

# mBIT Standard Division Problems

June 7, 2020

These problems are roughly ordered by difficulty. However, we suggest you look through and think about as many problems as you can in the time provided. Good luck and happy coding!

## Contents

<b>1</b>	<b>Ice Cream Truck</b>	<b>3</b>
<b>2</b>	<b>Pirating Parrots</b>	<b>4</b>
<b>3</b>	<b>Monkey Signs</b>	<b>5</b>
<b>4</b>	<b>Happy Bunnies</b>	<b>6</b>
<b>5</b>	<b>Snake Moves</b>	<b>7</b>
<b>6</b>	<b>Egg Interception</b>	<b>8</b>
<b>7</b>	<b>Musical Bowings</b>	<b>9</b>
<b>8</b>	<b>Icebergs</b>	<b>10</b>
<b>9</b>	<b>Bracelets</b>	<b>11</b>
<b>10</b>	<b>Zoo Tour</b>	<b>12</b>
<b>11</b>	<b>Raging Rhinos</b>	<b>13</b>
<b>12</b>	<b>Hen Hackers</b>	<b>14</b>

---

Thanks to Evan Chen for letting us use his style file

The memory limit for all problems is 256 MB.

Time Limits (seconds)			
Problem	C++	Java	Python
Ice Cream Truck	1	1	1
Monkey Signs	1	1	1
Pirating Parrots	1	1	1
Happy Bunnies	1	1	1
Snake Moves	1	1	1
Egg Interception	1	1	1
Musical Bowings	1	1	1
Icebergs	1	1	2
Bracelets	1	1	1
Zoo Tour	1	2	2
Raging Rhinos	1	2	2
Hen Hackers	1	1	1

### Advice:

**Go for partial credit!** Even if you don't know how to solve a problem with the full bounds, you can still earn some of the points by coding a solution that works for smaller test cases. **On each problem, we guarantee that 50% of our test cases have reduced input sizes.** This means that suboptimal solutions can earn partial credit.

**Watch out for integer overflow.** When a problem uses large values, make sure you use `long` (in Java) or `long long` (in C++).

**Consider using fast I/O.** For problems with large input sizes, you may want to use faster I/O methods. We have increased our time limits so that fast I/O should not be necessary, but it doesn't hurt to be safe. Here is how to use fast I/O in each language:

- In Python, write `from sys import stdin, stdout` at the top of your program. When reading input, use `stdin.readline()`. To write output, use `stdout.write()`.
- In Java, use a custom Scanner class as shown [here](#).
- In C++, write `ios_base::sync_with_stdio(0); cin.tie(NULL); cout.tie(NULL);` at the top of your main method. Then you can use `cin` and `cout` as usual.

**Special considerations for Python.** In Python, avoid multidimensional lists. Nesting lists may slow your program down. You can use a trick known as [array flattening](#) instead. Additionally, if you are coding in Python, consider submitting your solution in PyPy. It behaves in the same way as Python, but is often faster (the time limits for PyPy are the same as Python). Putting your code in a `main` method can speed up run time as well. Finally, **do not** use `exit()`.

**Ask for clarifications!** If you are confused about a problem statement, do not hesitate to message us.

## §1 Ice Cream Truck



*The world record for eating ice cream is 16 pints in six minutes!*

Natsuo, Rui, and Hina are visiting the zoo on a hot summer day and are up for some ice cream. The ice cream truck at the zoo sells either a single ice cream cone for  $A$  dollars, or a bundle of five ice cream cones for  $B$  dollars. If the trio wants to buy **exactly**  $N$  cones, what is the minimum amount of dollars necessary?

### **Input Format:**

The only line contains  $A$ ,  $B$  and  $N$  ( $1 \leq A, B, N \leq 100$ ).

### **Output Format:**

One line with the minimum dollars necessary.

### **Sample Input:**

1 6 19

### **Sample Output:**

19

## §2 Pirating Parrots



*Most parrot eat seeds. Others may eat fruit, nectar, flowers or small insects.*

The parrots of mBIT zoo are planning a covert operation to steal all of the bird seeds! The zoo can be represented as a coordinate plane with the parrot habitat at point  $(0, 0)$  and a crate of seeds at  $(x, y)$ . To reach the crate, the parrots plan to follow a string composed of the commands R, L, U, and D which tell the parrots to go right, left, up, and down respectively. However, the string is not yet complete! Help the parrots find the minimum number of letters that need to be added to the end of the string so that they will be able to make it to the crate of treats from their habitat.

### Input Format:

The first line contains  $N$ , the length of the string so far ( $1 \leq N \leq 500$ ).

The second line contains the string.

The third line contains the coordinates  $(X, Y)$  of the seeds ( $-100 \leq X, Y \leq 100$ ).

### Output Format:

Output the minimum number of commands that must be added.

### Sample Input:

```
6
UUDLRR
5 3
```

### Sample Output:

```
6
```

After the given sequence of moves, the parrots would end up at  $(1, 1)$ . The seeds are at  $(5, 3)$ , so at least 6 extra moves must be added to fix the command.

### §3 Monkey Signs



*There are currently 264 known species of monkeys.*

The monkeys are planning some mischief: they want to change some of the letters on a sign at the zoo. Since their favorite letter is ‘M’, they want to change the sign so that it has as many ‘M’s as possible. To accomplish this mischief, the monkeys plan to choose some letter of the alphabet, then replace all occurrences of that letter in the sign with an ‘M’. What is the maximum number of ‘M’s that the sign could have after the monkeys are done with it? The monkeys **do not care about capitalization**, so lowercase and uppercase ‘M’s are both counted in the total. Spaces do not count as a valid letter to replace.

#### **Input Format:**

The text of the original sign will be on one line. The length of the sign will no more than 1000, and it will only contain letters and spaces.

#### **Output Format:**

Output the maximum number of ‘M’s the sign could have after the monkeys change it.

#### **Sample Input:**

Mammal Atrium

#### **Sample Output:**

7

If the monkeys replace all ‘A’s, the sign would read ‘Mmmml Mtrium’.

## §4 Happy Bunnies



*Rabbits are one of the fastest reproducing animal species.*

Sagiri is checking up on the happiness of the bunnies in the zoo. There are  $B - A + 1$  bunnies with consecutive happiness values from  $A$  to  $B$ , inclusive. She now wants to compute the total happiness of the bunnies in the following way: For each bunny with no 7 in any digit of its happiness value, Sagiri will add its happiness value and the sum of the digits of its happiness value to the total. What is the total happiness?

### **Input Format:**

The only line contains  $A$  and  $B$ , which are defined in above. ( $1 \leq A \leq B \leq 10^5$ )

### **Output Format:**

One line with the total happiness.

### **Sample Input:**

6 11

### **Sample Output:**

70

The valid values of happiness are 6, 8, 9, 10, 11 (we exclude 7). The total happiness is  $(6 + 8 + 9 + 10 + 11) + (6 + 8 + 9 + 1 + 0 + 1 + 1) = 70$ .

## §5 Snake Moves



*Snakes can travel on smooth surfaces by making waves in their bodies and using a rolling motion.*

Queen Sarah the Snake is at the coordinates  $(x_0, y_0)$  on a  $W \times H$  grid, where the bottom left cell is  $(1, 1)$ . She wants to eat a series of  $N$  eggs at the coordinates  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$  in that order. However, like a queen in chess (and because she is a snake), Sarah likes to travel in only straight lines that are either horizontal, vertical, or diagonal. She has asked you to calculate the minimum number of such moves she needs to eat all of the eggs. She cannot eat an egg in the middle of a move, and she only occupies one cell at a time.

### Input Format:

The first line contains  $W$  and  $H$  ( $1 \leq W, H \leq 10^6$ ).

The second line contains  $x_0$  and  $y_0$  ( $1 \leq x_0 \leq W, 1 \leq y_0 \leq H$ ).

The third line contains  $N$  ( $1 \leq N \leq 10^4$ ), the number of eggs to travel to.

Among the next  $N$  lines, the  $i$ th line contains  $x_i$  and  $y_i$  ( $1 \leq x_i \leq W, 1 \leq y_i \leq H$ ), the coordinates for the  $i$ th egg.

It is guaranteed that no consecutive pairs of coordinates will be identical.

### Output Format:

Print the minimum number of moves Sarah must use.

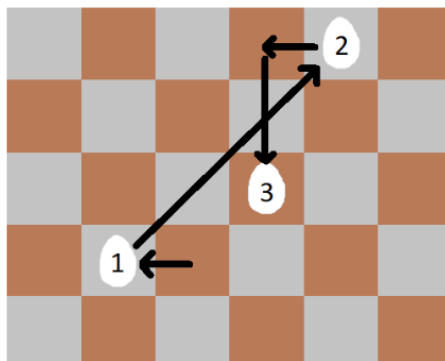
### Sample Input:

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

### Sample Output:

```
4
```

Sarah starts at  $(3, 2)$ . One optimal path she could follow (egg coordinates in bold) is  $(3, 2) \rightarrow (2, 2) \rightarrow (5, 5) \rightarrow (4, 5) \rightarrow (4, 3)$ . This path uses 4 moves.



## §6 Egg Interception



*There are 25 billion chickens on earth!*

Farmer Fred is raising  $N$  chickens and wants to collect eggs from them. However, to make more room for other animals, he has placed the chickens very high up in his barn. When they lay eggs, the eggs fall from the top of his barn (they all fall at the same, constant speed) and he has to catch them. The  $i$ th chicken ( $1 \leq i \leq N$ ) is at the  $x$ -coordinate  $i$ . Each chicken is at the same  $y$ -coordinate. Fred starts at the  $x$ -coordinate  $K$  and can run along the  $x$ -axis. He knows that the  $i$ th chicken will lay an egg at a *distinct* time  $T_i$ . Assuming that Fred can run infinitely fast, compute the minimum distance he must run to catch all the eggs before they hit the ground.

### Input Format:

The first line contains integers  $N$  ( $1 \leq N \leq 10^5$ ) and  $K$  ( $1 \leq K \leq N$ ).

$N$  integers follow, where the  $i$ th integer is  $T_i$  ( $1 \leq T_i \leq 10^9$ ), the time at which the  $i$ th chicken lays its egg. All values of  $T_i$  will be distinct.

### Output Format:

Print the minimum distance Fred needs to run.

### Sample Input:

```
6 4
4 2 1 6 10 3
```

### Sample Output:

```
15
```

Fred starts at  $x$ -coordinate 4. He runs to the  $x$ -coordinates  $(4) \rightarrow 3 \rightarrow 2 \rightarrow 6 \rightarrow 1 \rightarrow 4 \rightarrow 5$  for a total distance of 15.



## §7 Musical Bowings



Apparently horses really like *violin playing*.

Momo is the conductor of the mBIT zoo orchestra, the first ever orchestra to consist of solely animals. If you're familiar with string instruments like the violin, you may know that they are played by pushing and pulling a stick with horsehair (known as a "bow") across the strings to produce a sound. There are two types of bowing strokes: a "down bow" is a pulling motion, and an "up bow" is a pushing motion (check [this](#) video if you're still confused).

Typically, notes on music sheets are marked with symbols denoting up or down bow to instruct the player how they should play that note. However, because some of the members of the orchestra are quite literally monkeys, they have absolutely no clue what bowings to do.

Momo has decided to take matters into her own hands. She will look at the monkeys' music sheets and mark in all unmarked notes with a certain direction (up or down). However, she wants to avoid "hooked bowings," which are two consecutive notes with the same direction marked, as such a technique is too advanced for the monkeys to pull off reliably. Help her determine a set of bowings that minimizes the number of hooked bowings!

### Input Format:

The first line contains  $N$ , the number of notes in the piece ( $1 \leq N \leq 1000$ ).

The second line contains  $N$  characters representing the bowing marking on the note: either 'D' for down, 'U' for up, or 'B' for blank/no marking.

### Output Format:

One line with  $N$  space-separated characters denoting the marking of the notes. You cannot change 'U' or 'D' markings that were originally present in the input, and you cannot have any 'B's in your output. If there are multiple solutions with the fewest number of hooked bowings, print any of them.

### Sample Input:

```
7
B D B B B U U
```

### Sample Output:

```
U D D U D U U
```

The given answer contains 2 hooked bowings, which is the minimal possible. Note that U D U D U U U is also an acceptable answer.

## §8 Icebergs



*Some penguins can swim up to 22 miles per hour!*

The penguins have gotten into a fight over their fish! Initially,  $N$  penguins numbered 1 through  $N$  are each standing on their own iceberg at a distinct location on the coordinate plane. Once the fight starts, the penguins take turns throwing rocks at each other until only one remains. On a penguin's turn, it hurls a rock at the nearest other penguin (if there is a tie, it picks the lower indexed penguin). When an iceberg is hit by a rock, it immediately sinks and the penguin standing on it is eliminated from the fight. The penguins take turns in increasing order, starting from penguin 1 and wrapping around after penguin  $N$  (so the order is  $1, 2, 3, \dots, N - 1, N, 1, 2, \dots$ ). If a penguin is already eliminated, its turn is skipped. Find the last penguin remaining.

### Input Format:

The first line contains  $N$  ( $1 \leq N \leq 10^3$ ), the initial number of penguins.

In the next  $N$  lines, the  $i$ th line consists of integers  $x_i$  and  $y_i$  ( $-10^9 \leq x_i, y_i \leq 10^9$ ), the coordinates of the  $i$ th penguin's iceberg.

### Output Format:

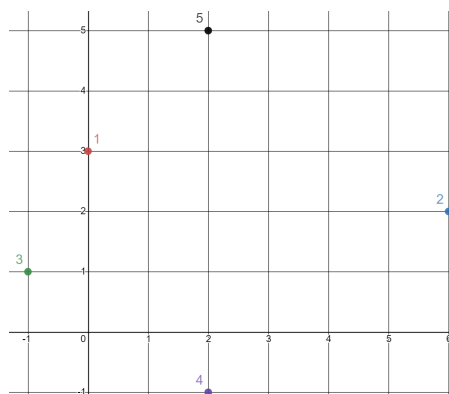
Print the index assigned to the last remaining penguin.

### Sample Input:

```
5
0 3
4 2
-1 1
2 -1
2 5
```

### Sample Output:

```
2
```



Penguin 1 has the first turn; it sinks Penguin 3. Penguin 2's turn is next; the closest penguins are Penguins 4 and 5, so he sinks Penguin 4 since 4 is the lower index. The next remaining penguin is Penguin 5, who sinks Penguin 1. Finally, Penguin 2 has another turn and sinks Penguin 5. Penguin 2 is the last one remaining.

## §9 Bracelets



*Gorillas are highly intelligent, and one even mastered sign language!*

Two gorillas are each wearing a circular bracelet with  $N$  numbered beads. Each integer from 1 through  $N$  appears **exactly once** on each bracelet. The gorillas want to rotate the bracelets to maximize the number of matching beads at a fixed orientation. Here, “matching” means that the bracelets have the same value at the same place. Compute this maximum number of matching values. (Two bracelets at a fixed orientation can be represented as sequences  $S$  and  $T$  of length  $N$ ; they have  $k$  matching numbers if there are exactly  $k$  indices  $i$  for which  $S_i = T_i$ ).

### Input Format:

The first line contains  $N$  ( $1 \leq N \leq 10^5$ ), the number of integers on each bracelet.

The second line contains the initial order of the numbers on the first bracelet, separated by spaces.

The third line contains the initial order of the numbers on the second bracelet, separated by spaces.

### Output Format:

Print the maximum number of matching values that can be achieved by rotating the bracelets.

### Sample Input:

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

### Sample Output:

```
4
```

Of the 6 rotations of the first bracelet (1 6 4 3 5 2, 2 1 6 4 3 5, 5 2 1 6 4 3, ...), the one that has the most matching values with the second bracelet is 3 5 2 1 6 4:

<b>3</b>	<b>5</b>	<b>2</b>	<b>1</b>	<b>6</b>	<b>4</b>
<b>3</b>	<b>5</b>	<b>6</b>	<b>1</b>	<b>2</b>	<b>4</b>

There are 4 matching pairs.

## §10 Zoo Tour



An estimated 600 million people visit zoos each year.

Lev is a new tour guide at the zoo, and he busy showing Dr. Roman around. The zoo habitats are arranged in a loop and are labeled from 1 to  $N$ . For any  $i$  from 1 to  $N - 1$ , there is a path of length  $A_i$  connecting habitats  $i$  and  $i + 1$ . There is also a path of length  $A_N$  connecting habitats 1 and  $N$ . Since Dr. Roman is a veterinarian who may need to get from one habitat to another at a moment's notice, he is interested in the shortest distance between certain pairs of habitats. Dr. Roman will ask Lev  $Q$  questions, where each question consists of two habitat numbers. For each question, help Lev out by determining the shortest distance between the two habitats. All paths can be traversed in both directions. Dr. Roman must travel along the paths.

### Input Format:

The first line contains  $N$  ( $1 \leq N \leq 10^5$ ) and  $Q$  ( $1 \leq Q \leq 10^5$ ).

The second line contains  $N$  integers:  $A_1, A_2, \dots, A_N$  ( $1 \leq A_i \leq 10^9$ ).

The next  $Q$  lines each contain a pair of integers,  $u$  and  $v$  ( $u \neq v$ ;  $1 \leq u, v \leq N$ ).

### Output Format:

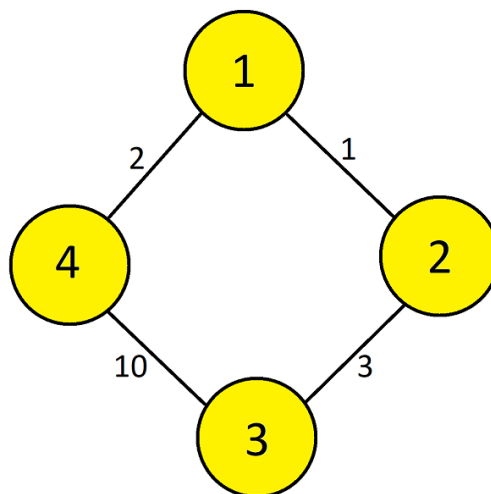
Print the shortest distance between each pair of habitats.

### Sample Input:

```
4 3
1 3 10 2
1 3
4 2
3 4
```

### Sample Output:

```
4
3
6
```



## §11 Raging Rhinos



*Fully grown rhinos can weigh over 5000 pounds!*

The rhinos are in a fit of rage, and they are ready to duke it out in an all-out frenzy. The rules are as follows: the rhinos all stand in a line facing either to the left or right. All rhinos begin with a positive amount of stamina. At the sound of the bell, all rhinos will begin charging in their respective directions at the same speed, and upon encountering another rhino, they will collide. The rhino with less stamina is knocked out of the line while the rhino with more stamina continues in the same direction, *with its stamina reduced by the stamina of its opponent*. If two rhinos with the same stamina collide, both drop out of the line. Can you determine the state of the rhinos after all collisions? Note that a rhino never changes directions.

### Input Format:

The first line contains  $N$ , the number of rhinos ( $1 \leq N \leq 10^5$ ).

Among the next  $N$  lines, the  $i$ th line describes the  $i$ th rhino from the left. Each line contains two integers  $s_i$  and  $d_i$  denoting the initial stamina and direction of each rhino ( $1 \leq s_i \leq 10^9$ ). If  $d_i = 0$ , then the rhino is facing left, otherwise if  $d_i = 1$ , it is facing right.

### Output Format:

Print one line containing  $M$ , the number of rhinos still running in on the line.

On the next  $M$  lines, print two space-separated integers containing the remaining stamina and direction of each rhino. Describe the rhinos in the order they appear on the line (from left to right).

### Sample Input:

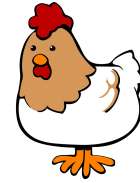
```
4
39 1
20 0
21 0
12 0
```

### Sample Output:

```
2
2 0
12 0
```

The first and second rhino collide, and the first rhino is victorious with 19 remaining stamina. Then the first and third rhino collide, and the third rhino is victorious with 2 remaining stamina. The third and fourth rhino never touch because they are moving in the same direction.

## §12 Hen Hackers



*If chickens listen to classical music, they can lay bigger and heavier eggs.*

This problem involves interaction between your program and the grader.

In an attempt to “liberate” their fellow birds, a group of highly intelligent hens are attempting to compromise the zoo’s security system. To get access to the zoo’s mainframe, the hens first need to guess the administrative password. The hens know that the password is a string built from up to 62 possible characters: a, b, c, ..., z, A, B, C, ..., Z, 0, 1, ... 9. They also know that **no character appears more than once** in the password, which means that it has a maximum length of 62 (and a minimum length of 1). The password is case-sensitive.

We define a guess be *close* if and only if the real password can be obtained by inserting additional characters among the letters and numbers of the original guess. In other words, a guess is close if it is a (not necessarily contiguous) subsequence of the password. Every time the hens guess a password, the zoo’s security system will do one of three things:

- If the guess is exactly equal to the password, it will print ‘C’ and grant the hens access to the mainframe.
- If the guess *close* but not exactly equal to the password, it will print ‘Y’.
- If the guess is neither *close* nor exactly equal to the password, it will print ‘N’.

The mainframe permits at most 750 guesses before it locks further attempts and alerts the zoo officials of suspicious activity. The hens have hired you to guess the password before you use all of your allotted attempts. Can you help them?

### I/O Format:

There will be no initial input. Every time your program makes a guess, the grader will output ‘C’, ‘Y’, or ‘N’, which your program can then read via the usual input channel. The grader will only respond to the first 750 guesses. **Make sure that you end each guess with a new line.**

### Sample Interaction:

```
CorectH07sEBaT3RyS1Ap1      N
unt2                          Y
HuNtEr2                       N
run2the                       N
hunter2                       C
```

In this example, the program guessed the password correctly with 5 attempts.

*Note:* You are responsible for making sure your program terminates after receiving the 'C' response. If you make further queries after guessing the password you may receive a time limit error. Also, make sure that you flush the output stream every time your program makes a guess. For example, in C++ you may use `fflush(stdout)` or `cout << flush`. In Java you can use `System.out.flush()`, but it may be easier to use `System.out.println()` to simultaneously print a guess and flush the output stream at once. In Python you can use `stdout.flush()` after you import `stdout` from the `sys` module. You can learn more about interactive problems [here](#).