BUET Inter University Programming Contest 2022

https://toph.co/c/buet-inter-university-2022

Toph

A. Chop Chop

There is a convex polygon A of n vertices and you want to divide it into two convex polygons P and Q such that their areas are equal (half of A) and the sum of perimeters of P and Q is minimum possible.

Input

The first line of the input is an integer T which denotes the number of test cases.

For each test case, the first line contains an integer n, denoting the number of vertices in the polygon A. n lines follow, i^{th} line containing two real numbers (with four digit precision after decimal point) $x_i,\ y_i$ denoting the co-ordinate of the i^{th} vertex of A in a counter-clockwise manner.

Constraints:

- $1 \leq T \leq 100.$
- $3 \leq n \leq 2 \times 10^5.$
- $-10^7 \leq x_i,\ y_i \leq 10^7.$
- Sum of n across all testcases is less than or equal to 2×10^5.

Output

For each test case, output the minimum total sum of perimeters of P and Q on a single line.

Your output will be considered correct if the absolute or relative error is less than or equal to 10^{-6}.

Samples

Input

```
1
3
0.0000 0.0000
1000.0000 0.0000
500.0000 1.0000
```

Output

```
2002.0019989980
```

B. Up You Go

Arya is taking part in a game of throwing marbles upwards. There are n participants in this game in total, where Arya is the participant with label 1. Each participant is given a marble and initially each of the marbles are held at ground level (height 0). Each participant x throws his marble straight upwards at time s_x with initial velocity u_x and it drops back on the ground after some time.

The marble will follow basic rules of physics while travelling in the air. So if a marble hasn't yet dropped back on the ground and t time units have passed since it was thrown with initial velocity u, its current velocity v and current height h can be expressed as follows:

$$v = u - gt$$

$$h = ut - 0.5 * gt^2$$

Here, $g = 10$.

You have to answer q queries, where you will be given an integer T in each query.

For each query, count how many marbles have a strictly greater height than Arya's marble at time T.

Input

First line of the input contains two space-separated integers n and q, denoting the number of participants and the number of queries.

Next n lines each contains two integers s_x and u_x, denoting when participant x threw the marble and its initial velocity.

Next q lines each contains one integer T, denoting the query time.

$1 \leq n, q \leq 10^5$

$0 \leq u_x, s_x, T \leq 10^8$

Output

For each query, output a single line containing one single integer denoting the number of marbles with a strictly greater height than Arya's marble at the query time.

Samples

Input	Output
5 10 5 40 4 50 3 50 2 50 1 50 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 4 4 4 3 3

C. Peculiar Partitioning

You are given an array a of length n. You need to find a sub-sequence S of the array such that $gcd(sum(S), sum(\overline{S})) > 1$. In other words, the gcd of sum of elements in S and sum of elements not in S is greater than 1. It is also required that both S and $\overline{S}$ are non-empty.

Input

The first line contains an integer, $n(1 \leq n \leq 10^5)$.

Check Toph description

The second line contains n space-separated integers $a_1, a_2, \cdots, a_n (1 \leq a_i \leq 2000)$.

Output

If there is no sub-sequence of the array such that print a single word "No".

Otherwise, on the first line print a single word "Yes". On the second the line print a single integer k $(1 \leq k \leq n-1)$, the length of the chosen subsequence. On the third line print k space seperated distinct integers $c_1, c_2, \cdots c_k$ $(1 \leq c_i \leq n)$, the indices of the elements of the chosen subsequence.

DP

If there are multiple solutions, print any of them. You can print the indices of the chosen subsequence in any order.

17

5 8

Samples

Input	Output
5 6 2 3 5 8	Yes 3 4 2 3

Here a_4, a_2, a_3 are the chosen elements. Their sum is $5+2+3=10$. The other elements sum to $6+8=14$. Since, $gcd(10, 14) > 1$, this is a valid answer.

$[1,5]$, $[3,4]$, $[1,2,5,4]$ are also valid subsequences.

Input	Output
2 1 3	No

D. Yet Another Frog Jumping Problem

This is an interactive problem. (If you want to know about interactive problems, click here)

There are N stones. Stones are enumerated from 1 to N. There is an intelligent frog too. Jumping capability of the frog never decreases, but it may change at most 39 times. If his jumping capability is D on a given day, he can go **exactly** to stone $x + D$ from stone x in a valid jump, if $x + D$ is not greater than N.

Initially, all of the stones are uncolored. The frog wishes to color all of the stones. Every morning, the frog discovers himself in the first uncolored stone (i.e. uncolored stone with smallest index). During that whole day, his jumping capability does not change and he keeps jumping **as long as it can**. Whenever he reaches an uncolored stone, he colors it with color D if his jumping capability is D during that day. However, the frog is clever. Whenever he reaches an already colored stone, he does not color it again and keeps moving (Note that he must color initial stone of that day). He stops jumping for that day, only if jumping from his current stone is not valid.

Whenever all of the stones are colored, the frog goes back to play with his friends.

You do not know the number of days to complete the task, you do not even know the jumping capability of the frog each day. You can do 720 queries, asking for color of a stone.

Retrieve the number of days to complete the task and color of every stone.

Input

First line contains an integer T $(1 \leq T \leq 1000)$, the number of test cases. For each case, first line contains an integer N $(1 \leq N \leq 10^5)$, the number of stones.

Sum of N over all test cases does not exceed 10^6.

Output

You can ask at most 720 queries for each test case. To ask a query just print an integer, i $(1 \leq i \leq N)$, then read an integer C $(1 \leq C \leq 10^9)$, the color of the stone i. There will be at most 40 distinct colors for each test case.

When you can determine the color of all stones print 0 (which is not counted as a query). In the next line, **print the number of days** the frog needed to color all stones.

In the next line print N integers, **separated by spaces**, $C_1, C_2, \cdots, C_N$ where C_i is the color of stone i.

After that read an integer, res. If res is -1, then it means your output is wrong for that test case and you should terminate the program immediately and you will receive Wrong Answer verdict. If res is not -1, then process the next test case. If at any moment the number of query asked is greater than 720 or an invalid query is asked, then you will receive an integer -1 and you should terminate the program immediately and you will receive Wrong Answer verdict.

Imagine a valid example, where $T = 1$, $N = 5$ and colors of the stones are $[2, 3, 2, 3, 2]$.

Interaction may happen as —,

> 1

> 5

< 1

> 2

< 4

> 3

< 0

< 3

$< 2\ 3\ 2\ 3\ 2$

> 1

It takes the frog 3 days to complete.

$>$ indicates what your program reads and $<$ indicates what your program writes. These symbols are here to make it easy to understand. You do not have to print such symbols from your program.

E. Make More Money

You, the great businessman of TEUB, arrived at ESC land which consists of N cities numbered from 1 to N. You are currently at city 1. For each city i, you arrive at (including your current city) in ESC land, you do some business and make profit a_i. The profit can be negative which means you have lost $|a_i|$ amount of money in your business.But there is a special rule for travelling ESC land:

1. First you can go from your current location i.e. city 1 to city i through the path $1, 2, 3, \cdots, i \,(1 \leq i \leq N)$.
2. Then you can go from city i to city j through the path $i, i-1, i-2, \cdots, j$ ($1 \leq j < i$).

So, your profit will be $a_1 + a_2 + a_3 + \ldots + a_i + a_{i-1} + a_{i-2} + \cdots + a_j$

Note that you may stay at city 1 and you may also choose not to go to reverse direction (i.e. direction to which city index decreases).

Find out maximum profit you can achieve.

J < 1

Input

First line will be number of test cases T

For each test case:

In the first line, the number of cities N will be given.

Then there will be N integers $a_1, a_2, \cdots, a_N$ where a_i denotes achievable profit from city i at a time.

Constraints

$1 \leq T \leq 10$

$1 \leq N \leq 10^5$

$-10^4 \leq a_i \leq 10^4$

Over all test cases, $\sum N \leq 2 * 10^5$

Output

For each test case, print one line containing maximum possible profit you can gain.

Samples

Input	Output
4 1 4 2 4 -2 3 5 2 -10 4 -1 3 -2 4	4 6 12 5

Optimal paths for the 4 testcases are:

1. path: 1, profit: 4
2. path: $1 \to 2 \to 1$, profit: $4 + (-2) + 4 = 6$
3. path: $1 \to 2 \to 1$, profit: $5 + 2 + 5 = 12$
4. path: $1 \to 2 \to 3 \to 4 \to 3 \to 2$, profit: $(-1) + 3 + (-2) + 4 + (-2) + 3 = 5$

F. Choco Fun

You brought C chocolates to a party where there are A adults and K kids. **You know that an adult will always eat the chocolate fully and a kid will always eat the chocolate partially.** For each chocolate, you know how much happiness it brings to one who eats fully or partially. You want to give everyone **exactly one chocolate** so that the total happiness they get is maximized.

Input

The first line contains one integer T - the number of test cases. For each test case, the first line contains 3 integers C, A, and K denoting the number of chocolates, adults, and kids respectively. For the next C lines, $i'th$ line contains two integers F_i and P_i, denoting the happiness one gets if he eats the $i'th$ chocolate fully and partially respectively.

$1 \le T \le 10^6$,
$1 \le C \le 10^5$,
$0 \le A, K \le 10^5$,
$A + K \le C$,
$0 \le P_i \le F_i \le 10^9$,

The sum of C over all test cases does not exceed 10^6

Output

For each test case, output in a line, the maximum sum of happiness they get. See sample for details.

Samples

Input

```
2
2 1 1
30 12
20 1
3 1 1
20 10
12 10
15 15
```

Output

```
32
35
```

100

11

G. Red Blue Graph

You are given a connected, undirected, unweighted graph. You need to assign one of the two colors Red or Blue to every edge of the graph in such a way that, for each node $|degree_R - degree_B| \leq 1$.
$degree_R$ of a node is the number of adjacent edges to that node which are colored Red.
$degree_B$ of a node is the number of adjacent edges to that node which are colored Blue.

Input

The first line contains one integer T - the number of test cases. First line of each testcase contains two integers N, M - the number of vertices and the number of edges in the graph respectively. Each of the next M lines contain two integers u_i and v_i describing an undirected edge in the graph.

$1 \leq T \leq 100$
$1 \leq N \leq 10^5$
$N - 1 \leq M \leq min(3 * 10^5, (N * (N - 1))/2)$
Sum of N over all testcases doesn't exceed 10^6 and sum of M over all testcases also doesn't exceed 10^6.
It is guaranteed that the graph is connected, it has no self-loop and there is at most one edge between every pair of vertices.

Output

If it is not possible to find such a coloring, print -1. Otherwise print one such coloring of the edges in the order given. If there exists more than one coloring you can print any one. You should print a string containing only R and B to denote the coloring.

Samples

Input

```
2
6 5
1 2
1 3
2 4
2 5
5 6
3 3
1 2
2 3
1 3
```

Output

```
BRRRB
-1
```

H. Tomar Mobile E Games Ache?

After a long and hectic Eid day you are finally relaxing. But then your 5 year old cousin comes and wants to play games on your phone. Luckily you still have a nokia 1600. You open snake xenzia in the legendary phone, activate a cheat code to make the snake never grow, hand it over to the kid and watch as he plays.

If the snake moves out of screen from the top, it reappears from the bottom in the same column. If the snake moves out of screen from the left, it reappears from the right in the same row. Similar thing occurs for bottom and right. You notice that the snake takes 16 moves to repeat its position if allowed to move vertically. The snake takes 20 moves to repeat its position if allowed to move horizontally. A move can be defined as movement of the snake in one unit time, upward, downward, to the left or to the right. The goal of the snake is to eat a reward located somewhere on the screen.

Now you wonder, if the whole screen is regarded as a grid of size 16×20 of possible positions for the snake and also for the reward, what is the minimum number of moves required for a snake at cell (i_1, j_1) to reach a reward at cell (i_2, j_2).

Input

The first line contains a single integer $T (1 \leq T \leq 30)$— the number of test cases.

Each of the following T lines consist of 4 integers i_1, j_1, i_2, j_2, $(1 \leq i_1, i_2 \leq 16, 1 \leq j_1, j_2 \leq 20)$.

Output

Print the answer of each test case in a new line.

Samples

Input

```
5
1 1 16 20
4 20 16 1
9 9 9 9
1 15 16 15
10 10 15 15
```

Output

```
2
5
0
1
10
```

Input Output

Sample 1: Snake starts at $(1, 1)$. In one upward move, it goes to $(16, 1)$. Next, it goes to $(16, 20)$ with one leftward move.

Sample 2: It starts at $(4, 20)$. With one rightward move, it goes to $(4, 1)$. After that, with 4 consecutive upward moves, it reaches $(16, 1)$.

I. Colded LCM

Chris Martin is challenged with another problem. This time our favorite violinist Lindsey Stirling asked him to solve this problem. Lindsey will give Chris two integers L and R, such that $L < R$. Chris has to find two integers a and b, such that $L \le a < b \le R$, where LCM of a and b is as minimum as possible. Print this minimum value of LCM.

By definition, LCM or Least Common Multiple of two integers x and y is the minimum positive integer divisible by both x and y.

Input

First line of the input will be an integer T $(1 \le T \le 2000)$, denoting number of testcases. Each of the next T lines will contain two integers L and R $(1 \le L < R \le 10^9)$, denoting the numbers Lindsey gave to Chris in the corresponding testcase.

Output

For each testcase, print the answer in one line.

Samples

Input

```
2
1 3
4 6
```

Output

```
2
12
```

$i \to L$

$j = i+1$

J. Shine on You Crazy Diamond

This problem is authored by one of our late friends, Sifat, who unexpectedly left us last year. He was a brilliant programmer, an astute philosopher and an amazing human being. Had he been with us, he would have been delighted to see this year's brilliant IUPC coming to life and would wish you good luck for the contest. We sincerely hope you enjoy his problem.

You are given an array a of size n. You can select a number $x > 1$ (x may or may not be an element of a) and remove every multiple of x from the array. You need to find the minimum size of the array a after performing this operation exactly twice.

Input

The first line contains an integer, $n(1 \le n \le 10^5)$.

The second line contains n space-separated integers $a_1, a_2, \cdots, a_n(1 \le a_i \le 10^6)$.

Output

On the first line, output a single integer, the minimum size of array a after performing this operation at most twice.

On the second line, output two integers x_1 and x_2 $(2 \le x_1, x_2 \le 2^{31} - 1)$, the chosen values of x for the two operations.

Samples

Input

```
4
4 8 25 21
```

Output

```
1
4 21
```

You can chose 4 and remove 4 8 to get the new array 25 21. Then you can chose 25 to get the new array 21 . So the answer is 1.

There exists other choices that can give the same result too. For example 2 3 is a valid solution as well.

Input

```
3
2 4 6
```

Output

```
0
2 1605113
```

Input

Output

If you choose 2 first, all elements are removed after the first move. Then you can choose any integer for the second move. As such 2 x for any $2 \leq x \leq 2^{31} - 1$ is a valid output.