Educational Codeforces Round 6

A. Professor GukiZ's Robot

0.5 seconds, 256 megabytes

Professor GukiZ makes a new robot. The robot are in the point with coordinates (x_1, y_1) and should go to the point (x_2, y_2). In a single step the robot can change any of its coordinates (maybe both of them) by one (decrease or increase). So the robot can move in one of the 8 directions. Find the minimal number of steps the robot should make to get the finish position.

Input

The first line contains two integers x_1, y_1 ($-10^9 \le x_1, y_1 \le 10^9$) — the start position of the robot.

The second line contains two integers x_2, y_2 ($-10^9 \le x_2, y_2 \le 10^9$) — the finish position of the robot.

Output

Print the only integer d — the minimal number of steps to get the finish position.

input
0 0 4 5
output
5

input
3 4 6 1
output
3

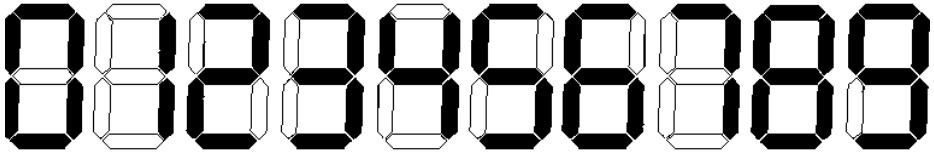
In the first example robot should increase both of its coordinates by one four times, so it will be in position $(4, 4)$. After that robot should simply increase its y coordinate and get the finish position.

In the second example robot should simultaneously increase x coordinate and decrease y coordinate by one three times.

B. Grandfather Dovlet's calculator

1 second, 256 megabytes

Once Max found an electronic calculator from his grandfather Dovlet's chest. He noticed that the numbers were written with seven-segment indicators (https://en.wikipedia.org/wiki/Seven-segment_display).



Max starts to type all the values from a to b. After typing each number Max resets the calculator. Find the total number of segments printed on the calculator.

For example if $a = 1$ and $b = 3$ then at first the calculator will print 2 segments, then — 5 segments and at last it will print 5 segments. So the total number of printed segments is 12.

Input

The only line contains two integers a, b ($1 \le a \le b \le 10^6$) — the first and the last number typed by Max.

Output

Print the only integer a — the total number of printed segments.

input
1 3
output
12

input
10 15

output
39

C. Pearls in a Row

2 seconds, 256 megabytes

There are n pearls in a row. Let's enumerate them with integers from 1 to n from the left to the right. The pearl number i has the type a_i.

Let's call a sequence of consecutive pearls a `segment`. Let's call a segment `good` if it contains two pearls of the same type.

Split the row of the pearls to the maximal number of good segments. Note that each pearl should appear in exactly one segment of the partition.

As input/output can reach huge size it is recommended to use fast input/output methods: for example, prefer to use `scanf/printf` instead of `cin/cout` in C++, prefer to use `BufferedReader/PrintWriter` instead of `Scanner/System.out` in `Java`.

Input

The first line contains integer n ($1 \le n \le 3 \cdot 10^5$) — the number of pearls in a row.

The second line contains n integers a_i ($1 \le a_i \le 10^9$) – the type of the i-th pearl.

Output

On the first line print integer k — the maximal number of segments in a partition of the row.

Each of the next k lines should contain two integers l_j, r_j ($1 \le l_j \le r_j \le n$) — the number of the leftmost and the rightmost pearls in the j-th segment.

Note you should print the correct partition of the row of the pearls, so each pearl should be in exactly one segment and all segments should contain two pearls of the same type.

If there are several optimal solutions print any of them. You can print the segments in any order.

If there are no correct partitions of the row print the number "-1".

input
5 1 2 3 4 1
output
1 1 5

input
5 1 2 3 4 5
output
-1

input
7 1 2 1 3 1 2 1
output
2 1 3 4 7

D. Professor GukiZ and Two Arrays

3 seconds, 256 megabytes

Professor GukiZ has two arrays of integers, `a` and `b`. Professor wants to make the sum of the elements in the array `a` s_a as close as possible to the sum of the elements in the array `b` s_b. So he wants to minimize the value $v = |s_a - s_b|$.

In one operation professor can swap some element from the array `a` and some element from the array `b`. For example if the array `a` is $[5, 1, 3, 2, 4]$ and the array `b` is $[3, 3, 2]$ professor can swap the element 5 from the array `a` and the element 2 from the array `b` and get the new array `a` $[2, 1, 3, 2, 4]$ and the new array `b` $[3, 3, 5]$.

Professor doesn't want to make more than two swaps. Find the minimal value v and some sequence of no more than two swaps that will lead to the such value v. Professor makes swaps one by one, each new swap he makes with the new arrays `a` and `b`.

Input
The first line contains integer n ($1 \le n \le 2000$) — the number of elements in the array a.

The second line contains n integers a_i ($-10^9 \le a_i \le 10^9$) — the elements of the array a.

The third line contains integer m ($1 \le m \le 2000$) — the number of elements in the array b.

The fourth line contains m integers b_j ($-10^9 \le b_j \le 10^9$) — the elements of the array b.

Output
In the first line print the minimal value $v = |s_a - s_b|$ that can be got with no more than two swaps.

The second line should contain the number of swaps k ($0 \le k \le 2$).

Each of the next k lines should contain two integers x_p, y_p ($1 \le x_p \le n$, $1 \le y_p \le m$) — the index of the element in the array a and the index of the element in the array b in the p-th swap.

If there are several optimal solutions print any of them. Print the swaps in order the professor did them.

input
```
5
5 4 3 2 1
4
1 1 1 1
```
output
```
1
2
1 1
4 2
```

input
```
5
1 2 3 4 5
1
15
```
output
```
0
0
```

input
```
5
1 2 3 4 5
4
1 2 3 4
```
output
```
1
1
3 1
```

E. New Year Tree

3 seconds, 256 megabytes

The New Year holidays are over, but Resha doesn't want to throw away the New Year tree. He invited his best friends Kerim and Gural to help him to redecorate the New Year tree.

The New Year tree is an undirected tree with n vertices and root in the vertex 1.

You should process the queries of the two types:

1. Change the colours of all vertices in the subtree of the vertex v to the colour c.
2. Find the number of different colours in the subtree of the vertex v.

Input
The first line contains two integers n, m ($1 \le n, m \le 4 \cdot 10^5$) — the number of vertices in the tree and the number of the queries.

The second line contains n integers c_i ($1 \le c_i \le 60$) — the colour of the i-th vertex.

Each of the next $n - 1$ lines contains two integers x_j, y_j ($1 \le x_j, y_j \le n$) — the vertices of the j-th edge. It is guaranteed that you are given correct undirected tree.

The last m lines contains the description of the queries. Each description starts with the integer t_k ($1 \le t_k \le 2$) — the type of the k-th query. For the queries of the first type then follows two integers v_k, c_k ($1 \le v_k \le n, 1 \le c_k \le 60$) — the number of the vertex whose subtree will be recoloured with the colour c_k. For the queries of the second type then follows integer v_k ($1 \le v_k \le n$) — the number of the vertex for which subtree you should find the number of different colours.

Output

For each query of the second type print the integer a — the number of different colours in the subtree of the vertex given in the query.

Each of the numbers should be printed on a separate line in order of query appearing in the input.

input

```
7 10
1 1 1 1 1 1
1 2
1 3
1 4
3 5
3 6
3 7
1 3 2
2 1
1 4 3
2 1
1 2 5
2 1
1 6 4
2 1
2 2
2 3
```

output

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

input

```
23 30
1 2 2 6 5 3 2 1 1 1 2 4 5 3 4 4 3 3 3 3 3 4 6
1 2
1 3
1 4
2 5
2 6
3 7
3 8
4 9
4 10
4 11
6 12
6 13
7 14
7 15
7 16
8 17
8 18
10 19
10 20
10 21
11 22
11 23
2 1
2 5
2 6
2 7
2 8
2 9
2 10
2 11
2 4
1 12 1
1 13 1
1 14 1
1 15 1
1 16 1
1 17 1
1 18 1
1 19 1
1 20 1
1 21 1
1 22 1
1 23 1
2 1
2 5
2 6
2 7
2 8
```

```
2 9
2 10
2 11
2 4
```

output

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

F. Xors on Segments

10 seconds, 512 megabytes

You are given an array with n integers a_i and m queries. Each query is described by two integers (l_j, r_j).

Let's define the function $f(u, v) = u \oplus (u+1) \oplus \ldots \oplus v$. The function is defined for only $u \le v$.

For each query print the maximal value of the function $f(a_x, a_y)$ over all $l_j \le x, y \le r_j,\ a_x \le a_y$.

Input

The first line contains two integers n, m $(1 \le n \le 5 \cdot 10^4,\ 1 \le m \le 5 \cdot 10^3)$ — the size of the array and the number of the queries.

The second line contains n integers a_i $(1 \le a_i \le 10^6)$ — the elements of the array a.

Each of the next m lines contains two integers l_j, r_j $(1 \le l_j \le r_j \le n)$ – the parameters of the j-th query.

Output

For each query print the value a_j on a separate line — the maximal value of the function $f(a_x, a_y)$ over all $l_j \le x, y \le r_j,\ a_x \le a_y$.

input

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

output

```
7
7
7
```

input

```
1 1
1
1 1
```

output

```
1
```

input

```
6 20
10 21312 2314 214 1 322
1 1
1 2
1 3
1 4
1 5
1 6
2 2
2 3
2 4
2 5
2 6
3 4
3 5
3 6
4 4
4 5
4 6
5 5
5 6
6 6
```

output
10 21313 21313 21313 21313 21313 21312 21313 21313 21313 21313 2314 2315 2315 214 215 323 1 323 322