の組は 18 個以下

入力

入力は以下の形式で標準入力から与えられる。

```
H W T
A1,1 A1,2 ... A1,W
⋮
AH,1 AH,2 ... AH,W
```

出力

T 回以下の移動によってゴールマスに到達することが不可能ならば -1 を出力せよ。可能ならば、最終的にゴールマスにいるという条件のもとで、移動の途中に訪れるお菓子マスの数の最大値を出力せよ。

入力例 1

```
3 3 5
S.G
o#o
.#.
```

出力例 1

```
1
```

$(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3) \rightarrow (1, 3)$ と 4 回移動すると、1 個のお菓子マスを訪れた上で最終的にゴールマスにすることができます。5 回以下の移動で 2 個のお菓子マスを訪れた上で最終的にゴールマスにすることはできないので、1 が答えです。

なお、 $(1, 1) \rightarrow (2, 1) \rightarrow (1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3)$ と移動すると 5 回の移動で 2 個のお菓子マスを訪れることができますが、最終的にゴールマスにいないため無効であることに注意してください。

入力例 2

```
3 3 1
S.G
.#o
o#.
```

出力例 2

```
-1
```

1 回以下の移動でゴールマスに到達することはできません。

入力例 3

```
5 10 2000000
S.o..ooo..
..o..o.o..
..o..ooo..
..o..o.o..
..o..ooo.G
```

出力例 3

```
18
```

F - Anti-DDoS

実行時間制限: 2 sec / メモリ制限: 1024 MB

配点 : 500 点

問題文

英大文字・英小文字からなる長さ 4 の文字列で、以下の 2 条件をともに満たすものを DDoS 型文字列と呼ぶことにします。

- 1, 2, 4 文字目が英大文字で、3 文字目が英小文字である
- 1, 2 文字目が等しい

例えば DDoS, AAaA は DDoS 型文字列であり、ddos, IPoE は DDoS 型文字列ではありません。

英大文字・英小文字および ? からなる文字列 S が与えられます。 S に含まれる ? を独立に英大文字・英小文字に置き換えてできる文字列は、 S に含まれる ? の個数を q として 52^q 通りあります。このうち DDoS 型文字列を部分列に含まないものの個数を 998244353 で割ったあまりを求めてください。

注記

文字列の**部分列**とは、文字列から 0 個以上の文字を取り除いた後、残りの文字を元の順序で連結して得られる文字列のことをいいます。

例えば、AC は ABC の部分列であり、RE は ECR の部分列ではありません。

制約

- S は英大文字・英小文字および ? からなる
- S の長さは 4 以上 3×10^5 以下

入力

入力は以下の形式で標準入力から与えられる。

S

出力

答えを出力せよ。

入力例 1

DD??S

出力例 1

676

? の少なくとも一方が英小文字のとき、DDoS 型文字列を部分列に含みます。

入力例 2

??

出力例 2

858572093

998244353 で割ったあまりを求めてください。

入力例 3

?D??S

出力例 3

136604

G - Worst Picture

実行時間制限: 4 sec / メモリ制限: 1024 MB

配点 : 600 点

問題文

3次元空間に N 人の人がいます。 i 番目の人は座標 (X_i, Y_i, Z_i) にいます。

人のいる座標は相異なり、全ての i で $X_i > 0$ です。

あなたは $x < 0$ であるような点 $p = (x, y, z)$ を選び、そこから x 軸正の方向を向いて写真を撮ります。

点 p と人のいる場所 A, B が、 p, A, B の順に一直線に並ぶとき、 B にいる人は写真に写りません。これ以外に人が写真に写らなくなることはありません。

p を適切に選んだ時の、写真に写る人数の最小値を求めてください。

制約

- $1 \leq N \leq 50$
- $0 < X_i \leq 1000$
- $-1000 \leq Y_i, Z_i \leq 1000$
- (X_i, Y_i, Z_i) は相異なる
- 入力は全て整数

入力

入力は以下の形式で標準入力から与えられる。

```
N
X_1 Y_1 Z_1
⋮
X_N Y_N Z_N
```

出力

答えを出力せよ。

入力例 1

3
1 1 1
2 2 2
100 99 98

出力例 1

2

例えば、点 $(-0.5, -0.5, -0.5)$ から写真を撮ると、2 番目の人は写真に写りません。

入力例 2

8
1 1 1
1 1 -1
1 -1 1
1 -1 -1
3 2 2
3 2 -2
3 -2 2
3 -2 -2

出力例 2

4

点 $(-1, 0, 0)$ から写真を撮ると、写る人数は 4 人になります。

Ex - Difference of Distance

実行時間制限: 5 sec / メモリ制限: 1024 MB

配点 : 625 点

問題文

頂点には 1 から N の番号が、辺には 1 から M の番号がついた N 頂点 M 辺の連結無向グラフがあります。辺 i は頂点 U_i と頂点 V_i を結んでおり、重みとして整数 W_i が定められています。ここで、 $1 \leq s, t \leq N$, $s \neq t$ に対して $d(s, t)$ を以下で定義します。

- 頂点 s と頂点 t を結ぶすべてのパスに対する「パス上の辺の重みの最大値」の最小値

今から与えられる Q 個のクエリにそれぞれ答えてください。 j 番目のクエリは以下の通りです。

- A_j, S_j, T_j が与えられる。辺 A_j の重みを 1 増やすと $d(S_j, T_j)$ がいくつ変化するか求めよ。

なお、各クエリにおいて実際には辺の重みは変更しないことに注意してください。

制約

- $2 \leq N \leq 2 \times 10^5$
- $N - 1 \leq M \leq 2 \times 10^5$
- $1 \leq U_i, V_i \leq N$
- $U_i \neq V_i$
- $1 \leq W_i \leq M$
- 与えられるグラフは連結
- $1 \leq Q \leq 2 \times 10^5$
- $1 \leq A_j \leq M$
- $1 \leq S_j, T_j \leq N$
- $S_j \neq T_j$
- 入力は全て整数

入力

入力は以下の形式で標準入力から与えられる。

```

$$\begin{array}{l} N \quad M \\ U_1 \quad V_1 \quad W_1 \\ \vdots \\ U_M \quad V_M \quad W_M \\ Q \\ A_1 \quad S_1 \quad T_1 \\ \vdots \\ A_Q \quad S_Q \quad T_Q \end{array}$$

```

出力

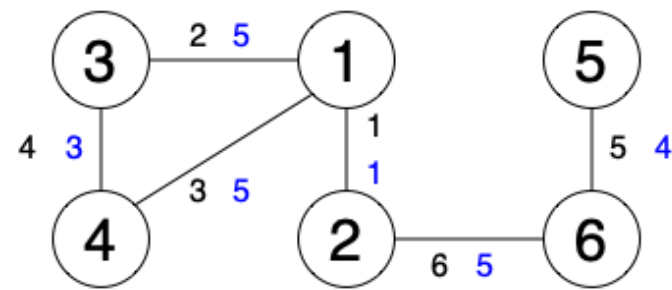
Q 行出力せよ。 j ($1 \leq j \leq Q$) 行目には、 j 番目のクエリに対する答えを出力せよ。

入力例 1

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

出力例 1

```
0
0
0
0
0
1
1
```



上の図においては、辺の番号を黒字で、辺の重みを青字で表記しています。

1 番目から 6 番目のクエリについて説明します。

まず与えられたグラフにおける $d(4, 6)$ について考えます。 $4 \rightarrow 1 \rightarrow 2 \rightarrow 6$ というパス上の辺の重みの最大値は 5 ですが、これは頂点 4 と頂点 6 を結ぶパスの中での最小値であるため、 $d(4, 6) = 5$ です。

次に、辺 x ($1 \leq x \leq 6$) の重みを 1 増やすと $d(4, 6)$ がいくつ変化するか考えます。 $x = 6$ のとき $d(4, 6) = 6$ となるため変化量は 1 ですが、 $x \neq 6$ のときは $d(4, 6) = 5$ となるため変化量は 0 です。たとえば $x = 3$ のとき、 $4 \rightarrow 1 \rightarrow 2 \rightarrow 6$ というパス上の辺の重みの最大値は 6 になりますが、 $4 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 6$ というパス上の辺の重みの最大値が 5 であるため $d(4, 6)$ は変わらず 5 のままです。

入力例 2

```
2 2
1 2 1
1 2 1
1
1 1 2
```

出力例 2

```
0
```

与えられるグラフは多重辺を含むこともあります。

A - Overall Winner

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

Takahashi and Aoki played N games. You are given a string S of length N , representing the results of these games. Takahashi won the i -th game if the i -th character of S is T, and Aoki won that game if it is A.

The overall winner between Takahashi and Aoki is the one who won more games than the other. If they had the same number of wins, the overall winner is the one who reached that number of wins first. Find the overall winner: Takahashi or Aoki.

Constraints

- $1 \leq N \leq 100$
- N is an integer.
- S is a string of length N consisting of T and A.

Input

The input is given from Standard Input in the following format:

```
 $N$   
 $S$ 
```

Output

If the overall winner is Takahashi, print T; if it is Aoki, print A.

Sample Input 1

```
5  
TTAAT
```

Sample Output 1

T

Takahashi won three games, and Aoki won two. Thus, the overall winner is Takahashi, who won more games.

Sample Input 2

6
ATTATA

Sample Output 2

T

Both Takahashi and Aoki won three games. Takahashi reached three wins in the fifth game, and Aoki in the sixth game. Thus, the overall winner is Takahashi, who reached three wins first.

Sample Input 3

1
A

Sample Output 3

A

B - Fill the Gaps

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 200 points

Problem Statement

We have a sequence of length N consisting of positive integers: $A = (A_1, \dots, A_N)$. Any two adjacent terms have different values.

Let us insert some numbers into this sequence by the following procedure.

1. If every pair of adjacent terms in A has an absolute difference of 1, terminate the procedure.
2. Let A_i, A_{i+1} be the pair of adjacent terms nearest to the beginning of A whose absolute difference is not 1.
 - If $A_i < A_{i+1}$, insert $A_i + 1, A_i + 2, \dots, A_{i+1} - 1$ between A_i and A_{i+1} .
 - If $A_i > A_{i+1}$, insert $A_i - 1, A_i - 2, \dots, A_{i+1} + 1$ between A_i and A_{i+1} .
3. Return to step 1.

Print the sequence when the procedure ends.

Constraints

- $2 \leq N \leq 100$
- $1 \leq A_i \leq 100$
- $A_i \neq A_{i+1}$
- All values in the input are integers.

Input

The input is given from Standard Input in the following format:

N
 S

Output

Print the terms in the sequence when the procedure ends, separated by spaces.

Sample Input 1

```
4
2 5 1 2
```

Sample Output 1

```
2 3 4 5 4 3 2 1 2
```

The initial sequence is (2, 5, 1, 2). The procedure goes as follows.

- Insert 3, 4 between the first term 2 and the second term 5, making the sequence (2, 3, 4, 5, 1, 2).
 - Insert 4, 3, 2 between the fourth term 5 and the fifth term 1, making the sequence (2, 3, 4, 5, 4, 3, 2, 1, 2).
-

Sample Input 2

```
6
3 4 5 6 5 4
```

Sample Output 2

```
3 4 5 6 5 4
```

No insertions may be performed.

C - AtCoder Cards

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 300 points

Problem Statement

A single-player card game is popular in AtCoder Inc. Each card in the game has a lowercase English letter or the symbol @ written on it. There is plenty number of cards for each kind. The game goes as follows.

1. Arrange the same number of cards in two rows.
2. Replace each card with @ with one of the following cards: a, t, c, o, d, e, r.
3. If the two rows of cards coincide, you win. Otherwise, you lose.

To win this game, you will do the following cheat.

- Freely rearrange the cards within a row whenever you want after step 1.

You are given two strings S and T , representing the two rows you have after step 1. Determine whether it is possible to win with cheating allowed.

Constraints

- S and T consist of lowercase English letters and @.
- The lengths of S and T are equal and between 1 and 2×10^5 , inclusive.

Input

The input is given from Standard Input in the following format:

S
 T

Output

If it is possible to win with cheating allowed, print Yes; otherwise, print No.

Sample Input 1

ch@ku@ai
choku@@i

Sample Output 1

Yes

You can replace the @s so that both rows become chokudai.

Sample Input 2

ch@kud@i
akidu@ho

Sample Output 2

Yes

You can cheat and replace the @s so that both rows become chokudai.

Sample Input 3

aoki
@ok@

Sample Output 3

No

You cannot win even with cheating.

Sample Input 4

aa
bb

Sample Output 4

No

D - Bitmask

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 400 points

Problem Statement

You are given an integer N and a string S consisting of 0, 1, and ?. Let T be the set of values that can be obtained by replacing each ? in S with 0 or 1 and interpreting the result as a binary integer. For instance, if $S = ?0?$, we have $T = \{000_{(2)}, 001_{(2)}, 100_{(2)}, 101_{(2)}\} = \{0, 1, 4, 5\}$.

Print (as a decimal integer) the greatest value in T less than or equal to N . If T does not contain a value less than or equal to N , print -1 instead.

Constraints

- S is a string consisting of 0, 1, and ?.
- The length of S is between 1 and 60, inclusive.
- $1 \leq N \leq 10^{18}$
- N is an integer.

Input

The input is given from Standard Input in the following format:

```
 $S$   
 $N$ 
```

Output

Print the answer.

Sample Input 1

```
?0?  
2
```

Sample Output 1

```
1
```

As shown in the problem statement, $T = \{0, 1, 4, 5\}$. Among them, 0 and 1 are less than or equal to N , so you should print the greatest of them, 1.

Sample Input 2

```
101
4
```

Sample Output 2

```
-1
```

We have $T = \{5\}$, which does not contain a value less than or equal to N .

Sample Input 3

```
?0?
10000000000000000000
```

Sample Output 3

```
5
```

E - Pac-Takahashi

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 475 points

Problem Statement

We have a grid with H rows and W columns. Let (i, j) denote the square at the i -th row from the top and j -th column from the left. Each square in the grid is one of the following: the start square, the goal square, an empty square, a wall square, and a candy square. (i, j) is represented by a character $A_{i,j}$, and is the start square if $A_{i,j} = \text{S}$, the goal square if $A_{i,j} = \text{G}$, an empty square if $A_{i,j} = \text{.}$, a wall square if $A_{i,j} = \text{\#}$, and a candy square if $A_{i,j} = \text{o}$. Here, it is guaranteed that there are exactly one start, exactly one goal, and **at most** 18 candy squares.

Takahashi is now at the start square. He can repeat moving to a vertically or horizontally adjacent non-wall square. He wants to reach the goal square in at most T moves. Determine whether it is possible. If it is possible, find the maximum number of candy squares he can visit on the way to the goal square, where he must finish. Each candy square counts only once, even if it is visited multiple times.

Constraints

- $1 \leq H, W \leq 300$
- $1 \leq T \leq 2 \times 10^6$
- H, W , and T are integers.
- $A_{i,j}$ is one of S, G, ., #, and o.
- Exactly one pair (i, j) satisfies $A_{i,j} = \text{S}$.
- Exactly one pair (i, j) satisfies $A_{i,j} = \text{G}$.
- At most** 18 pairs (i, j) satisfy $A_{i,j} = \text{o}$.

Input

The input is given from Standard Input in the following format:

```
H W T
A1,1A1,2...A1,W
⋮
AH,1AH,2...AH,W
```

Output

If it is impossible to reach the goal square in at most T moves, print -1 . Otherwise, print the maximum number of candy squares that can be visited on the way to the goal square, where Takahashi must finish.

Sample Input 1

```
3 3 5
S.G
o#o
.#.
```

Sample Output 1

```
1
```

If he makes four moves as $(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3) \rightarrow (1, 3)$, he can visit one candy square and finish at the goal square. He cannot make five or fewer moves to visit two candy squares and finish at the goal square, so the answer is 1.

Note that making five moves as $(1, 1) \rightarrow (2, 1) \rightarrow (1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3)$ to visit two candy squares is invalid since he would not finish at the goal square.

Sample Input 2

```
3 3 1
S.G
.#o
o#.
```

Sample Output 2

```
-1
```

He cannot reach the goal square in one or fewer moves.

Sample Input 3

```
5 10 2000000
S.o..ooo..
..o..o.o..
..o..ooo..
..o..o.o..
..o..ooo.G
```

Sample Output 3

```
18
```

F - Anti-DDoS

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 500 points

Problem Statement

A DDoS-type string is a string of length 4 consisting of uppercase and lowercase English letters satisfying both of the following conditions.

- The first, second, and fourth characters are uppercase English letters, and the third character is a lowercase English letter.
- The first and second characters are equal.

For instance, DDoS and AAaA are DDoS-type strings, while neither ddos nor IPoE is.

You are given a string S consisting of uppercase and lowercase English letters and ?. Let q be the number of occurrences of ? in S . There are 52^q strings that can be obtained by independently replacing each ? in S with an uppercase or lowercase English letter. Among these strings, find the number of ones that do not contain a DDoS-type string as a subsequence, modulo 998244353.

Notes

A **subsequence** of a string is a string obtained by removing zero or more characters from the string and concatenating the remaining characters without changing the order.

For instance, AC is a subsequence of ABC, while RE is not a subsequence of ECR.

Constraints

- S consists of uppercase English letters, lowercase English letters, and ?.
- The length of S is between 4 and 3×10^5 , inclusive.

Input

The input is given from Standard Input in the following format:

S

Output

Print the answer.

Sample Input 1

DD??S

Sample Output 1

676

When at least one of the ?s is replaced with a lowercase English letter, the resulting string will contain a DDoS-type string as a subsequence.

Sample Input 2

??

Sample Output 2

858572093

Find the count modulo 998244353.

Sample Input 3

?D??S

Sample Output 3

136604

G - Worst Picture

Time Limit: 4 sec / Memory Limit: 1024 MB

Score : 600 points

Problem Statement

There are N people in a three-dimensional space. The i -th person is at the coordinates (X_i, Y_i, Z_i) . All people are at different coordinates, and we have $X_i > 0$ for every i .

You will choose a point $p = (x, y, z)$ such that $x < 0$, and take a photo in the positive x -direction.

If the point p and the positions A, B of two people are on the same line in the order p, A, B , then the person at B will not be in the photo. There are no other potential obstacles.

Find the number of people in the photo when p is chosen to minimize this number.

Constraints

- $1 \leq N \leq 50$
- $0 < X_i \leq 1000$
- $-1000 \leq Y_i, Z_i \leq 1000$
- The triples (X_i, Y_i, Z_i) are distinct.
- All values in the input are integers.

Input

The input is given from Standard Input in the following format:

```
N
X1 Y1 Z1
⋮
XN YN ZN
```

Output

Print the answer.

Sample Input 1

```
3
1 1 1
2 2 2
100 99 98
```

Sample Output 1

```
2
```

For instance, if you take the photo from the point $(-0.5, -0.5, -0.5)$, it will not show the second person.

Sample Input 2

```
8
1 1 1
1 1 -1
1 -1 1
1 -1 -1
3 2 2
3 2 -2
3 -2 2
3 -2 -2
```

Sample Output 2

```
4
```

If you take the photo from the point $(-1, 0, 0)$, it will show four people.

Ex - Difference of Distance

Time Limit: 5 sec / Memory Limit: 1024 MB

Score : 625 points

Problem Statement

We have a connected undirected graph with N vertices numbered 1 to N and M edges numbered 1 to M . Edge i connects vertex U_i and vertex V_i , and has an integer weight of W_i . For $1 \leq s, t \leq N$, $s \neq t$, let us define $d(s, t)$ as follows.

- For every path connecting vertex s and vertex t , consider the maximum weight of an edge along that path. $d(s, t)$ is the minimum value of this weight.

Answer Q queries. The j -th query is as follows.

- You are given A_j, S_j, T_j . By what value will $d(S_j, T_j)$ increase when the weight of edge A_j is increased by 1?

Note that each query does not actually change the weight of the edge.

Constraints

- $2 \leq N \leq 2 \times 10^5$
 - $N - 1 \leq M \leq 2 \times 10^5$
 - $1 \leq U_i, V_i \leq N$
 - $U_i \neq V_i$
 - $1 \leq W_i \leq M$
 - The given graph is connected.
 - $1 \leq Q \leq 2 \times 10^5$
 - $1 \leq A_j \leq M$
 - $1 \leq S_j, T_j \leq N$
 - $S_j \neq T_j$
 - All values in the input are integers.
-

Input

The input is given from Standard Input in the following format:

```
 $N$   $M$   
 $U_1$   $V_1$   $W_1$   
 $\vdots$   
 $U_M$   $V_M$   $W_M$   
 $Q$   
 $A_1$   $S_1$   $T_1$   
 $\vdots$   
 $A_Q$   $S_Q$   $T_Q$ 
```

Output

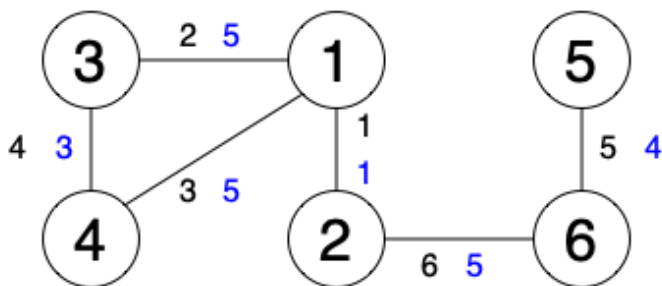
Print Q lines. The j -th line ($1 \leq j \leq Q$) should contain the answer to the j -th query.

Sample Input 1

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

Sample Output 1

```
0
0
0
0
0
1
1
```



The above figure shows the edge numbers in black and the edge weights in blue.

Let us explain the first through sixth queries.

First, consider $d(4, 6)$ in the given graph. The maximum weight of an edge along the path $4 \rightarrow 1 \rightarrow 2 \rightarrow 6$ is 5, which is the minimum over all paths connecting vertex 4 and vertex 6, so $d(4, 6) = 5$.

Next, consider the increment of $d(4, 6)$ when the weight of edge x ($1 \leq x \leq 6$) increases by 1. When $x = 6$, we have $d(4, 6) = 6$, and the increment is 1. On the other hand, when $x \neq 6$, we have $d(4, 6) = 5$, and the increment is 0. For instance, when $x = 3$, the maximum weight of an edge along the path $4 \rightarrow 1 \rightarrow 2 \rightarrow 6$ is 6, but the maximum weight of an edge along the path $4 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 6$ is 5, so $d(4, 6)$ is still 5.

Sample Input 2

```
2 2
1 2 1
1 2 1
1
1 1 2
```

Sample Output 2

```
0
```

The given graph may contain multi-edges.