# 目录

**目录**

# 1 A. Alkane

3s, 512MB

## 1.1 Problem Description

In organic chemistry, an alkane is an acyclic saturated hydrocarbon. In other words, an alkane consists of hydrogen and carbon atoms arranged in a tree structure in which all the carbon-carbon bonds are single. Alkanes have the general chemical formula $C_nH_{2n+2}$.

In an alkane, each carbon atom has 4 bonds (either C-C or C-H), and each hydrogen atom is joined to one of the carbon atoms (so in a C-H bond). The number of carbon atoms may be thought of as the size of the alkane.

Alkanes with more than 3 carbon atoms can be arranged in various different ways, forming structural isomers. The simplest isomer of an alkane is the one in which the carbon atoms are arranged in a single chain with no branches. This isomer is sometimes called the n-isomer. However the chain of carbon atoms may also be branched at one or more points. The number of possible isomers increases rapidly with the number of carbon atoms.

Moreover, we define an alkyl is an alkane missing one hydrogen. An acyclic alkyl has the general formula $C_nH_{2n+1}$. The defination of structural isomer of alkyls is similar to alkanes'.

Now you need to count the number of structural isomers of alkanes and alkyls of size $n$ modulo 998244353.

## 1.2 Input

The first line of the input contains an integer $T$, denoting the number of test cases.

In each test case, there is a single integer $n$ in one line, denoting the size of alkanes and alkyls.

$1 \le T \le 100000, 1 \le n \le 100000$

## 1.3 Output

For each test case, output one line contains two integers, denoting the number of structural isomers of alkanes and alkyls of size $n$ modulo 998244353.
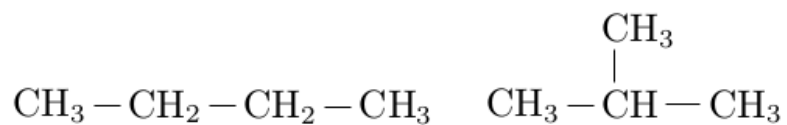
## 1.4 Sample Input
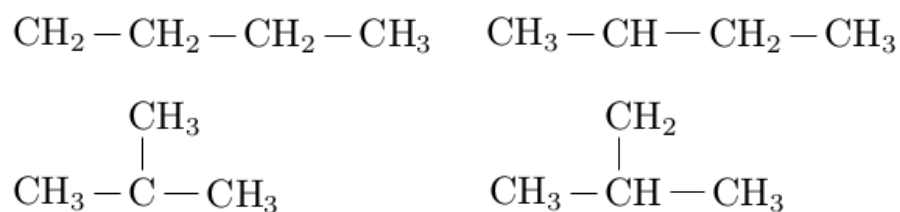
9

1
2
3
4
5
6
7
8
9

## 1.5  Sample Output

```
1 1
1 1
1 2
2 4
3 8
5 17
9 39
18 89
35 211
```

## 1.6  Sample Explanation

For alkanes of size 4 ($C_4H_{10}$), there are 2 structural isomers:

$$CH_3 - CH_2 - CH_2 - CH_3 \qquad CH_3 - \overset{\overset{\textstyle CH_3}{|}}{CH} - CH_3$$

For alkyls of size 4 ($C_4H_9$), there are 4 structural isomers:

$$CH_2 - CH_2 - CH_2 - CH_3 \qquad CH_3 - CH - CH_2 - CH_3$$

$$CH_3 - \overset{\overset{\textstyle CH_3}{|}}{C} - CH_3 \qquad CH_3 - \overset{\overset{\textstyle CH_2}{|}}{CH} - CH_3$$

# 2 B. Beads

4s, 512MB

## 2.1 Problem Description

There are $m$ different colors of beads. Define a necklace of length $n$ is a cyclic string that successively connects $n$ beads, **taking all rotations and overturnings as equivalent.**

For example, $[1, 2, 3, 4]$ is equivalent to $[2, 3, 4, 1]$, $[3, 4, 1, 2]$ and $[4, 1, 2, 3]$ (by rotation); and $[1, 2, 3, 4]$ is equivalent to $[1, 4, 3, 2]$, $[3, 2, 1, 4]$, $[2, 1, 4, 3]$ and $[4, 3, 2, 1]$ (by overturning).

Moreover, each time you could transpose the beads of color $i$ to color $(i \bmod m) + 1$ **for all $i$ at the same time**.

After some operations, if a necklace $A$ becomes $B$, then $B$ is equivalent to $A$.

Count the number of necklaces modulo 998244353.

## 2.2 Input

The first line of the input contains an integer $T$, denoting the number of test cases.

In each test case, there are two integers $n, m$ in one line, denoting the length of necklaces, and the number of colors.

$1 \le T \le 20, 3 \le n \le 10^{18}, 2 \le m \le 10^{18}, 998244353 \nmid n, m$

## 2.3 Output

For each test case, output one line contains a single integer, denoting the number of necklaces modulo 998244353.

## 2.4 Sample Input

```
5
3 2
4 2
8 5
9 5
2333 333
```

## 2.5   Sample Output

```
2
4
5079
22017
544780894
```

## 2.6   Sample Explanation

For $n = 3, m = 2$:

- $[1, 1, 1], [2, 2, 2]$ are equivalent

- $[1, 1, 2], [1, 2, 1], [2, 1, 1], [2, 2, 1], [2, 1, 2], [1, 2, 2]$ are equivalent

# 3 C. Calculate

3s, 512MB

## 3.1 Problem Description

Given $A, B, C$, Calculate

$$\sum_{i=1}^{A}\sum_{j=1}^{B}\sum_{k=1}^{C}\varphi(\gcd(i, j^2, k^3)) \pmod{2^{30}}$$

Where $\varphi(n)$ denotes the number of positive integers $\leq n$ that are relatively prime to $n$.

## 3.2 Input

The first line of the input contains an integer $T$, denoting the number of test cases. In each test case, there are three integers $A, B, C$ in one line, as described above. $1 \leq T \leq 10, 0 < A, B, C \leq 10^7$

## 3.3 Output

For each test case, output one line contains a single integer, denoting the answer modulo $2^{30}$.

## 3.4 Sample Input

```
4
96 93 95
970 906 893
92460 95043 54245
9760979 8053227 7156842
```

## 3.5 Sample Output

```
1114536
28070648
388873924
623507672
```

# 4 D. Permutation

2s, 512MB

## 4.1 Problem Description

Define the value of a permutation of $1 \ldots n$ as

$$\sum_{i=1}^{n} |p_i - i|$$

Given $n, p$, for every $s = 0 \ldots n^2 - 1$, count the number of permutations of length $n$ whose value equals to $s$.

Output the answer modulo a prime $p$.

## 4.2 Input

The first line of the input contains an integer $T$, denoting the number of test cases.

In each test case, there are two integers $n, p$ in one line, denoting the length of permutations and the modulo.

$1 \le T \le 10, 1 \le n \le 100, 10^9 < p < 1.1 \times 10^9$, only two cases' $n > 50$, $p$ is a prime.

## 4.3 Output

For each test case, output one line contains $n^2$ integer, denoting the number of permutations of length $n$ whose value equals to $s = 0 \ldots n^2 - 1$ modulo $p$.

## 4.4 Sample Input

```
4
4 1000000223
3 1000000321
2 1000000123
1 1000000447
```

## 4.5 Sample Output

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

# 5 E. TeaTree

3s, 1GB

## 5.1 Problem Description

Recently, TeaTree acquire new knoledge gcd (Greatest Common Divisor), now she want to test you.

As we know, TeaTree is a tree and her root is node 1, she have n nodes and n-1 edge, for each node i, it has it's value v[i].

For every two nodes i and j (**i is not equal to j**), they will tell their Lowest Common Ancestors (LCA) a number : gcd(v[i],v[j]).

For each node, you have to calculate the max number that it heard.

some definition:

In graph theory and computer science, the lowest common ancestor (LCA) of two nodes u and v in a tree is the lowest (deepest) node that has both u and v as descendants, where we define each node to be a descendant of itself.

## 5.2 Input

On the first line, there is a positive integer n, which describe the number of nodes.

Next line there are n-1 positive integers f[2] ,f[3], …, f[n], f[i] describe the father of node i on tree.

Next line there are n positive integers v[2] ,v[3], …, v[n], v[i] describe the value of node i.

n<=100000, f[i]<i, v[i]<=100000

## 5.3 Output

Your output should include n lines, for i-th line, output the max number that node i heard.

For the nodes who heard nothing, output -1.

## 5.4 Sample Input

```
4
1 1 3
4 1 6 9
```

## 5.5   Sample Output

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

# 6 F. NewNippori

3s, 512MB

## 6.1 Problem Description

Given a connected undirected graph with $n$ nodes and $m$ edges, the capacity of each edge is 1.

Define $\text{maxflow}(s, t)$ as the maxflow from $s$ to $t$.

Calculate

$$\sum_{i=1}^{n}\sum_{j=1}^{i-1}\min(\text{maxflow}(i, j), 3)$$

## 6.2 Input

The first line of the input contains an integer $T$, denoting the number of test cases.

In each test case, there are two integers $n, m$ in the first line, denoting the number of vertices and edges of the graph.

Then followed $m$ lines, each line contains two integers $u_i, v_i$, denoting an undirected edge $(u_i, v_i)$.

$1 \le T \le 20, 1 \le n \le 100000, 0 \le m \le 300000, \sum n \le 400000, \sum m \le 700000, u_i \ne v_i$

## 6.3 Output

For each test case, output one line contains a single integer, denoting the answer.

## 6.4 Sample Input

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

## 6.5 Sample Output

21

# 7 G. Cyclic

1s, 512MB

## 7.1 Problem Description

Count the number of cyclic permutations of length $n$ with no continuous subsequence $[i, i + 1 \bmod n]$.

Output the answer modulo 998244353.

## 7.2 Input

The first line of the input contains an integer $T$, denoting the number of test cases.

In each test case, there is a single integer $n$ in one line, denoting the length of cyclic permutations.

$1 \leq T \leq 20, 1 \leq n \leq 100000$

## 7.3 Output

For each test case, output one line contains a single integer, denoting the answer modulo 998244353.

## 7.4 Sample Input

```
3
4
5
6
```

## 7.5 Sample Output

```
1
8
36
```

# 8 H. Pow

1s, 512MB

## 8.1 Problem Description

There are $n$ numbers $3^0, 3^1, \ldots, 3^{n-1}$. Each time you can choose a subset of them (may be empty), and then add them up.

Count how many numbers can be represented in this way.

## 8.2 Input

The first line of the input contains an integer $T$, denoting the number of test cases.

In each test case, there is a single integers $n$ in one line, as described above.

$1 \le T \le 20, 0 \le n \le 1000$

## 8.3 Output

For each test case, output one line contains a single integer, denoting the answer.

## 8.4 Sample Input

```
4
9
7
8
233
```

## 8.5 Sample Output

```
512
128
256
13803492693581127574869511724554050904902217944340773110325048447598592
```

# 9  I. Count

2s, 512MB

## 9.1  Problem Description

Multiple query, for each $n$, you need to get

$$\sum_{i=1}^{n}\sum_{j=1}^{i-1}[\gcd(i+j, i-j) = 1]$$

## 9.2  Input

On the first line, there is a positive integer T, which describe the number of queries. Next there are T lines, each line give a positive integer n, as mentioned above. T<=1e5, n<=2e7

## 9.3  Output

Your output should include T lines, for each line, output the answer for the corresponding n.

## 9.4  Sample Input

```
4
978
438
233
666
```

## 9.5  Sample Output

```
194041
38951
11065
89963
```

# 10    J. CSGO

2s, 512MB

## 10.1    Problem Description

You are playing CSGO.

There are n Main Weapons and m Secondary Weapons in CSGO.

You can **only** choose one Main Weapon and one Secondary Weapon.

For each weapon, it has a composite score S.

The higher the composite score of the weapon is, the better for you.

Also each weapon has K performance evaluations x[1], x[2], …, x[K].(range, firing rate, recoil, weight…)

So you shold consider the cooperation of your weapons, you want two weapons that have big difference in each performance, for example, AWP + CZ75 is a good choose, and so do AK47 + Desert Eagle.

All in all, you will evaluate your weapons by this formula.(MW for Main Weapon and SW for Secondary Weapon)

$$S_{MW} + S_{SW} + \sum_{i=1}^{K} |x_{MW}[i] - x_{SW}[i]|$$

Now you have to choose your best Main Weapon & Secondary Weapon and output the maximum evaluation.

## 10.2    Input

Multiple query.

On the first line, there is a positive integer T, which describe the number of data.

Next there are T groups of data.

for each group, the first line have three positive integers n, m, K.

then, the next n line will describe n Main Weapons, K+1 integers each line S, x[1], x[2], …, x[K]

then, the next m line will describe m Secondary Weapons, K+1 integers each line S, x[1], x[2], …, x[K]

There is a blank line before each groups of data.

T<=100, n<=100000, m<=100000, K<=5, 0<=S<=1e9, |x[i]|<=1e9, sum of (n+m)<=300000

## 10.3   Output

Your output should include T lines, for each line, output the maximum evaluation for the corresponding datum.

## 10.4   Sample Input

```
2

2 2 1
0 233
0 666
0 123
0 456

2 2 1
100 0
1000 100
1000 100
100 0
```

## 10.5   Sample Output

```
543
2000
```

# 11  K. Pow2

2s, 512MB

## 11.1  Problem Description

SillyDarkGK has two type of numbers:

type1. $2^x$. (2, 4, 8, 16, …)

type2. $-(2^x)$. (-2, -4, -8, -16, …)

He wants a new number S, for getting S, he should choose some numbers that the sum of them is S.

For example, if SillyDarkGK wants 5, he can use 2, 2, -1, -1, -1, 8 and -4 to get it. (He can use number **arbitrary times**.)

(S is very big, so we will give you a 01-string to describe it.)

Choosing is boring, so SillyDarkGK wants to choose numbers as fewer as possible.

To increase the difficulty, SillyDarkGK can't use some special numbers.

(We will also give you two 01-string to describe what you can't use.)

it's guaranted that there is **at least** one way to solve it and answer won't exceed 1e9.

## 11.2  Input

Multiple query.

On the first line, there is a positive integer T, which describe the number of data.

Next there are T groups of data.

for each group, the first line have a positive integer n which describe the number of bits of S.

and on the second line there is an n-length-01-string to describe S, from low to high.

On the third line there is an n-length-01-string which describe the type1 number that you can't use, from low to high, if there is a '1' on the x-th digit, you can't use two to the power of x.

On the last line there is an n-length-01-string which describe the type2 number that you can't use, from low to high, if there is a '1' on the x-th digit, you can't use minus two to the power of x.

There is a blank line before each groups of data.

T<=1000, n<=100000, sum of n<=1000000, answer won't exceed 1e9

## 11.3 Output

Your output should include T lines, for each line, output the answer for the corresponding datum.

## 11.4 Sample Input

```
3

6
110010
110101
011111

9
100101110
011111111
111111111

5
11111
00000
00000
```

## 11.5 Sample Output

```
3
233
2
```

## 11.6 Sample Explanation

for the first datum, SillyDarkGK will use 4, 16 and -1;
for the second datum, SillyDarkGK will use 1 for 233 times;
for the third datum, SillyDarkGK will use 32 and -1.

# 12 L. Videos

2s, 512MB

## 12.1 Problem Description

C-bacteria takes charge of two kinds of videos: 'The Collection of Silly Games' and 'The Collection of Horrible Games'.

For simplicity's sake, they will be called as videoA and videoB.

There are some people who want to watch videos during today, and they will be happy after watching videos of C-bacteria.

There are n hours a day, m videos are going to be show, and the number of people is K.

Every video has a type(videoA or videoB), a running time, and the degree of happiness after someone watching whole of it.

People can watch videos continuous(If one video is running on 2pm to 3pm and another is 3pm to 5pm, people can watch both of them).

But each video **only** allows **one person** for watching.

For a single person, it's better to watch two kinds to videos alternately, or he will lose W happiness.

For example, if the order of video is 'videoA, videoB, videoA, videoB, ...' or 'B, A, B, A, B, ...', he won't lose happiness; But if the order of video is 'A, **B, B, B**, A, B, **A, A**', he will lose 3W happiness.

Now you have to help people to maximization the sum of the degree of happiness.

## 12.2 Input

Multiple query.

On the first line, there is a positive integer T, which describe the number of data.

Next there are T groups of data.

for each group, the first line have four positive integers n, m, K, W : n hours a day, m videos, K people, lose W happiness when watching same videos).

and then, the next m line will describe m videos, four positive integers each line S, T, w, op : video is the begin at S and end at T, the happiness that people can get is w, and op describe it's tpye(op=0 for videoA and op=1 for videoB).

There is a blank line before each groups of data.

T<=20, n<=200, m<=200, K<=200, W<=20, 1<=S<T<=n, W<=w<=1000, op=0 or op=1

## 12.3   Output

Your output should include T lines, for each line, output the maximum happiness for the corresponding datum.

## 12.4   Sample Input

```
2

10 3 1 10
1 5 1000 0
5 10 1000 1
3 9 10 0

10 3 1 10
1 5 1000 0
5 10 1000 0
3 9 10 0
```

## 12.5   Sample Output

```
2000
1990
```