

Problem A

ブービー賞

Time Limit: 2 seconds

あなたはボウリング大会の景品担当を任された。このボウリング大会では、ボウリングが苦手な人の参加意欲を高めるため、スコアが低い方から 2 番目の人にブービー賞を与えることになっている。

あなたの仕事は与えられた参加者全員のスコアから、ブービー賞を与える人のスコアを求めることである。

なお、景品担当にとっては幸いなことに、全員のスコアは全て異なっていた。

Input

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

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

1 行目にボウリング大会の参加者数を表す整数 N ($2 \leq N \leq 100$) が与えられる。

2 行目に N 個の整数 $a_1, \dots, a_N$ ($0 \leq a_i \leq 300$) が与えられ、 i 人目の参加者のスコアが a_i であることを表す。 $a_1, \dots, a_N$ はすべて相異なる。

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、ブービー賞を与える人のスコアを 1 行に出力せよ。

Sample Input

```
5
1 2 3 4 5
4
153 271 15 101
2
300 0
0
```

Sample Output

```
2
101
300
```

Problem B

Rearrange

Time Limit: 2 seconds

JAG に所属する JAG 郎君は、模擬国内予選に向けて問題セットを準備している。ところが、現在作業している問題のタイトルが、昨年出題した問題のタイトルと完全に重複していることに気が付いた。そこで、タイトルの文字を並び替えて、元のタイトルとは異なる新しいタイトルを作ることにした。

問題のタイトルの長さ N と、タイトルを表す英小文字からなる文字列 S が与えられる。 S の文字を並び替えて、元の文字列 S とは異なる文字列を作ることが可能か判定せよ。可能な場合は条件を満たす文字列を 1 つ出力し、不可能な場合は IMPOSSIBLE を出力せよ。

Input

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

N
 S

1 行目に整数 N ($1 \leq N \leq 100$) が与えられる。2 行目に英小文字からなる長さ 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、条件を満たす文字列を作ることが可能な場合はそのような文字列のうち 1 つを出力せよ。不可能な場合は IMPOSSIBLE を出力せよ。

Sample Input	Sample Output
9 rearrange 3 jag 1 x 5 aaaab 0	egnarraer gaj IMPOSSIBLE baaaa

Problem C

Loooong Vacation

Time Limit: 2 seconds

あなたは長ーい休みが大好きである。有給休暇を使うことで、できるだけ長い連休を作ろうと考えた。

o , $_$, x のみからなる長さ N の文字列 S が与えられる。 i 番目の文字はあなたの i 日後の予定を表していて、それぞれ以下を意味する。

- o : 休みの日
- $_$: 仕事の日（有給休暇を使うことで休める）
- x : 休めない日

あなたは有給休暇を K 日持っており、 $_$ を K 個まで選んで o に変えることで、休みにすることができる。適切に有給休暇を使ったとき、 N 日間のうち最大で何日連続で休めるかを求めよ。

Input

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

N K
 S

1 行目に文字列の長さ N ($1 \leq N \leq 5 \times 10^5$) と、持っている有給休暇の日数 K ($0 \leq K \leq N$) が与えられる。

2 行目に o , $_$, x 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、2 個のゼロ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 10^6 を超えない。

Output

各テストケースについて、連続で休める最大日数を 1 行に出力せよ。

Sample Input

```
14 2
oo__ox_ooooox_
7 5
ox__xoo
6 0
o_oxo_
0 0
```

Sample Output

```
6
2
1
```

Problem D Box Tower

Time Limit: 3 seconds

同じ大きさの立方体の箱が N 個ある。箱の各面には色が塗られており、各色は 1 以上 $6N$ 以下の整数で表される。ここからいくつかの箱を選び、それぞれを回転させた後、縦に積み上げることで、 1 本の箱のタワーを作ろうとしている。各箱は回転させて向きを決めた後、すべての箱の上面と下面が水平になるように置き、底面がぴったり重なるように真上に積み上げるものとする。

箱のタワーの美しさは、タワーの各側面に使われる色の系統がまとまっているかで決まる。ここでは、次の条件を満たすものを「美しいタワー」と呼ぶ：

- タワーの 4 つの側面に $1, 2, 3, 4$ の番号を付ける。側面 i に現れる色の最小値を m_i 、最大値を M_i とする。このとき、任意の $1 \leq i < j \leq 4$ について、区間 $[m_i, M_i]$ と $[m_j, M_j]$ が共通部分を持たないならば、そのタワーは美しい。

Figure D-1 に、サンプルの 1 つ目のテストケースの箱を用いて作れる、美しいタワーの例と美しいでないタワーの例を示す。なお、図中の数値は各面に塗られた色を表す。左の美しいタワーでは、各側面に使われている色の範囲の集合は $\{[1, 3], [5, 8], [9, 11], [13, 15]\}$ であり、どの 2 つの区間も共通部分を持たない。一方、右の美しいでないタワーでは、その集合は $\{[2, 6], [5, 8], [9, 12], [13, 15]\}$ であり、 $[2, 6]$ と $[5, 8]$ が共通部分を持つため美しいタワーではない。

タワーの高さを、そのタワーに使われている箱の個数としたとき、美しいタワーの高さの最大値を求めよ。

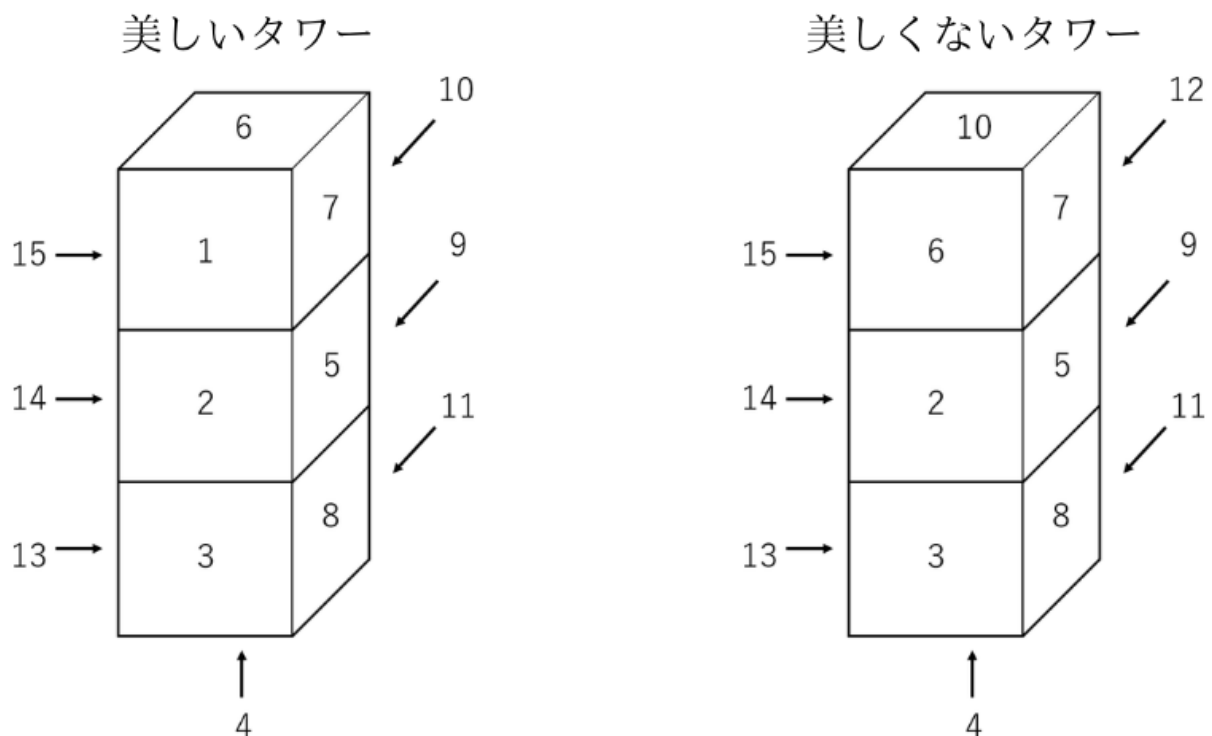


Figure D-1: 美しいタワーの例（左）と美しいでないタワーの例（右）

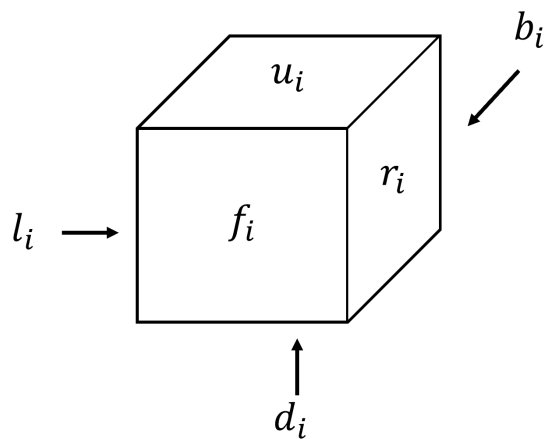


Figure D-2: 入力と各面の対応例

Input

入力は 10 個以内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。

$$\begin{array}{l} N \\ u_1 \ d_1 \ f_1 \ b_1 \ l_1 \ r_1 \\ \vdots \\ u_N \ d_N \ f_N \ b_N \ l_N \ r_N \end{array}$$

1 行目に箱の個数 N が与えられる ($1 \leq N \leq 6$)。

続く N 行には、箱の面の色情報が与えられる。 i 番目の箱の各面の色は、上下前後左右の順に、 $u_i, d_i, f_i, b_i, l_i, r_i$ で与えられる ($1 \leq u_i, d_i, f_i, b_i, l_i, r_i \leq 6N$)。ただし、入力に現れる $6N$ 個の色はすべて相異なる。入力と立方体の各面の対応は Figure D-2 を参照せよ。

入力の終わりは、1 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、美しいタワーの高さの最大値を 1 行に出力せよ。

Sample Input 1	Sample Output 1
3	3
6 12 1 10 15 7	2
5 14 2 9 17 16	1
8 13 18 4 11 3	
3	
4 16 17 10 13 15	
8 11 14 12 2 3	
18 9 7 5 6 1	
2	
5 11 8 2 4 3	
1 10 12 9 6 7	
0	

Sample Input 2

```
2
1 4 12 11 7 5
10 8 2 3 6 9
3
2 18 13 11 6 7
17 16 12 5 1 8
14 10 9 3 15 4
3
2 6 8 4 9 5
1 17 13 7 12 10
16 14 18 15 3 11
6
30 35 12 4 32 24
9 25 6 17 22 15
26 29 11 28 23 36
8 2 1 13 3 27
31 20 33 19 34 7
18 21 5 14 16 10
0
```

Sample Output 2

```
2
2
1
1
```

Problem E

秘密の数と最大公約数

Time Limit: 2 seconds

この問題はインタラクティブな問題である。2 つの秘密の整数 a と b があり、いずれも 1 以上 10^9 以下である。

あなたは、次の形式の質問を、100 回まで行うことができる。

- 0 以上 10^{18} 以下の 2 つの整数 x と y を出力する。その後、 $a+x$ と $b+y$ の最大公約数を得る。

整数 a と b を特定せよ。なお、 a と b の値はプログラムの実行前に決められていて、質問に応じてこれらの値が変化することはない。

Interaction

質問は以下の形式で標準出力に出力せよ。

? x y

ここで、 x と y は、0 以上 10^{18} 以下の整数である。この質問に対して、 $a+x$ と $b+y$ の最大公約数が、標準入力から与えられる。

a と b が特定できた場合、以下の形式で解答を標準出力に出力せよ。

! a b

質問は 100 回まで、解答は 1 回だけ行うことができる。解答を出力後、プログラムは直ちに終了しなければならない。

形式の違反や質問・解答回数の超過、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
3	? 2 5
11	? 1 17
1	? 3 0
	! 10 16

Problem F

ICPC is a Contest

Time Limit: 2 seconds

来る ICPC に向けて日々練習に励むあなたは、対策の一環として、ICPC という文字列の性質を考察しておくことにした。

手始めに、良い文字列という概念を定義しよう。ある文字列について、「連続な部分文字列として出現する ICPC を一箇所選び、その部分を文字 C に置き換える」という操作を考える。この操作を 0 回以上繰り返すことで ICPC に変化させられる文字列を、良い文字列と呼ぶことにする。例えば、ICPC, IICPCPC, ICPCICPC など良い文字列であるが、C, ICICPPC, JAG などは良い文字列ではない。

長さ n の文字列 s が与えられる。 s に（連続とは限らない）部分列として含まれる良い文字列は、いくつあるだろうか？すなわち、以下の条件を満たす整数集合 P の個数を求めてほしい。

- P の各要素は 1 以上 n 以下の整数である。また、 $P = \{p_1, p_2, \dots, p_m\}$ とするとき、 s の p_1 文字目、 p_2 文字目、..., p_m 文字目を消去したときに残る文字列は、良い文字列である。

答えは大きくなることがあるので、998 244 353 で割った余りを出力せよ。

Input

入力は複数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

テストケースは 2 行からなる。1 行目には 1 以上 7000 以下の整数 n が与えられる。2 行目には英大文字からなる長さ 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。全てのテストケースの n の合計は 7000 を超えない。

Output

各テストケースについて、問題文中の条件を満たす整数集合 P の個数を 998 244 353 で割った余りを出力せよ。

Sample Input

Sample Output

4	1
ICPC	40
12	0
ICPCICPCICPC	46
3	
JAG	
20	
ICWCPPCIWPCICWCICCCP	
0	

Problem G

XOR 旅行計画

Time Limit: 3 seconds

あなたは XOR と旅行が大好きである。これからあなたは XOR を用いて JAG 王国での旅行計画を立てたい。

JAG 王国にはいくつかの町が存在し、町から町へは一方通行の道で結ばれていて、道ごとに通行料が設定されている。JAG 王国では奇妙なことに、行先が出発地と同じであるような道も存在するようだ。ただし、出発地と行先の組が同じであるような相異なる 2 本の道は存在しない。

あなたは道をちょうど L 回通って移動するような旅行計画を立てたい。旅行計画では旅行を始める町、終わる町はどこでもよいし、同じ町や同じ道を複数回通ってもよい。各旅行計画のスコアを、通った道の通行料の bitwise XOR として定める。すべての旅行計画のうち、スコアが K 番目に大きいような旅行計画のスコアの値を求めて欲しい。

Input

入力は 10 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 L K
a1 b1 w1
a2 b2 w2
⋮
aM bM wM
```

1 行目には JAG 王国の町の数 N ($1 \leq N \leq 10$)、道の数 M ($0 \leq M \leq N^2$)、移動を行う回数 L ($1 \leq L \leq 8$)、何番目に大きいものを選ぶかを示す整数 K ($1 \leq K \leq 10^9$) が与えられる。

続く M 行には道の情報が与えられる。 i 番目の道の情報は、出発地、行先、通行料の順に、 a_i, b_i, w_i で与えられる ($1 \leq a_i, b_i \leq N, 0 \leq w_i \leq 10^{18}$)。任意の $1 \leq i, j \leq M$ について、 $i \neq j$ ならば $(a_i, b_i) \neq (a_j, b_j)$ であることが保証される。

入力の終わりは 4 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、 K 番目に大きいものが存在する場合そのスコアを、そのようなものが存在しない場合は -1 を 1 行に出力せよ。

Sample Input	Sample Output
2 4 3 2 1 1 5 1 2 4 2 1 3 2 2 7 3 3 2 4 1 2 3 2 3 4 3 1 5 6 5 1 3 1 2 1 1 3 2 1 4 2 1 5 2 1 6 2 0 0 0 0	6 -1 2

Problem H

石取りゲーム

Time Limit: 2 seconds

n 頂点からなる無向木 T があり、頂点には 1 から n までの番号が付けられている。また、整数列 $(s_1, \dots, s_n)$ がある。

q 個のクエリを順に処理せよ。このうち、 j 番目のクエリは、変更クエリまたはゲームクエリのいずれかである。

変更クエリでは、整数 p_j および r_j が与えられる。これは、 s_{p_j} の値を r_j に変更することを表す。

ゲームクエリでは、0 または 1 の整数 k_j が与えられる。このクエリにおいて、先手と後手、2 人のプレイヤーが、以下のルールで石取りゲームを行う。両者が勝利を目指して最適に行動するとき、どちらのプレイヤーが勝利するか判定せよ。

1. はじめに、 T 上の各頂点 u ($1 \leq u \leq n$) に、 s_u 個の石が置かれた状態にする。
2. その後、先手のプレイヤーから始めて交互に手番を繰り返す。それぞれのプレイヤーは、自身の手番において、以下の 2 つの条件をともに満たす頂点 v が存在するならば、そのような頂点 v を 1 個選び、その頂点から石を 1 個取り除く。ただし、そのような頂点が存在しないならばそのプレイヤーは敗北となり、手番でないプレイヤーの勝利となって石取りゲームは終了する。
 - v には 1 個以上の石が置かれている。
 - v に隣接する頂点 (v 自身を含まない) に置かれている石の個数の合計を 2 で割った余りは、 k_j に等しい。

Input

入力は複数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 b_1
⋮
a_{n-1} b_{n-1}
s_1 ⋯ s_n
q
query_1
⋮
query_q
```

1 行目には 1 以上 3×10^5 以下の整数 n が与えられる。

続く $n-1$ 行には、無向木 T の隣接情報が与えられる。このうち i 行目は 2 つの整数 a_i, b_i ($1 \leq a_i, b_i \leq n$) からなり、 T の頂点 a_i と b_i が隣接していることを表す。与えられるグラフは木であることが保証される。

次の 1 行には、0 以上 10^9 以下の n 個の整数 $s_1, \dots, s_n$ が与えられる。

次の 1 行には、1 以上 3×10^5 以下の整数 q が与えられ、これはクエリの個数を表す。続く q 行にはクエリの情報が与えられる。このうち j 番目のクエリは、次の 2 つのいずれかの形式からなる。

変更クエリは以下の形式で与えられる。ここで、 p_j は 1 以上 n 以下の整数、 r_j は 0 以上 10^9 以下の整数である。

1 p_j r_j

ゲームクエリは以下の形式で与えられる。ここで、 k_j は 0 または 1 である。

2 k_j

各テストケースにおいて、ゲームクエリが 1 つ以上含まれることが保証される。

入力の終わりは、1 個のゼロだけからなる行で表される。全てのテストケースの n の合計は 3×10^5 を超えない。また、全てのテストケースの q の合計は 3×10^5 を超えない。

Output

各ゲームクエリについて、先手が勝利するならば First と、後手が勝利するならば Second と、1 行に出力せよ。

Sample Input	Sample Output
3	First
3 1	First
3 2	Second
6 1 3	Second
4	
1 1 9	
2 0	
1 2 6	
2 1	
1	
8	
3	
2 0	
1 1 3	
2 1	
0	

Problem I

悪のロボット軍団を作ろう

Time Limit: 2 seconds

あなたは世界征服を企んでいる。具体的には『週刊・悪のロボットを作ろう』を定期購読することにより、悪のロボットのパーツを集め、それらを組み立てることにより世界制服を企んでいる。

ただし、この計画にはいくつか問題がある。

まずは『週刊・悪のロボットを作ろう』の送ってくるパーツの不親切さだ。悪のロボットはパーツ 1, パーツ 2, ..., パーツ N の N 個のパーツをひとつずつ使うことで完成する。『週刊・悪のロボットを作ろう』は M 週間にわたって毎週パーツを送ってくるが、 i 週目 ($1 \leq i \leq M$) にはパーツ L_i , パーツ $L_i + 1$, ..., パーツ R_i をひとつずつ送ってくるのだ。

もう一つの問題は、パーツの収納場所（アジト）の耐荷重である。組み立て終わった悪のロボットは野に放てばよいから問題ないが、パーツの状態だとそうはいかない。パーツ i ($1 \leq i \leq N$) の重さは w_i であるが、アジトの床が抜けないために、収納しているパーツの総重量は K を超えないようにする必要がある。ただし、これ以上アジトに入らないからとパーツを受け取り拒否してしまうと、『週刊・悪のロボットを作ろう』を発行している組織からどんな仕打ちを受けるかわからない。よって、パーツが送られてくる前日の夜までに、パーツを受け取れるように秘密裏にパーツを廃棄しなければならない。

以上をまとめると以下ようになる。

- 最初、パーツをひとつも持っていない状態から開始する。
- $i = 1, 2, \dots, M$ の順に、以下を上から順に行う。
 - 持っているパーツを 0 個以上廃棄する。
 - パーツ L_i , パーツ $L_i + 1$, ..., パーツ R_i をひとつずつ入手する。
(このとき、今回入手したパーツも含め、持っている（まだ組み立てていない）パーツの重量の総和が K 以下でなければならない.)
 - 以下の操作を 0 回以上行う。
 - * パーツ 1, パーツ 2, ..., パーツ N の N 個のパーツをひとつずつ使うことで、悪のロボットを 1 体組み立てる。

持っているパーツの重量の総和が K を超えないようにしながら適切に廃棄するパーツを選ぶことで、最大で何体の悪のロボットを組み立てることができるかを求めよ。なお、本問題の制約下では、持っているパーツの重量の総和が K を超えないようなパーツの廃棄方法が存在することが証明できる。

Input

入力は複数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 K
w1 w2 ... wN
L1 R1
⋮
LM RM
```

1 行目にはパーツの種類数を表す整数 N ($1 \leq N \leq 2 \times 10^5$), パーツの送られてくる回数を表す整数 M ($1 \leq M \leq 2 \times 10^5$), アジトの耐荷重を示す整数 K ($K \leq 4 \times 10^{18}$) がこの順に空白区切りで与えられる。
2 行目には N 個の整数 $w_1, w_2, \dots, w_N$ ($1 \leq w_i \leq 10^8$) が与えられ、パーツ i の重さが w_i であることを

表す. また, $w_1 + w_2 + \dots + w_N \leq K$ が成り立つことが保証される.

続く M 行には送られてくるパーツの情報が与えられる. このうち i 行目は 2 つの整数 L_i, R_i ($1 \leq L_i \leq R_i \leq N$) が与えられ, i 週目に送られてくるパーツがパーツ L_i , パーツ $L_i + 1$, $\dots$, パーツ R_i であることを表す.

入力の終わりは, 3 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. また, 全てのテストケースの M の合計は 2×10^5 を超えない.

Output

各テストケースについて, 組み立てることのできる悪のロボットの数の最大値を 1 行に出力せよ.

Sample Input	Sample Output
6 6 30 3 1 4 1 5 9 6 6 3 5 5 6 1 3 1 4 3 4 6 6 30 3 1 4 1 5 9 3 5 5 6 1 3 1 4 3 4 6 6 2 4 9 1 8 1 1 1 1 2 2 2 2 8 9 183 20 17 20 15 19 17 19 19 5 6 1 4 6 8 7 7 5 8 1 2 1 5 3 8 3 4 0 0 0	1 2 1 3