

# 2023 Canadian Computing Olympiad

## Day 1, Problem 1

### Binaria

**Time Limit: 1 second**

#### Problem Description

You have been hired by the Cheap Communication Organization (CCO) to work on a communication breakthrough: sub-message sum (SMS). This revolutionary idea works as follows.

Given a binary string of length  $N$ , and some positive integer  $K$  with  $K \leq N$ , the SMS for the string consists of a sequence of  $N - K + 1$  sums. The first sum in the sequence is the sum of digits 1 through  $K$ , the second sum is the sum of digits 2 through  $K + 1$ , and so on until the last sum which is the sum of digits  $N - K + 1$  through  $N$ .

For example, if  $K = 4$ , the SMS of the binary string 110010 is 2,2,1. This is because  $1 + 1 + 0 + 0 = 2$ ,  $1 + 0 + 0 + 1 = 2$ , and  $0 + 0 + 1 + 0 = 1$ .

Since you are a very junior developer, your job is not to find the original binary string from a given SMS, but rather the number of binary strings that could have formed this SMS.

#### Input Specification

The first line of input contains the two space-separated integers  $N$  and  $K$  where  $1 \leq K \leq N$ .

The second line of input contains  $N - K + 1$  space-separated integers which is the SMS of at least one binary string.

Marks Awarded	Bounds on $N$	Additional Bounds on $K$
3 marks	$1 \leq N \leq 10$	$K \leq 3$
3 marks	$1 \leq N \leq 10$	None
4 marks	$1 \leq N \leq 1\,000$	$K \leq 10$
4 marks	$1 \leq N \leq 10^6$	$K \leq 20$
4 marks	$1 \leq N \leq 10^6$	$K \leq 3\,000$
7 marks	$1 \leq N \leq 10^6$	None

#### Output Specification

Output the remainder of  $T$  divided by the prime number  $10^6 + 3$  where  $T$  is the positive integer equal to the total number of possible binary strings that correspond to the given SMS.

#### Sample Input

```
7 4
3 2 2 2
```

**Output for Sample Input**

3

**Explanation of Output for Sample Input**

The possible strings of length 7 are 1011001, 1101010, and 1110011.

2023 Canadian Computing Olympiad  
 Day 1, Problem 2  
**Real Mountains**

**Time Limit: 5 seconds**

**Problem Description**

Thanks to your help with cropping her picture, Rebecca’s scenic photo is now featured on the front cover of the newest issue of her magazine. However, it seems that some of her readers still aren’t pleased with the picture. In particular, they seem to believe that the mountain in the picture is fake!

For simplicity, we can describe the picture as a sequence of  $N$  columns of pixels. In the  $i$ -th column, the first  $h_i$  pixels from the bottom are of mountains. Her readers will only believe that the picture contains a real mountain if it contains a single (possibly wide) peak. That is, if there exists some index  $p$  with  $1 \leq p \leq N$  such that  $h_1 \leq h_2 \leq \dots \leq h_p \geq \dots \geq h_{N-1} \geq h_N$ .

Luckily, Rebecca can still pay her editors to modify the picture and reprint the magazine. Unfortunately for her though, the editors have a very peculiar pricing scheme for their work. The only way Rebecca can edit the picture is by sending emails to her editors containing the integers  $(i, j, k)$  such that  $1 \leq i < j < k \leq N$  and  $h_i > h_j < h_k$ . The editors will then add an extra pixel of mountains in the  $j$ -th column (i.e. increment  $h_j$  by 1) for a cost of  $h_i + h_j + h_k$  cents. Note that the change in  $h_j$  may affect the costs of future edits.

To please her readers, Rebecca would like to edit the picture so that they believe it contains a real mountain. Can you tell her the minimum cost required to do so?

**Input Specification**

The first line of input contains an integer  $N$ .

The second line of input contains  $N$  space-separated integers  $h_1, h_2, \dots, h_N$ .

Marks Awarded	Bounds on $N$	Bounds and constraints on $h_i$
3 marks	$3 \leq N \leq 5\,000$	$1 \leq h_i \leq 100$ ; $h_1 \geq h_2 \geq \dots \geq h_p \leq \dots \leq h_{N-1} \leq h_N$ for some $p, 1 \leq p \leq N$
3 marks	$3 \leq N \leq 5\,000$	$1 \leq h_i \leq 100$
3 marks	$3 \leq N \leq 5\,000$	$1 \leq h_i \leq 10^6$
3 marks	$3 \leq N \leq 5\,000$	$1 \leq h_i \leq 10^9$
4 marks	$3 \leq N \leq 10^6$	$1 \leq h_i \leq 100$
5 marks	$3 \leq N \leq 10^6$	$1 \leq h_i \leq 10^6$
4 marks	$3 \leq N \leq 10^6$	$1 \leq h_i \leq 10^9$

### Output Specification

Output the remainder of  $T$  divided by the prime number  $10^6 + 3$  where  $T$  is the minimum cost (in cents) that Rebecca would need to incur in order to please her readers.

### Sample Input

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

### Output for Sample Input

```
14
```

### Explanation of Output for Sample Input

Rebecca can send two emails, the first containing the integers  $(2, 6, 7)$  and the second containing the integers  $(1, 2, 5)$ . The first email costs 5 cents and increases  $h_6$  by 1, while the second email costs 9 cents and increases  $h_2$  by 1.

The  $h_i$  values in the final picture will be  $[3, 3, 4, 5, 4, 2, 2, 1]$ . Her readers will believe this final picture contains a real mountain.

2023 Canadian Computing Olympiad  
Day 1, Problem 3  
**Line Town**

**Time Limit: 2 seconds**

**Problem Description**

The  $N$  residents of Line Town have arranged themselves in a line. Initially, the residents have happiness values of  $h_1, h_2, \dots, h_N$  from left to right along the line.

Since you are the mayor of Line Town, you are implementing the third pillar of your plan entitled “Community, Candy, and Organization” (CCO). As such, you have taken the mayoral power to swap the resident’s locations. In one swap, you may tell two *adjacent* residents to swap their positions in the line. However, this swap will cause both residents to negate their happiness values.

You would like to perform some swaps so that the residents’ happiness values are in non-decreasing order from left to right in the line. Determine whether this is possible, and if so, the minimum number of swaps needed.

**Input Specification**

The first line of input contains a single integer  $N$ .

The next line of input contains  $N$  integers  $h_1, \dots, h_N$  ( $-10^9 \leq h_i \leq 10^9$ ), the happiness values of the residents from left to right.

Marks Awarded	Bounds on $N$	Bounds on $h_i$
3 marks	$1 \leq N \leq 2\,000$	$ h_i  = 1$ for all $i$
3 marks	$1 \leq N \leq 500\,000$	$ h_i  = 1$ for all $i$
3 marks	$1 \leq N \leq 2\,000$	$ h_i  \leq 1$ for all $i$
4 marks	$1 \leq N \leq 500\,000$	$ h_i  \leq 1$ for all $i$
4 marks	$1 \leq N \leq 2\,000$	$ h_i  \neq  h_j $ for all $i \neq j$
3 marks	$1 \leq N \leq 500\,000$	$ h_i  \neq  h_j $ for all $i \neq j$
2 marks	$1 \leq N \leq 2\,000$	No additional constraints.
3 marks	$1 \leq N \leq 500\,000$	No additional constraints.

**Output Specification**

On a single line, output the minimum number of swaps, or  $-1$  if the task is impossible.

**Sample Input 1**

6  
-2 7 -1 -8 2 8

### **Output for Sample Input 1**

3

### **Explanation of Output for Sample Input 1**

It is possible to perform 3 swaps as follows:

1. Swap the 2nd and 3rd resident so that the line becomes  $[-2, 1, -7, -8, 2, 8]$ .
2. Swap the 4th and 5th resident so that the line becomes  $[-2, 1, -7, -2, 8, 8]$ .
3. Swap the 3rd and 4th resident so that the line becomes  $[-2, 1, 2, 7, 8, 8]$ .

The residents are now arranged in non-decreasing order of happiness values as required. No non-decreasing arrangement can be obtained with less than 3 swaps.

### **Sample Input 2**

4

1 -1 1 -1

### **Output for Sample Input 2**

-1

### **Explanation of Output for Sample Input 2**

There is no sequence of swaps that will place residents in non-decreasing order of happiness values.

# 2023 Canadian Computing Olympiad

## Day 2, Problem 1

### Flip it and Stick it

**Time Limit: 1 second**

#### Problem Description

Finn is playing a game of “Flip it and Stick it” which is abbreviated as FiSi. FiSi is a one-player game played on two strings,  $S$  and  $T$ , of 0s and 1s. Finn is allowed to make moves of the following form:

- Select a substring of  $S$  and reverse it, gluing the pieces of the string back together in their original order to form the new string  $S$ .

For example, Finn may take the string  $S = 101100$ , take the substring  $011$  starting at index 2 (assuming 1-based string indexing), and create the string  $S = 111000$  in one move.

Finn wins the game if  $S$  does **not** contain  $T$  as a substring. Your task is to help Finn determine the length of the shortest winning sequence of moves or tell him that the game cannot be won.

#### Input Specification

The first line of input contains the string  $S$  ( $1 \leq |S| \leq 200\,000$ ).

The second line of input contains the string  $T$  ( $1 \leq |T| \leq 3$ ).

In the table below,  $T_1$  is the first bit in  $T$ ,  $T_2$  is the second bit in  $T$ , and  $T_3$  is the third bit in  $T$ , when reading from left-to-right.

Marks Awarded	Bounds on $T$
1 mark	$ T  = 1$
3 marks	$ T  = 2, T_1 \neq T_2$
4 marks	$ T  = 2$
5 marks	$ T  = 3, T_1 \neq T_3$
5 marks	$ T  = 3, T_1 \neq T_2$
7 marks	$ T  = 3$

#### Output Specification

Output the minimum number of moves needed or  $-1$  if it is impossible to win the game.

#### Sample Input 1

100110

10

### **Output for Sample Input 1**

2

### **Explanation of Output for Sample Input 1**

Finn starts with the string 100110. He cannot avoid 10 as a substring in one move, but he can in two moves.

For example, his first move could be to reverse the substring from index 4 to index 6 (110) to get 100011. Then, his second move can be to reverse the substring from index 1 to index 4 (1000) to get 000111, which does not have 10 as a substring.

### **Sample Input 2**

000

00

### **Output for Sample Input 2**

-1

### **Explanation of Output for Sample Input 2**

No matter how many moves Finn makes, the string  $S$  will always contain  $T$  as a substring.

# 2023 Canadian Computing Olympiad

## Day 2, Problem 2

### Travelling Trader

**Time Limit: 2 seconds**

#### Problem Description

A trader would like to make a business of travelling between cities, moving goods from one city to another in exchange for a profit. There are  $N$  cities labelled  $1, \dots, N$  and  $N - 1$  roads. Each road joins two cities and takes one day to traverse. It is possible to reach any city from any other city using these roads.

The  $i$ -th city can give a profit of  $p_i$  if the trader is currently in that city and chooses to do business in that city, but this profit may only be obtained once. The trader starts by doing business in city 1 and wants to travel along the roads, visiting cities to maximize their total profit. However, the trader's boss will get unhappy and lay off the trader as soon as the trader goes more than  $K$  days in a row without increasing their total profit. Note that the trader will take only one day to move between adjacent cities, regardless of whether the trader does business in either city. We would like to know the maximum profit the trader can make under this condition and a route that obtains this profit.

#### Input Specification

The first line of input contains two space-separated integers  $N$  and  $K$ .

The next  $N - 1$  lines of input each contain two space-separated integers  $u_i$  and  $v_i$  ( $1 \leq u_i, v_i \leq N, u_i \neq v_i$ ), describing a road.

The last line of input contains  $N$  integers  $p_1, \dots, p_N$  ( $1 \leq p_i \leq 10^9$ ), the profits given by choosing to do business in the corresponding city.

Marks Awarded	Bounds on $N$	Bounds on $K$
2 marks	$2 \leq N \leq 200\,000$	$K = 1$
7 marks	$2 \leq N \leq 200$	$K = 2$
3 marks	$2 \leq N \leq 2\,000$	$K = 2$
4 marks	$2 \leq N \leq 200\,000$	$K = 2$
4 marks	$2 \leq N \leq 2\,000$	$K = 3$
5 marks	$2 \leq N \leq 200\,000$	$K = 3$

#### Output Specification

On the first line, output the maximum possible total profit.

On the second line, output  $M$  ( $1 \leq M \leq N$ ), the number of cities the trader does business in on an optimal route.

On the third line, output  $M$  space-separated integers  $x_1, \dots, x_M$ , the cities the trader does business in on an optimal route in order, starting with  $x_1 = 1$ .

If there are multiple possible correct outputs, any correct output will be accepted.

### Sample Input 1

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

### Output for Sample Input 1

```
7
2
1 3
```

### Explanation of Output for Sample Input 1

On day 1, the trader starts by doing business in city 1, making a profit of 3.

On day 2, the trader moves to city 3 and does business there, making a profit of 4.

At this point, the trader cannot reach another city in which they have not done business before getting laid off, so their total profit is 7.

### Sample Input 2

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

### Output for Sample Input 2

```
14
5
1 4 5 2 3
```

### Explanation of Output for Sample Input 2

The trader can make a profit in every city by visiting them in the order 1, 2, 4, 2, 5, 2, 1, 3.

Note that the trader strategically delays doing business in city 2 to ensure they do not go more than 2 days without making a profit.

2023 Canadian Computing Olympiad  
Day 2, Problem 3  
**Triangle Collection**

**Time Limit: 4 seconds**

**Problem Description**

Alice has a collection of sticks. Initially, she has  $c_\ell$  sticks of length  $\ell$  for each  $\ell = 1, \dots, N$ .

Alice would like to use her sticks to make some isosceles triangles. An isosceles triangle is made of two sticks of the same length, say  $\ell$ , and a third stick with a length between 1 and  $2\ell - 1$  inclusive. Note that the triangles must strictly obey the triangle inequality, and equilateral triangles are okay. Each stick may be used in at most one triangle. Alice would like to know the maximum number of isosceles triangles she can make with her sticks.

There are  $Q$  events that change the collection of sticks she has. The  $i$ -th event consists of two integers  $\ell_i$  and  $d_i$ , representing that the number of sticks of length  $\ell_i$  changes by  $d_i$ . Note that  $d_i$  may be positive, negative, or even 0, but Alice will never have a negative number or more than  $10^9$  sticks of each length.

Your task is to determine the maximum number of isosceles triangles Alice can make after each event if she uses her sticks optimally.

**Input Specification**

The first line of input contains two space-separated integers  $N$  and  $Q$ .

The second line of input contains  $N$  space-separated integers  $c_1, c_2, \dots, c_N$  ( $0 \leq c_i \leq 10^9$ ), representing Alice's initial collection.

The next  $Q$  lines of input each contain two space-separated integers  $\ell_i$  and  $d_i$  ( $1 \leq \ell_i \leq N$ ,  $-10^9 \leq d_i \leq 10^9$ ), representing an event.

Initially and after each event, the number of sticks of length  $\ell$  is between 0 and  $10^9$  for all  $\ell = 1, \dots, N$ .

Marks Awarded	Bounds on $N, Q$	Additional Constraints
5 marks	$1 \leq N, Q \leq 2\,000$	There are at most 2000 sticks in total initially and after each event.
5 marks	$1 \leq N, Q \leq 2\,000$	No additional constraints.
5 marks	$1 \leq N, Q \leq 200\,000$	The number of sticks of each length is either 0, 1, or 2 initially and after each event.
5 marks	$1 \leq N, Q \leq 200\,000$	For each event, $ d_i  = 1$ .
5 marks	$1 \leq N, Q \leq 200\,000$	No additional constraints.

### Output Specification

Output  $Q$  lines each containing a single integer, the answer after each event.

### Sample Input

```
4 3
3 1 4 1
3 -3
1 6
2 1
```

### Output for Sample Input

```
1
3
4
```

### Explanation of Output for Sample Input

After the first event, Alice can make a single triangle with sticks of lengths  $(1, 1, 1)$ .

After the second event, Alice can make 3 triangles with sticks of lengths  $(1, 1, 1)$ .

After the third event, Alice can make 3 triangles with sticks of lengths  $(1, 1, 1)$  and a triangle with sticks of lengths  $(2, 2, 3)$ .