unioeste

Maratoninha

1ª Competição – 14 Julho 2023

8h30 – 13h00

Maratona de Programação

PROBLEMA A

B. Atilla's Favorite Problem

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

In order to write a string, Atilla needs to first learn all letters that are contained in the string.

Atilla needs to write a message which can be represented as a string s. He asks you what is the minimum alphabet size required so that one can write this message.

The alphabet of size x ($1 \leq x \leq 26$) contains **only the first** x Latin letters. For example an alphabet of size 4 contains **only** the characters $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{d}$.

Input

The first line contains a single integer t ($1 \leq t \leq 1000$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 100$) — the length of the string.

The second line of each test case contains a string s of length n, consisting of lowercase Latin letters.

Output

For each test case, output a single integer — the minimum alphabet size required to so that Atilla can write his message s.

Example

input Copy

```
5
1
a
4
down
10
codeforces
3
bcf
5
zzzzz
```

output Copy

```
1
23
19
6
26
```

Note

For the first test case, Atilla needs to know only the character $\mathbf{a}$, so the alphabet of size 1 which only contains $\mathbf{a}$ is enough.

For the second test case, Atilla needs to know the characters $\mathbf{d}$, $\mathbf{o}$, $\mathbf{w}$, $\mathbf{n}$. The smallest alphabet size that contains all of them is 23 (such alphabet can be represented as the string $\mathbf{abcdefghijklmnopqrstuvw}$).

PROBLEMA B

A. Watermelon

time limit per test: 1 second
memory limit per test: 64 megabytes
input: standard input
output: standard output

One hot summer day Pete and his friend Billy decided to buy a watermelon. They chose the biggest and the ripest one, in their opinion. After that the watermelon was weighed, and the scales showed w kilos. They rushed home, dying of thirst, and decided to divide the berry, however they faced a hard problem.

Pete and Billy are great fans of even numbers, that's why they want to divide the watermelon in such a way that each of the two parts weighs even number of kilos, at the same time it is not obligatory that the parts are equal. The boys are extremely tired and want to start their meal as soon as possible, that's why you should help them and find out, if they can divide the watermelon in the way they want. For sure, each of them should get a part of positive weight.

Input

The first (and the only) input line contains integer number w $(1 \leq w \leq 100)$ — the weight of the watermelon bought by the boys.

Output

Print `YES`, if the boys can divide the watermelon into two parts, each of them weighing even number of kilos; and `NO` in the opposite case.

Examples

input Copy

```
8
```

output Copy

```
YES
```

Note

For example, the boys can divide the watermelon into two parts of 2 and 6 kilos respectively (another variant — two parts of 4 and 4 kilos).

PROBLEMA C

B. Even-Odd Increments

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given n of integers $a_1, a_2, \ldots, a_n$. Process q queries of two types:

- query of the form "0 x_j": add the value x_j to all even elements of the array a,
- query of the form "1 x_j": add the value x_j to all odd elements of the array a.

Note that when processing the query, we look specifically at the odd/even value of a_i, not its index.

After processing each query, print the sum of the elements of the array a.

Please note that the answer for some test cases won't fit into 32-bit integer type, so you should use at least 64-bit integer type in your programming language (like `long long` for C++).

Input

The first line of the input contains an integer t $(1 \le t \le 10^4)$ — the number of test cases.

The descriptions of the test cases follow.

The first line of each test case contains two integers n and q $(1 \le n, q \le 10^5)$ — the length of array a and the number of queries.

The second line of each test case contains exactly n integers: $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$ — elements of the array a.

The following q lines contain queries as two integers $type_j$ and x_j $(0 \le type_j \le 1$, $1 \le x_j \le 10^4)$.

It is guaranteed that the sum of values n over all test cases in a test does not exceed 10^5. Similarly, the sum of values q over all test cases does not exceed 10^5.

Output

For each test case, print q numbers: the sum of the elements of the array a after processing a query.

Example

input

```
4
1 1
1
1 1
3 3
1 2 4
0 2
1 3
0 5
6 7
1 3 2 4 10 48
1 6
0 5
0 4
0 5
1 3
0 12
0 1
6 7
1000000000 1000000000 1000000000 11 15 17
0 17
1 10000
1 51
0 92
0 53
1 16
0 1
```

output

```
2
11
14
29
80
100
100
100
118
190
196
3000000094
3000060094
3000060400
3000060952
3000061270
3000061366
3000061366
```

Note

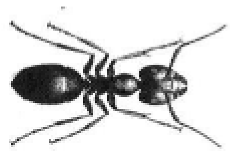
In the first test case, the array $a = [2]$ after the first query.

In the third test case, the array a is modified as follows: $[1, 3, 2, 4, 10, 48] \rightarrow [7, 9, 2, 4, 10, 48] \rightarrow [7, 9, 7, 9, 15, 53] \rightarrow [7, 9, 7, 9, 15, 53] \rightarrow [10, 12, 10, 12, 18, 56] \rightarrow [22, 24, 22, 24, 30, 68] \rightarrow [23, 25, 23, 25, 31, 69]$.

PROBLEMA D

‹ Hide

Ants



An army of ants walk on a horizontal pole of length l cm, each with a constant speed of 1 cm/s. When a walking ant reaches an end of the pole, it immediatelly falls off it. When two ants meet they turn back and start walking in opposite directions. We know the original positions of ants on the pole, unfortunately, we do not know the directions in which the ants are walking. Your task is to compute the earliest and the latest possible times needed for all ants to fall off the pole.

Input

The first line of input contains one integer giving the number of cases that follow, at most 100. The data for each case start with two integer numbers: the length l of the pole (in cm) and n, the number of ants residing on the pole. These two numbers are followed by n integers giving the position of each ant on the pole as the distance measured from the left end of the pole, in no particular order. All input integers are between 0 and 1 000 000 and they are separated by whitespace.

Output

For each case of input, output two numbers separated by a single space. The first number is the earliest possible time when all ants fall off the pole (if the directions of their walks are chosen appropriately) and the second number is the latest possible such time.

Sample Input 1

```
2
10 3
2 6 7
214 7
11 12 7 13
176 23 191
```

Sample Output 1

```
4 8
38 207
```

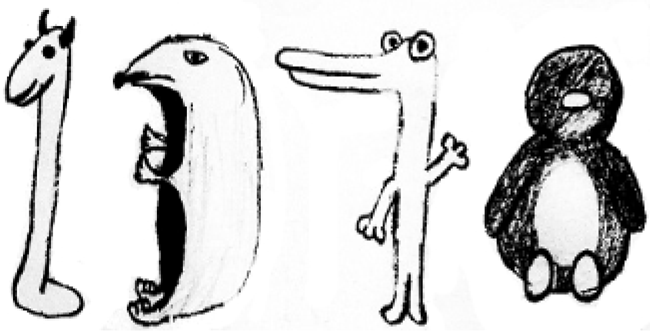
PROBLEMA E

A. Arpa's hard exam and Mehrdad's naive cheat

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

There exists an island called Arpa's land, some beautiful girls live there, as ugly ones do.

Mehrdad wants to become minister of Arpa's land. Arpa has prepared an exam. Exam has only one question, given n, print the last digit of 1378^n.



Mehrdad has become quite confused and wants you to help him. Please help, although it's a naive cheat.

Input

The single line of input contains one integer n ($0 \le n \le 10^9$).

Output

Print single integer — the last digit of 1378^n.

Examples

input	Copy
1	
output	Copy
8	

input	Copy
2	
output	Copy
4	

Note

In the first example, last digit of $1378^1 = 1378$ is 8.

In the second example, last digit of $1378^2 = 1378 \cdot 1378 = 1898884$ is 4.

PROBLEMA F

B. Rudolph and Tic-Tac-Toe

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

Rudolph invented the game of tic-tac-toe for three players. It has classic rules, except for the third player who plays with pluses. Rudolf has a 3×3 field — the result of the completed game. Each field cell contains either a cross, or a nought, or a plus sign, or nothing. The game is won by the player who makes a horizontal, vertical or diagonal row of 3's of their symbols.

Rudolph wants to find the result of the game. Either exactly one of the three players won or it ended in a draw. It is guaranteed that multiple players cannot win at the same time.

Input

The first line contains one integer t $(1 \leq t \leq 10^4)$ — the number of test cases.

Each test case consists of three lines, each of which consists of three characters. The symbol can be one of four: "X" means a cross, "O" means a nought, "+" means a plus, "." means an empty cell.

Output

For each test case, print the string "X" if the crosses won, "O" if the noughts won, "+" if the pluses won, "DRAW" if there was a draw.

Example

input Copy

```
5
+X+
OXO
OX.
O+.
+OX
X+O
.XO
OX.
+++
O.+
X.O
+..
.++
X.O
+..
```

output Copy

```
X
O
+
DRAW
DRAW
```

PROBLEMA G

C. Traffic Light

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

You find yourself on an unusual crossroad with a weird traffic light. That traffic light has three possible colors: red (r), yellow (y), green (g). It is known that the traffic light repeats its colors every n seconds and at the i-th second the color s_i is on.

That way, the order of the colors is described by a string. For example, if $s =$"rggry", then the traffic light works as the following: red-green-green-red-yellow-red-green-green-red-yellow- ... and so on.

More formally, you are given a string $s_1, s_2, \ldots, s_n$ of length n. At the first second the color s_1 is on, at the second — s_2, ..., at the n-th second the color s_n is on, at the $n+1$-st second the color s_1 is on and so on.

You need to cross the road and that can only be done when the green color is on.

You know which color is on the traffic light at the moment, but you don't know the current moment of time. You need to find the minimum amount of time in which you are guaranteed to cross the road.

You can assume that you cross the road immediately.

For example, with $s =$"rggry" and the current color r there are two options: either the green color will be on after 1 second, or after 3. That way, the answer is equal to 3 — that is the number of seconds that we are guaranteed to cross the road, if the current color is r.

Input

The first line contains a single integer t $(1 \leq t \leq 10^4)$ — the number of test cases.

Then the description of the test cases follows.

The first line of each test case contains an integer n and a symbol c ($1 \leq n \leq 2 \cdot 10^5$, c is one of allowed traffic light colors r, y or g)— the length of the string s and the current color of the traffic light.

The second line of each test case contains a string s of the length n, consisting of the letters r, y and g.

It is guaranteed that the symbol g is in the string s and the symbol c is in the string s.

It is guaranteed, that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case output the minimal number of second in which you are guaranteed to cross the road.

Example

input Copy

```
6
5 r
rggry
1 g
g
3 r
rrg
5 y
yrrgy
7 r
rgrgyrg
9 y
rrrgyyygy
```

output

Copy

```
3
0
2
4
1
4
```

Note

The first test case is explained in the statement.

In the second test case the green color is on so you can cross the road immediately.

In the third test case, if the red color was on at the second second, then we would wait for the green color for one second, and if the red light was on at the first second, then we would wait for the green light for two seconds.

In the fourth test case the longest we would wait for the green color is if we wait for it starting from the fifth second.

PROBLEMA H

CSES Problem Set

Array Division

TASK | STATISTICS

Time limit: 1.00 s **Memory limit:** 512 MB

You are given an array containing n positive integers.

Your task is to divide the array into k subarrays so that the maximum sum in a subarray is as small as possible.

Input

The first input line contains two integers n and k: the size of the array and the number of subarrays in the division.

The next line contains n integers $x_1, x_2, \ldots, x_n$: the contents of the array.

Output

Print one integer: the maximum sum in a subarray in the optimal division.

Constraints

- $1 \leq n \leq 2 \cdot 10^5$
- $1 \leq k \leq n$
- $1 \leq x_i \leq 10^9$

Example

Input:
```
5 3
2 4 7 3 5
```

Output:
```
8
```

Explanation: An optimal division is $[2, 4], [7], [3, 5]$ where the sums of the subarrays are $6, 7, 8$. The largest sum is the last sum 8.

PROBLEMA I

CSES Problem Set

Tree Diameter

TASK | STATISTICS

Time limit: 1.00 s **Memory limit:** 512 MB

You are given a tree consisting of n nodes.

The *diameter* of a tree is the maximum distance between two nodes. Your task is to determine the diameter of the tree.

Input

The first input line contains an integer n: the number of nodes. The nodes are numbered $1, 2, \ldots, n$.

Then there are $n - 1$ lines describing the edges. Each line contains two integers a and b: there is an edge between nodes a and b.

Output

Print one integer: the diameter of the tree.

Constraints

- $1 \leq n \leq 2 \cdot 10^5$
- $1 \leq a, b \leq n$

Example

Input:
```
5
1 2
1 3
3 4
3 5
```

Output:
```
3
```

Explanation: The diameter corresponds to the path $2 \to 1 \to 3 \to 5$.

PROBLEMA J

CSUMQ - Cumulative Sum Query

no tags

William Macfarlane wants to look at an array.

You are given a list of **N** numbers and **Q** queries. Each query is specified by two numbers *i* and *j*; the answer to each query is the sum of every number between the range [i, j] (inclusive).

Note: the query ranges are specified using 0-based indexing.

Input

The first line contains **N**, the number of integers in our list (**N <= 100,000**). The next line holds **N** numbers that are guaranteed to fit inside an integer. Following the list is a number **Q** (**Q <= 10,000**). The next **Q** lines each contain two numbers **i** and **j** which specify a query you must answer (**0 <= i, j <= N-1**).

Output

For each query, output the answer to that query on its own line in the order the queries were made.

Example

```
Input:
3
1 4 1
3
1 1
1 2
0 2

Output:
4
5
6
```

PROBLEMA K

ABCPATH - ABC Path

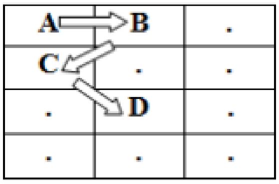
no tags

You will be given a 2-dimensional grid of letters. Find the length of the longest path of consecutive letters, starting at 'A'. Paths can step from one letter in the grid to any adjacent letter (horizontally, vertically, or diagonally).

For example, in the following grid, there are several paths from 'A' to 'D', but none from 'A' to 'E':

A	**B**	**E**
C	**F**	**G**
B	**D**	**H**
A	**B**	**C**

One such path is:



Input

Each test case will start with a line contains two integers H, W the height and width of the grid respectively 1 <= H, W <= 50. Then H lines follow each of W uppercase letters only. Input terminates with H = 0 and W = 0.

Output

For each test case print "`Case C: X`" without quotes where C is the case number starting with 1 and X is the solution.

Example

```
Sample Input:
4 3
ABE
CFG
BDH
ABC
0 0

Sample Output:
Case 1: 4
```

PROBLEMA L

C - Socks Editorial

Time Limit: 4 sec / Memory Limit: 1024 MB

Score : 300 points

Problem Statement

You have N socks. The color of the i-th sock is A_i.

You want to perform the following operation as many times as possible. How many times can it be performed at most?

- Choose two same-colored socks that are not paired yet, and pair them.

Constraints

- $1 \leq N \leq 5 \times 10^5$
- $1 \leq A_i \leq 10^9$
- All values in the input are integers.

Input

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

N
$A_1 \ A_2 \ \ldots \ A_N$

Output

Print an integer representing the answer.

Sample Input 1 Copy

```
6
4 1 7 4 1 4
```

Copy

Sample Output 1 Copy

```
2
```

Copy

You can do the operation twice as follows.

- Choose two socks with the color 1 and pair them.
- Choose two socks with the color 4 and pair them.

Then, you will be left with one sock with the color 4 and another with the color 7, so you can no longer do the operation. There is no way to do the operation three or more times, so you should print 2.

Sample Input 2 Copy

```
1
158260522
```

Copy

Sample Output 2 Copy

```
0
```

Copy

Sample Input 3 Copy

```
10
295 2 29 295 29 2 29 295 2 29
```

Copy

Sample Output 3 Copy

```
4
```

Copy