A. Dead Pixel

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Screen resolution of Polycarp's monitor is $a \times b$ pixels. Unfortunately, there is one dead pixel at his screen. It has coordinates (x, y) ($0 \le x < a, 0 \le y < b$). You can consider columns of pixels to be numbered from 0 to $a-1$, and rows — from 0 to $b-1$.

Polycarp wants to open a rectangular window of maximal size, which doesn't contain the dead pixel. The boundaries of the window should be parallel to the sides of the screen.

Print the maximal area (in pixels) of a window that doesn't contain the dead pixel inside itself.

Input

In the first line you are given an integer t ($1 \le t \le 10^4$) — the number of test cases in the test. In the next lines you are given descriptions of t test cases.

Each test case contains a single line which consists of 4 integers a, b, x and y ($1 \le a, b \le 10^4$; $0 \le x < a$; $0 \le y < b$) — the resolution of the screen and the coordinates of a dead pixel. It is guaranteed that $a + b > 2$ (e.g. $a = b = 1$ is impossible).

Output

Print t integers — the answers for each test case. Each answer should contain an integer equal to the maximal possible area (in pixels) of a rectangular window, that doesn't contain the dead pixel.

Example

input

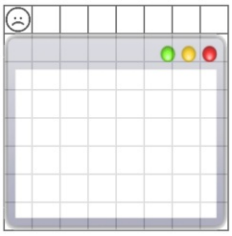
```
6
8 8 0 0
1 10 0 3
17 31 10 4
2 1 0 0
5 10 3 9
10 10 4 8
```

output

```
56
6
442
1
45
80
```

Note

In the first test case, the screen resolution is 8×8, and the upper left pixel is a dead pixel. Here you can see one of two possible layouts of the maximal window.



B. Homecoming

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

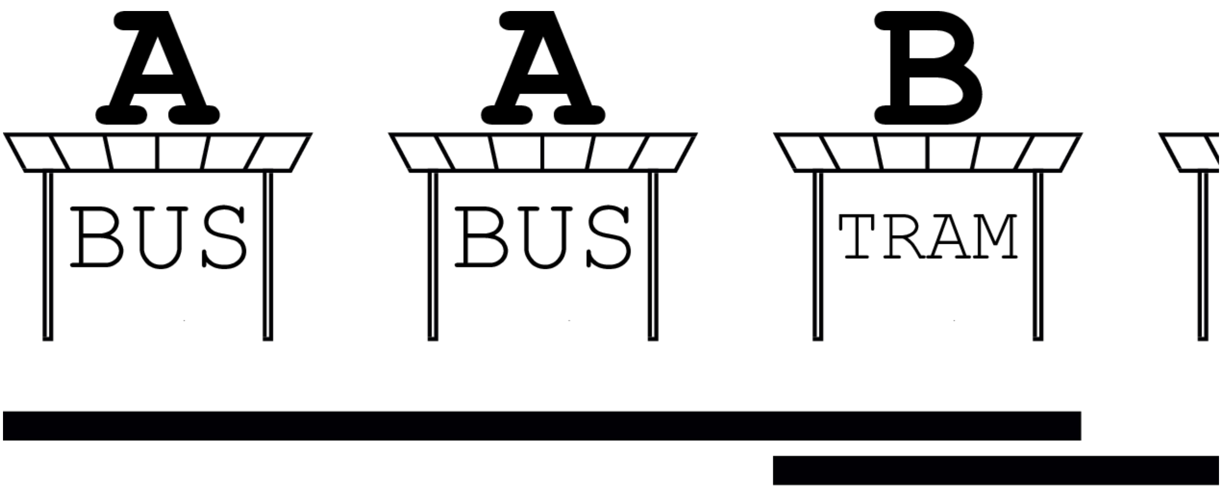
After a long party Petya decided to return home, but he turned out to be at the opposite end of the town from his home. There are n crossroads in the line in the town, and there is either the bus or the tram station at each crossroad.

The crossroads are represented as a string s of length n, where $s_i = \texttt{A}$, if there is a bus station at i-th crossroad, and $s_i = \texttt{B}$, if there is a tram station at i-th crossroad. Currently Petya is at the first crossroad (which corresponds to s_1) and his goal is to get to the last crossroad (which corresponds to s_n).

If for two crossroads i and j for all crossroads $i, i+1, \ldots, j-1$ there is a bus station, one can pay a roubles for the bus ticket, and go from i-th crossroad to the j-th crossroad by the bus (it is not necessary to have a bus station at the j-th crossroad). Formally, paying a roubles Petya can go from i to j if $s_t = \texttt{A}$ for all $i \le t < j$.

If for two crossroads i and j for all crossroads $i, i+1, \ldots, j-1$ there is a tram station, one can pay b roubles for the tram ticket, and go from i-th crossroad to the j-th crossroad by the tram (it is not necessary to have a tram station at the j-th crossroad). Formally, paying b roubles Petya can go from i to j if $s_t = \texttt{B}$ for all $i \le t < j$.

For example, if s="AABBBAB", $a = 4$ and $b = 3$ then Petya needs:



- buy one bus ticket to get from 1 to 3,
- buy one tram ticket to get from 3 to 6,
- buy one bus ticket to get from 6 to 7.

Thus, in total he needs to spend $4+3+4=11$ roubles. Please note that the type of the stop at the last crossroad (i.e. the character s_n) does not affect the final expense.

Now Petya is at the first crossroad, and he wants to get to the n-th crossroad. After the party he has left with p roubles. He's decided to go to some station on foot, and then go to home using only public transport.

Help him to choose the closest crossroad i to go on foot the first, so he has enough money to get from the i-th crossroad to the n-th, using only tram and bus tickets.

Input

Each test contains one or more test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$).

The first line of each test case consists of three integers a, b, p ($1 \le a, b, p \le 10^5$) — the cost of bus ticket, the cost of tram ticket and the amount of money Petya has.

The second line of each test case consists of one string s, where $s_i = \mathtt{A}$, if there is a bus station at i-th crossroad, and $s_i = \mathtt{B}$, if there is a tram station at i-th crossroad ($2 \le |s| \le 10^5$).

It is guaranteed, that the sum of the length of strings s by all test cases in one test doesn't exceed 10^5.

Output

For each test case print one number — the minimal index i of a crossroad Petya should go on foot. The rest of the path (i.e. from i to n he should use public transport).

Example

input

```
5
2 2 1
BB
1 1 1
AB
3 2 8
AABBBBAABB
5 3 4
BBBBB
2 1 1
ABABAB
```

output

```
2
1
3
1
6
```

C. Restoring Permutation

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a sequence $b_1, b_2, \ldots, b_n$. Find the lexicographically minimal permutation $a_1, a_2, \ldots, a_{2n}$ such that $b_i = \min(a_{2i-1}, a_{2i})$, or determine that it is impossible.

Input

Each test contains one or more test cases. The first line contains the number of test cases t ($1 \le t \le 100$).

The first line of each test case consists of one integer n — the number of elements in the sequence b ($1 \le n \le 100$).

The second line of each test case consists of n different integers $b_1, \ldots, b_n$ — elements of the sequence b ($1 \le b_i \le 2n$).

It is guaranteed that the sum of n by all test cases doesn't exceed 100.

Output

For each test case, if there is no appropriate permutation, print one number -1.

Otherwise, print $2n$ integers $a_1, \ldots, a_{2n}$ — required lexicographically minimal permutation of numbers from 1 to $2n$.

Example

input

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

output

```
1 2
-1
4 5 1 2 3 6
-1
1 3 5 6 7 9 2 4 8 10
```

D. Recommendations

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

VK news recommendation system daily selects interesting publications of one of n disjoint categories for each user. Each publication belongs to exactly one category. For each category i batch algorithm selects a_i publications.

The latest A/B test suggests that users are reading recommended publications more actively if each category has a different number of publications within daily recommendations. The targeted algorithm can find a single interesting publication of i-th category within t_i seconds.

What is the minimum total time necessary to add publications to the result of batch algorithm execution, so all categories have a different number of publications? You can't remove publications recommended by the batch algorithm.

Input

The first line of input consists of single integer n — the number of news categories ($1 \le n \le 200\,000$).

The second line of input consists of n integers a_i — the number of publications of i-th category selected by the batch algorithm ($1 \le a_i \le 10^9$).

The third line of input consists of n integers t_i — time it takes for targeted algorithm to find one new publication of category i ($1 \le t_i \le 10^5$).

Output

Print one integer — the minimal required time for the targeted algorithm to get rid of categories with the same size.

Examples

input

```
5
3 7 9 7 8
```

```
5 2 5 7 5
```

output

```
6
```

input

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

output

```
0
```

Note

In the first example, it is possible to find three publications of the second type, which will take 6 seconds.

In the second example, all news categories contain a different number of publications.

E. Double Elimination

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

The biggest event of the year – Cota 2 world championship "The Innernational" is right around the corner. 2^n teams will compete in a double-elimination format (please, carefully read problem statement even if you know what is it) to identify the champion.

Teams are numbered from 1 to 2^n and will play games one-on-one. All teams start in the upper bracket.

All upper bracket matches will be held played between teams that haven't lost any games yet. Teams are split into games by team numbers. Game winner advances in the next round of upper bracket, losers drop into the lower bracket.

Lower bracket starts with 2^{n-1} teams that lost the first upper bracket game. Each lower bracket round consists of two games. In the first game of a round 2^k teams play a game with each other (teams are split into games by team numbers). 2^{k-1} loosing teams are eliminated from the championship, 2^{k-1} winning teams are playing 2^{k-1} teams that got eliminated in this round of upper bracket (again, teams are split into games by team numbers). As a result of each round both upper and lower bracket have 2^{k-1} teams remaining. See example notes for better understanding.

Single remaining team of upper bracket plays with single remaining team of lower bracket in grand-finals to identify championship winner.

You are a fan of teams with numbers $a_1, a_2, \ldots, a_k$. You want the championship to have as many games with your favourite teams as possible. Luckily, you can affect results of every championship game the way you want. What's maximal possible number of championship games that include teams you're fan of?

Input

First input line has two integers n, k — 2^n teams are competing in the championship. You are a fan of k teams ($2 \le n \le 17; 0 \le k \le 2^n$).

Second input line has k distinct integers $a_1, \ldots, a_k$ — numbers of teams you're a fan of ($1 \le a_i \le 2^n$).

Output

Output single integer — maximal possible number of championship games that include teams you're fan of.

Examples

input

```
3 1
6
```

output

```
6
```

input

```
3 3
1 7 8
```

output

```
11
```

input

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

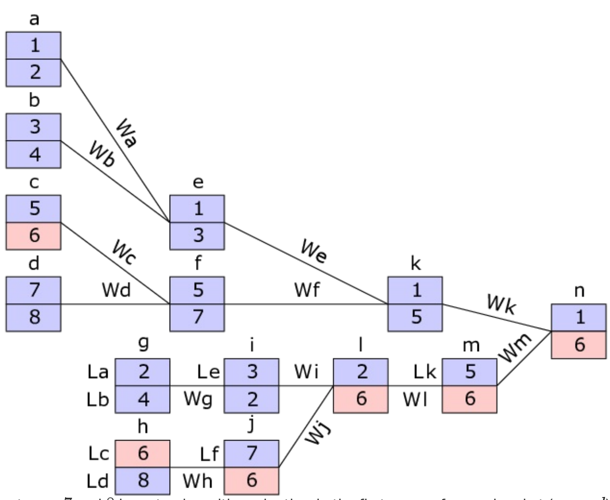
output

```
14
```

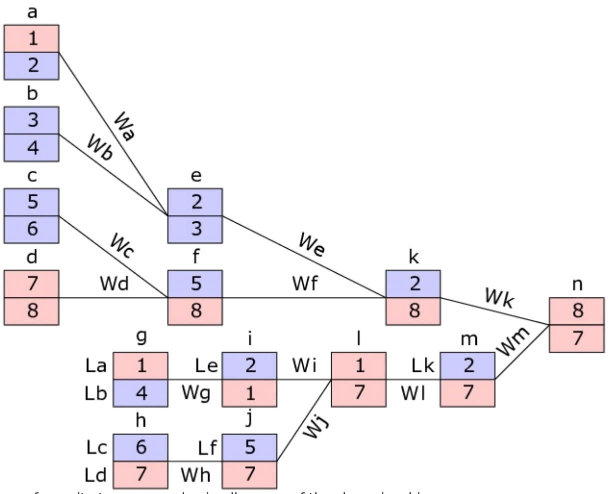
Note

On the image, each game of the championship is denoted with an English letter (a to n). Winner of game i is denoted as Wi, loser is denoted as Li. Teams you're a fan of are highlighted with red background.

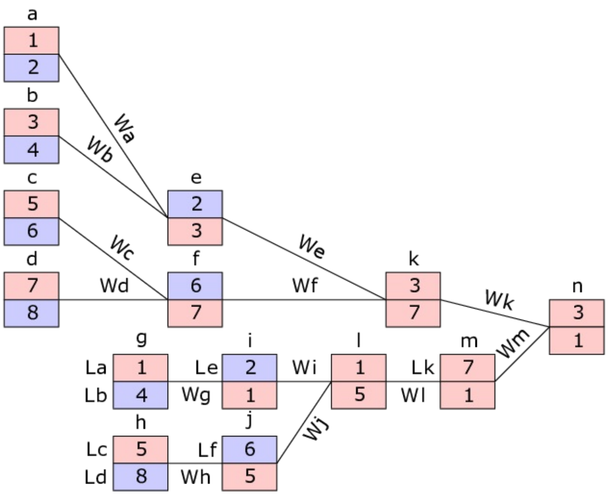
In the first example, team 6 will play in 6 games if it looses the first upper bracket game (game c) and wins all lower bracket games (games h, j, l, m).



In the second example, teams 7 and 8 have to play with each other in the first game of upper bracket (game d). Team 8 can win all remaining games in upper bracket, when teams 1 and 7 will compete in the lower bracket.



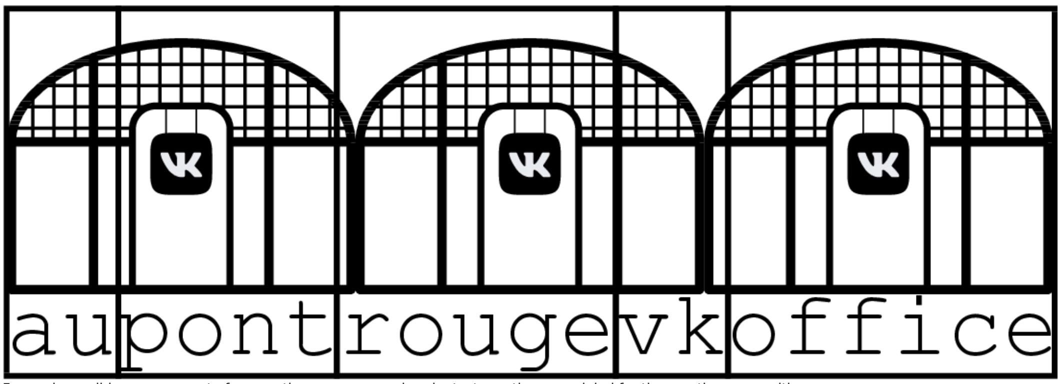
In the third example, your favourite teams can play in all games of the championship.



F. Au Pont Rouge

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

VK just opened its second HQ in St. Petersburg! Side of its office building has a huge string s written on its side. This part of the office is supposed to be split into m meeting rooms in such way that meeting room walls are strictly between letters on the building. Obviously, meeting rooms should not be of size 0, but can be as small as one letter wide. Each meeting room will be named after the substring of s written on its side.



For each possible arrangement of m meeting rooms we ordered a test meeting room label for the meeting room with lexicographically **minimal** name. When delivered, those labels got sorted **backward** lexicographically.

What is printed on kth label of the delivery?

Input

In the first line, you are given three integer numbers n, m, k — length of string s, number of planned meeting rooms to split s into and number of the interesting label ($2 \le n \le 1\,000; 1 \le m \le 1\,000; 1 \le k \le 10^{18}$).

Second input line has string s, consisting of n lowercase english letters.

For given n, m, k there are at least k ways to split s into m substrings.

Output

Output single string – name of meeting room printed on k-th label of the delivery.

Examples

input
```
4 2 1
abac
```
output
```
aba
```

input
```
19 5 1821
aupontrougevkoffice
```
output
```
au
```

Note

In the first example, delivery consists of the labels "aba", "ab", "a".

In the second example, delivery consists of 3060 labels. The first label is "aupontrougevkof" and the last one is "a".