# Codeforces Round 1017 (Div. 4)

## A. Trippi Troppi

1 second, 256 megabytes

Trippi Troppi resides in a strange world. The ancient name of each country consists of three strings. The first letter of each string is concatenated to form the country's modern name.

Given the country's ancient name, please output the modern name.

### Input
The first line contains an integer $t$ – the number of independent test cases ($1 \le t \le 100$).

The following $t$ lines each contain three space-separated strings. Each string has a length of no more than $10$, and contains only lowercase Latin characters.

### Output
For each test case, output the string formed by concatenating the first letter of each word.

| input |
| --- |
| 7<br>united states america<br>oh my god<br>i cant lie<br>binary indexed tree<br>believe in yourself<br>skibidi slay sigma<br>god bless america |

| output |
| --- |
| usa<br>omg<br>icl<br>bit<br>biy<br>sss<br>gba |

## B. Bobritto Bandito

1 second, 256 megabytes

In Bobritto Bandito's home town of residence, there are an infinite number of houses on an infinite number line, with houses at $\ldots, -2, -1, 0, 1, 2, \ldots$. On day $0$, he started a plague by giving an infection to the unfortunate residents of house $0$. Each succeeding day, the plague spreads to **exactly one** healthy household that is next to an infected household. It can be shown that each day the infected houses form a continuous segment.

Let the segment starting at the $l$-th house and ending at the $r$-th house be denoted as $[l, r]$. You know that after $n$ days, the segment $[l, r]$ became infected. Find any such segment $[l', r']$ that could have been infected on the $m$-th day ($m \le n$).

### Input
The first line contains an integer $t$ ($1 \le t \le 100$) – the number of independent test cases.

The only line of each test case contains four integers $n$, $m$, $l$, and $r$ ($1 \le m \le n \le 2000, -n \le l \le 0 \le r \le n, r - l = n$).

### Output
For each test case, output two integers $l'$ and $r'$ on a new line. If there are multiple solutions, output any.

| input |
| --- |
| 4<br>4 2 -2 2<br>4 1 0 4<br>3 3 -1 2<br>9 8 -6 3 |

| output |
| --- |
| -1 1<br>0 1<br>-1 2<br>-5 3 |

In the first test case, it is possible that on the $1$-st, $2$-nd, and $3$-rd days the interval of houses affected is $[-1, 0]$, $[-1, 1]$, $[-2, 1]$. Therefore, $[-1, 1]$ is a valid output.

## C. Brr Brrr Patapim

2 seconds, 256 megabytes

Brr Brrr Patapim is trying to learn of Tiramisù's secret passcode, which is a permutation* of $2 \cdot n$ elements. To help Patapim guess, Tiramisù gave him an $n \times n$ grid $G$, in which $G_{i,j}$ (or the element in the $i$-th row and $j$-th column of the grid) contains $p_{i+j}$, or the $(i + j)$-th element in the permutation.

Given this grid, please help Patapim crack the forgotten code. It is guaranteed that the permutation exists, and it can be shown that the permutation can be determined uniquely.

*A permutation of $m$ integers is a sequence of $m$ integers which contains each of $1, 2, \ldots, m$ exactly once. For example, $[1, 3, 2]$ and $[2, 1]$ are permutations, while $[1, 2, 4]$ and $[1, 3, 2, 3]$ are not.

### Input
The first line contains an integer $t$ — the number of test cases ($1 \le t \le 200$).

The first line of each test case contains an integer $n$ ($1 \le n \le 800$).

Each of the following $n$ lines contains $n$ integers, giving the grid $G$. The first of these lines contains $G_{1,1}, G_{1,2}, \ldots, G_{1,n}$; the second of these lines contains $G_{2,1}, G_{2,2}, \ldots, G_{2,n}$, and so on. ($1 \le G_{i,j} \le 2 \cdot n$).

It is guaranteed that the grid encodes a valid permutation, and the sum of $n$ over all test cases does not exceed $800$.

### Output
For each test case, please output $2n$ numbers on a new line: $p_1, p_2, \ldots, p_{2n}$.

| input |
| --- |
| 3<br>3<br>1 6 2<br>6 2 4<br>2 4 3<br>1<br>1<br>2<br>2 3<br>3 4 |

| output |
| --- |
| 5 1 6 2 4 3<br>2 1<br>1 2 3 4 |

## D. Tung Tung Sahur

2 seconds, 256 megabytes

You have two drums in front of you: a left drum and a right drum. A hit on the left can be recorded as "L", and a hit on the right as "R".

The strange forces that rule this world are fickle: sometimes, a blow sounds once, and sometimes it sounds twice. Therefore, a hit on the left drum could have sounded as either "L" or "LL", and a hit on the right drum could have sounded as either "R" or "RR".

The sequence of hits made is recorded in the string $p$, and the sounds heard are in the string $s$. Given $p$ and $s$, determine whether it is true that the string $s$ could have been the result of the hits from the string $p$.

For example, if $p =$"LR", then the result of the hits could be any of the strings "LR", "LRR", "LLR", and "LLRR", but the strings "LLLR" or "LRL" cannot.

### Input
The first line contains an integer $t$ ($1 \le t \le 10^4$) – the number of independent test cases.

The first line of each test case contains the string $p$ ($1 \le |p| \le 2 \cdot 10^5$) consisting only of the characters "R" and "L", where $|p|$ denotes the length of the string $p$.

The second line of each test case contains the string $s$ ($1 \le |p| \le |s| \le 2 \cdot 10^5$) consisting only of the characters "R" and "L".

It is guaranteed that the sum of $|s|$ does not exceed $2 \cdot 10^5$ across all test cases.

### Output
For each set of input data, output "YES" if $s$ can be the heard sound, and "NO" otherwise. You may output in any case.

| input |
|---|
| 5<br>R<br>RR<br>LRLR<br>LRLR<br>LR<br>LLLR<br>LLLLLRL<br>LLLLRRLL<br>LLRLRLRRL<br>LLLRLRRLLRRRL |

| output |
|---|
| YES<br>YES<br>NO<br>NO<br>YES |

## E. Boneca Ambalabu

2 seconds, 256 megabytes

Boneca Ambalabu gives you a sequence of $n$ integers $a_1, a_2, \ldots, a_n$.

Output the maximum value of $(a_k \oplus a_1) + (a_k \oplus a_2) + \ldots + (a_k \oplus a_n)$ among all $1 \le k \le n$. Note that $\oplus$ denotes the bitwise XOR operation.

### Input
The first line contains an integer $t$ ($1 \le t \le 10^4$) – the number of independent test cases.

The first line of each test case contains an integer $n$ ($1 \le n \le 2 \cdot 10^5$) – the length of the array.

The second line of each test case contains $n$ integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 2^{30}$).

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 maximum value on a new line.

| input |
|---|
| 5<br>3<br>18 18 18<br>5<br>1 2 4 8 16<br>5<br>8 13 4 5 15<br>6<br>625 676 729 784 841 900<br>1<br>1 |

| output |
|---|
| 0<br>79<br>37<br>1555<br>0 |

In the first test case, the best we can do is $(18 \oplus 18) + (18 \oplus 18) + (18 \oplus 18) = 0$.

In the second test case, we choose $k = 5$ to get $(16 \oplus 1) + (16 \oplus 2) + (16 \oplus 4) + (16 \oplus 8) + (16 \oplus 16) = 79$.

## F. Trulimero Trulicina

2 seconds, 256 megabytes

Trulicina gives you integers $n$, $m$, and $k$. It is guaranteed that $k \ge 2$ and $n \cdot m \equiv 0 \pmod{k}$.

Output a $n$ by $m$ grid of integers such that each of the following criteria hold:

- Each integer in the grid is between $1$ and $k$, inclusive.
- Each integer from $1$ to $k$ appears an equal number of times.
- No two cells that share an edge have the same integer.

It can be shown that such a grid always exists. If there are multiple solutions, output any.

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

The first line of each test case contains three integers $n$, $m$, and $k$ ($2 \le n \cdot m \le 2 \cdot 10^5, 2 \le k \le n \cdot m, n \cdot m \equiv 0 \pmod{k}$).

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

### Output
For each test case, output $n$ lines, each containing $m$ integers that satisfy the criteria. If there are multiple solutions, output any.

| input |
|---|
| 3<br>2 2 2<br>3 4 6<br>5 5 25 |

| output |
|---|
| 1 2<br>2 1<br>1 6 1 6<br>2 5 2 5<br>3 4 3 4<br>17 2 12 25 14<br>3 1 6 19 11<br>8 20 23 24 4<br>9 10 5 13 21<br>22 7 15 18 16 |

# G. Chimpanzini Bananini

2 seconds, 256 megabytes

*Chimpanzini Bananini stands on the brink of a momentous battle—one destined to bring finality.*

For an arbitrary array $b$ of length $m$, let's denote the *rizziness* of the array to be $\sum_{i=1}^{m} b_i \cdot i = b_1 \cdot 1 + b_2 \cdot 2 + b_3 \cdot 3 + \ldots + b_m \cdot m$.

Chimpanzini Bananini gifts you an empty array. There are three types of operations you can perform on it.

1. Perform a cyclic shift on the array. That is, the array $[a_1, a_2, \ldots, a_n]$ becomes $[a_n, a_1, a_2, \ldots, a_{n-1}]$.
2. Reverse the entire array. That is, the array $[a_1, a_2, \ldots, a_n]$ becomes $[a_n, a_{n-1}, \ldots, a_1]$.
3. Append an element to the end of the array. The array $[a_1, a_2, \ldots, a_n]$ becomes $[a_1, a_2, \ldots, a_n, k]$ after appending $k$ to the end of the array.

After each operation, you are interested in calculating the *rizziness* of your array.

Note that all operations are **persistent**. This means that each operation modifies the array, and subsequent operations should be applied to the current state of the array after the previous operations.

## Input

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

The first line of the input contains an integer $q$ ($1 \leq q \leq 2 \cdot 10^5$) — the number of operations you perform on your array.

The following $q$ lines first contain a single integer $s$ ($1 \leq s \leq 3$) — the operation type.

- If $s = 1$, then the cyclic shift operation should be performed.
- If $s = 2$, then the reversal operation should be performed.
- If $s = 3$, then the line will contain an additional integer $k$ ($1 \leq k \leq 10^6$), denoting the element appended to the back of the array.

It is guaranteed that the sum of $q$ will not exceed $2 \cdot 10^5$ over all test cases. Additionally, it is guaranteed that the first operation on each test case will be one with $s = 3$.

## Output

For each test case, output $q$ lines, outputting the *rizziness* of your array after each operation.

**input**

```
1
13
3 1
3 2
3 3
1
3 4
2
3 5
1
3 6
2
3 7
2
1
```

**output**

```
1
5
14
11
27
23
48
38
74
73
122
102
88
```

The first six states of the array:

- $[1]$
- $[1, 2]$
- $[1, 2, 3]$
- $[3, 1, 2]$
- $[3, 1, 2, 4]$
- $[4, 2, 1, 3]$

# H. La Vaca Saturno Saturnita

4 seconds, 256 megabytes

Saturnita's mood depends on an array $a$ of length $n$, which only he knows the meaning of, and a function $f(k, a, l, r)$, which only he knows how to compute. Shown below is the pseudocode for his function $f(k, a, l, r)$.

```
function f(k, a, l, r):
   ans := 0
   for i from l to r (inclusive):
      while k is divisible by a[i]:
         k := k/a[i]
      ans := ans + k
   return ans
```

You are given $q$ queries, each containing integers $k$, $l$, and $r$. For each query, please output $f(k, a, l, r)$.

## Input

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

The first line of each test case contains two integers $n$ and $q$ ($1 \leq n \leq 10^5, 1 \leq q \leq 5 \cdot 10^4$).

The following line contains $n$ integers $a_1, a_2, \ldots, a_n$ ($2 \leq a_i \leq 10^5$).

The following $q$ lines each contain three integers $k$, $l$, and $r$ ($1 \leq k \leq 10^5, 1 \leq l \leq r \leq n$).

It is guaranteed that the sum of $n$ does not exceed $10^5$ over all test cases, and the sum of $q$ does not exceed $5 \cdot 10^4$ over all test cases.

## Output

For each query, output the answer on a new line.

**input**

```
2
5 3
2 3 5 7 11
2 1 5
2 2 4
2310 1 5
4 3
18 12 8 9
216 1 2
48 2 4
82944 1 4
```

**output**

```
5
6
1629
13
12
520
```