标题：A、Connecting segments | 时间限制：3 秒 | 内存限制：256M

Niuniu has N directed segments. Each segment has one color. He wants to make a powerful sword by connecting the segments. He can choose at most K segments. He isn't allowed to choose more than one segment from each color, or rotate segments. The powerfulness of the sword is calculated by the square of the distance between the beginning of the first segment and the end of the last segment. Niuniu wants to know the answer when K = 0,1,…,n-1.

输入描述:
The input has the format as described below.

N
x_1 y_1 c_1
x_2 y_2 c_2
...
x_n y_n c_n

N is the number of segments. ($1 \leqslant N \leqslant 1000$) The ith segment starts at (0,0), ends at(x_i,y_i), with color c_i. ($-1000000 \leqslant x_i,y_i \leqslant 1000000$, $x_i^2+y_i^2>0$, $1 \leqslant c_i \leqslant n$)
输出描述:
You should output the answers in one line, separated by a space.

备注:
You can consider the segment in this problem as 2D vector.
示例 1
输入
4
1 2 1
2 1 1
2 3 2
3 1 2
输出
13 34 34 34
示例 2
输入
2
1 1 1
-1 -1 2
输出
2 2
示例 3
输入
10
1 4 1

-2 3 2
3 2 1
2 5 3
-4 2 3
-3 3 1
3 1 2
-10 -10 4
7 7 4
7 7 5
输出
200 392 617 818 976 976 976 976 976 976

标题：B、Filling pools | 时间限制：3 秒 | 内存限制：256M

Niuniu is interested in a game. The goal of the game is to fill a pool with water. The pool is a n*n square. Every cell of the pool is either empty or filled with water. All the cells are empty at the beginning. NiuNiu has to choose exactly one cell in each row to fill with water. Every moment, for every cell, if there're at least 2 adjacent cells filled with water, the cell will be filled with water too. Note that two cells are adjacent if and only if they share a side. Niuniu wants to calculate the number of ways to fill the pool. The answer may be large, so you only need to calculate it modulo 998244353.

输入描述:

There’s only one number n in the only row. (1 ≤ n < 262144)

输出描述:

You should print exactly one number, which is the answer modulo 998244353.

示例 1

输入

3

输出

6

示例 2

输入

4

输出

22

示例 3

输入

50

输出

780401176

标题：C、Counting paths | 时间限制：1 秒 | 内存限制：256M

Niuniu is interested in counting paths. He has a tree with n vertices initially painted white. For a vertex set S, Niuniu wants to calculate f(S). f(S) is calculated as below. First the vertices in set S are painted black. If there is any white vertex which lies on the path between any two black vertices, f(S)=0. Otherwise, he will choose some set of paths. (a path from x to y is same as y to x, a path from x to x is allowed). The paths in the set cannot contain any black vertices. Next he will paint the vertices in the paths of the set red. f(S) is the number of different path set which makes all adjacent vertices of black vertices black or red. You need to calculate the sum of f(S) for every possible vertex set S. Of course S should contain at least one element. The answer may be large, you only need to calculate it modulo 998244353.

输入描述:
The input has the format as described below。

n
x_1 y_1
x_2 y_2
...
x_{n-1} y_{n-1}
n is the number of vertices. (1<=n<=200000) There’s an edge between x_i and y_i. (1 $\leqslant$ x_i, y_i $\leqslant$ n). It is guaranteed the graph is a tree.

输出描述:
You should print exactly one number, which is the
answer modulo 998244353.
示例 1
输入
2
1 2
输出
3
说明
f({1})=1

f({2})=1

f({1,2})=1
示例 2
输入
3
1 2
2 3
输出

16
说明
f({1})=f({3})=6

f({2})=2

f({1,2})=f({2,3})=f({1,2,3})=1
示例 3
输入
5
1 2
2 3
2 4
1 5
输出
3128

标题：D、Compressing data | 时间限制：3 秒 | 内存限制：512M

Niuniu has a string S of length n which only contains lowercase letters. The string will be transmitted to his best friend, Doge, in a compressed form. Niuniu would like to compress the string so that the cost is minimal. A compressed form of a string is a sequence of segments. Each segment is comprised of a number X(X ≥ 1) and a string S, which represents a new string S' =SS⋯S(repeated X times). Connecting the represented string of the segments in the sequence from left to right will result in the original string. Each segment costs A*|S|+B. The total cost of the compressed form is the sum of the costs of the segments in the sequence. You are asked to help Niuniu calculate the minimal cost of the compressed form.

输入描述:

The first line contains one integer T(T ≤ 10), which means the number of test cases.

Each test case has two lines. The first line contains two integers A and B separated by a space. (0 ≤ A,B ≤ 1000000000) The second line contains the string S, which is the original string you have to compress.

(1 ≤ |S| ≤ 100000, sum of |S| ≤ 550000)

输出描述:

For each test case，print one number in a single line, which is the minimal cost.

示例 1

输入

```
5
1 1
tooooooooeasy
100 1
yooooooooooooooooooooo
1 100
yooooooooooooooooooooo
2 1
fizzydavid
2 3
ababccdefdefdef
```

输出

```
9
202
123
21
21
```

说明

The compressed form with the minimal cost is [1,t][9,o][1,easy]

The compressed form with the minimal cost is [1,y][22,o]

The compressed form with the minimal cost is [1,yooooooooooooooooooooo]

The compressed form with the minimal cost is [1,fizzydavid] or [1,fi][2,z][1,ydavid]

The compressed form with the minimal cost is [2,ab][2,c][3,def]

标题：E、Touring cities | 时间限制：2 秒 | 内存限制：256M

Niuniu wants to tour the cities in Moe country. Moe country has a total of n*m cities. The positions of the cities form a grid with n rows and m columns. We represent the city in row x and column y as (x,y). (1 ≤ x ≤ n,1 ≤ y ≤ m) There are bidirectional railways between adjacent cities. (x1,y1) and (x2,y2) are called adjacent if and only if |x1-x2|+|y1-y2|=1. There are also K bidirectional air lines between K pairs of cities. It takes Niuniu exactly one day to travel by a single line of railway or airplane. Niuniu starts and ends his tour in (1,1). What is the minimal time Niuniu has to travel between cities so that he can visit every city at least once?

输入描述:

The first line contains oneinteger T(T≤20), which means the number of test cases.

Each test case has the format as described below.

n m K

ax_1 ay_1 bx_1 by_1

ax_2 ay_2 bx_2 by_2

…

ax_K ay_K bx_K by_K

(2 ≤ K ≤ 10. 1 ≤ n*m ≤ 100)

There is one bidirectional air line between (ax_i,ay_i) and (bx_i,by_i). (1 ≤ ax_i,bx_i ≤ n , 1 ≤ ay_i,by_i ≤ m)

输出描述:

For each test case， print one number in a single line, which is the minimal number of days Niuniu has to travel between cities so that he can visit every city at least once.

备注:

The air line may start and end in the same city.

示例 1

输入

3

2 2 1

1 1 2 2

3 3 1

1 1 3 3

3 3 0

输出

4

9

10

标题：F、Protecting lawn | 时间限制：4 秒 | 内存限制：512M

Niuniu is interested in a game called Plants vs Zombies. Niuniu likes wall-nuts and spikeweeds very much. The wall-nuts have hard shells which you can use to protect other plants. Spikeweeds hurt any zombies that step on them. The game with only wall-nuts and spikeweeds work as follow:

We can represent the lawn as a straight line of length L. Your house is located at position 0. The zombies appear at position L. The lawn has L grids. The length of a grid is 1. (You can represent the i-th grid with a segment [i-1,i]) You can place plants in the grids and one grid cannot contain more than one plant. The zombies walk from right to left with a speed of 1 per second. Every zombie has HP. When HP decreases to 0, the zombie will die and disappear from the lawn. The wall-nuts can stop the zombies. The zombies will stop to eat the wall-nut when they step into the grid of the wall-nut. (the zombie will stop at position i if the wall-nut is in the i-th grid) Every wall-nut has a DUR. The wall-nuts will disappear after being eaten by at least one zombie for exactly DUR seconds. **(It is not affected by the number of zombies)** The spikeweed will damage the zombies in its grid with continuous damage of 1HP per second. Note that if the i-th grid is a wall-nut and the (i+1)-th grid is a spikeweed, then the zombies will be damaged by the spikeweed while they are eating the wall-nut.

Niuniu wants to do four types of queries:
1) At moment t_i, place a wall-nut with DURi in the p_i-th grid. The old wall-nut will be placed by the new wall-nut if exists. It is guaranteed that the grid hasn't been placed any plants before.

2) At moment t_i, place a spikeweed in the pi-th grid. It is guaranteed that the grid hasn't been placed any plants before. (The grid isn't allowed to place a plant even though the wall-nut in the grid disappears)

3) At moment t_i, a zombie with HP_i appears at position L.

4) At moment t_i, query the position of the zombie which appeared in the x_i-th query. If the zombie has reached your house with HP strictly higher than zero, then you should print "0". If the zombie is dead(or it dies exactly at moment t_i), then you should print the position where it died. Otherwise you should print the position of it.

Can you answer the queries?

输入描述:

The first line contains twointegers Q and L(1 $\leqslant$ Q,L $\leqslant$ 400000), which means the number of queries andthe length of the lawn.

Each of the following Q lines describes a query:

Query 1: W t_i p_i DUR_i
Query 2: S t_i p_i

Query 3: Z t_i HP_i

Query 4: Q t_i x_i

See the description of the query in the problem description.

$(1 \le t_i \le t_{i+1} \le 1e8 \quad 1 \le p_i \le L \quad 1 \le DUR_i, HP_i \le 1e8 \quad \sum DUR_i \le 1e8)$

输出描述:

For each query of type 4, print one number in a single line, which is the position of the zombie.

备注:

here're |x-y| seconds between moment x and moment y.

The zombies won't be stopped at position i if there is a wall-nut in the (i+1)-th grid. (This happens when you place a wall-nut behind the zombie)

示例 1

输入

12 5
S 1 5
Z 1 1
Z 1 5
Z 2 3
Q 3 2
W 3 4 1
W 4 1 4
S 6 2
Q 6 4
Q 6 3
Q 10 4
Q 10 3

输出

4
2
1
1
0

标题：G、Counting regions | 时间限制：1 秒 | 内存限制：128M

Niuniu likes mathematics. He also likes drawing pictures. One day, he was trying to draw a regular polygon with n vertices. He connected every pair of the vertices by a straight line as well. He counted the number of regions inside the polygon after he completed his picture. He was wondering how to calculate the number of regions without the picture. Can you calculate the number of regions modulo 1000000007? **It is guaranteed that n is odd.**

输入描述:

The only line contains one odd number n(3 ≤ n ≤ 1000000000), which is the number of vertices.

输出描述:

Print a single line with one number, which is the answer modulo 1000000007.

备注:

The following picture shows the picture which is drawn by Niuniu when n=5. Note that no more than three diagonals share a point when n is odd.

示例 1

输入

3

输出

1

示例 2

输入

5

输出

11

标题：H、Playing games | 时间限制：1 秒 | 内存限制：256M

Niuniu likes playing games. He has n piles of stones. The i-th pile has a_i stones. He wants to play with his good friend, UinUin. Niuniu can choose some piles out of the n piles. They will play with the chosen piles of stones. UinUin takes the first move. They take turns removing at least one stone from one chosen pile. The player who removes the last stone from the chosen piles wins the game. Niuniu wants to choose the maximum number of piles so that he can make sure he wins the game. Can you help Niuniu choose the piles?

输入描述:

The first line contains one integer n (1 ≤ n ≤ 500000), which means the number of piles.

The second line describes the piles, containing n non-negative integers, a_1 a_2 … a_n, separated by a space. The integers are less than or equal to 500000.

输出描述:

Print a single line with one number, which is the maximum number of piles Niuniu can choose to make sure he wins. If Niuniu cannot always win whatever piles he chooses, print 0.

示例 1

输入

8

1 9 2 6 0 8 1 7

输出

7

标题：I、Permuting cows | 时间限制：3 秒 | 内存限制：512M

Niuniu likes cows. He has n cows in his farm. He loves the cows very much, but the cows seem unhappy. Niuniu figures out why this happens. The cows are arranged in a sequence. Two cows may become unhappy when they are adjacent in the sequence.Every cow has a characteristic value, named v_i. The unhappiness between cow i and i+1 is defined as v_i xor v_{i+1}. (xor is exclusive OR) Niuniu wants the largest unhappiness as low as possible. You need to tell him a permutation p_i, so that when the cow i goes to the position p_i, the largest unhappiness of the new sequence is as low as possible. Among all the permutations, you need to tell him the **lexicographically smallest** one.

输入描述:

The first line contains one integer n, which is the number of cows. The second line contains n integers, v_1 v_2 … v_n, separated by a space. ($2 \leqslant n \leqslant 300000$，$0 \leqslant v_i \leqslant 1000000000$)

输出描述:

Print a single line with n numbers, which means the permutation.

备注:

This problem contains a large number of testcases. Please, check your code carefully before you submit and wait patiently for the result.

示例 1

输入

8

1 9 2 6 0 8 1 7

输出

1 2 6 5 3 4 7 8

标题：J、Calculating sums | 时间限制：3 秒 | 内存限制：256M

Niuniu likes calculating sums. He has recently learnt how to calculate sums using various methods. Here is one of them:

$$\sum_{i=0}^{N}\sum_{j=0}^{M}\binom{i}{j}\cdot[i\ \equiv\ 0(mod\ 2)]\cdot[j\ \equiv\ 0(mod\ 2)]$$

Note that [x] is 1 when x is true and 0 when x is false.

Can you calculate the sum? The answer may be large, so please calculate the sum modulo a given number K.

输入描述:

The only line contains three integers N, M, K.

$1 \leqslant N \leqslant 10^9$, $1 \leqslant M \leqslant 10^6$, $1 \leqslant K \leqslant 10^9$

输出描述:

Print a single line with one number, which is the answer.

示例 1

输入

2 3 3

输出

0

标题：K、Decoding graphs | 时间限制：2 秒 | 内存限制：256M

Niuniu likes cryptology. His friend, Gougou, sent him a graph as his birthday present. The graph is encoded, so Niuniu must answer a question to decode it. The graph has n vertices and m edges. Every vertex has a non-negative integer a_i. Here follows the question:

You need to assign each vertex another non-negative integer b_i, satisfying:

1. For an edge (u, v)，$b_u \neq b_v$.
2. For a vertex u，$0 \leqslant b_u \leqslant a_u$.
3. b_1 xor b_2 xor ... xor b_n = C. (xor means exclusive OR)

What is the number of valid assignments modulo 998244353?

Can you help Niuniu answer the question?

输入描述:

The first line contains three integers n, m, C，which are the number of vertices, number of edges and the given number C.

The second line contains n non-negative integers, a_1, a_2, a_3, ..., a_n, separated by a space.

The following m lines each line contains two different numbers, u and v, meaning that there is an edge between u and v. There are no multiple edges in the graph.

$1 \leqslant n \leqslant 13, 0 \leqslant m \leqslant n(n-1)/2, 0 \leqslant a_i, C \leqslant 10^{18}$

输出描述:

Print a single line with one number, which means the answer.

示例 1

输入

3 1 2

1 2 3

1 2

输出

4