2025 T1 O-Week Snake Contest Problems

UNSW CPMSoc

10 February 2025

Contents

Cobra Cubes [Maths]

Cobras consume chicken cubes.

What is $1^3 + 2^3 + 3^3 + 4^3 + 5^3 + 6^3 + 7^3 + 8^3 + 9^3$?

Enter your answer into the box below to submit and check your answer!

Boa Buckets [Maths]

Identical buckets containing boas are to be stacked carefully in a square pyramid or else the boas might break out.

Buckets on each layer are in a square arrangement, and the bottom of each bucket makes contact with the top of four buckets from the layer below. The buckets are all inverted truncated cones.

If the radius of the top of each bucket is 300 mm, what is the minimum radius of the bottom, rounded up to the nearest mm?

Python Points [Maths]

Lucy likes to consider the python points of numbers. The python points of a number is the sum of all of its digits.

For example, the python points of 986 would be $9 + 8 + 6 = 23$ and the python points of 30 would just be 3.

Subtask 1 (40% of points)

If they calculate the python points of all numbers between 423 and 968, how many distinct values will they find?

Subtask 2 (30% of points)

If they calculate the python points of all numbers between 375 and 400, how many distinct values will they find?

Subtask 3 (30% of points)

If they calculate the python points of all numbers between 596 and 632, how many distinct values will they find?

Submission

Submit your answer to the subtasks as a comma-separated list of integers. For example, if your answers to the subtasks are 1, 3 and 4, you should submit `1,3,4`. Note that if you have not solved a subtask, you can submit a dummy answer for that subtask. For example, if your answer to the first subtask is 1, you could submit `1,0,0`.

Labelling Die [Maths]

While cleaning his room, Frank found a perfect regular icosahedron.

He wants to label the faces from 1 to 20 to use as a die, but he doesn't know how it is usually done.

How many possible ways are there to label it? Two labellings are considered the same if you can rotate one to match the other.

Exam Scores [Maths]

You are walking through K17 when you overhear the following conversation.

Kyle: Ughh my students did so bad in the latest competition.

Frank: How bad?

Kyle: Do you want to guess their scores? I have three students and the product of their scores is 90.

Frank: You aren't giving me enough information.

Kyle: Well, the sum of their scores is your sister's age.

Frank: Hmmm, that still isn't enough information.

Kyle: The student with the lowest score is Jerry.

Frank: Oh, now I know the scores.

You know that the competition only allows nonnegative integer marks. What are the scores of Kyle's students?

Submit your answer as a comma-separated list of the scores. For example, if you think the scores are 1, 3 and 4, enter `1,3,4` as your answer.

Bouncing [Maths]

This task has subtasks. You are encouraged to solve them one at a time, and submit your answers as you go.

Sally's machine takes an input sequence A with N integers between 0 and 9, and outputs a new sequence B of length $N-1$, where $B_i = A_{i+1} - A_i$

For each subtask, if all the sequences of length N are inputted, how many unique sequences will be outputted?

Subtask 1 (30% of points) $N = 3$

Subtask 2 (30% of points) $N = 6$

Subtask 3 (40% of points) $N = 15$

Submission Submit your answer to the subtasks as a comma-separated list of integers. For example, if your answers to the subtasks are 1, 2 and 3, you should submit `1,2,3`. Note that if you have not solved a subtask, you can submit a dummy answer for that subtask. For example, if your answer to the first subtask is 1, you could submit `1,0,0`.

Number Theory 2 [Maths]

For what positive integers n will $n^2 + 3n - 3$ be a perfect square?

For full marks, you must also show that you have found all possible answers.

Taking Stones [Maths]

Alice and Bob are playing a game. There is initially a pile of N stones, and Alice and Bob take turns removing stones from the pile.

Each turn, the number of stones a player removes must be a power of two. The player who takes the last stone wins.

If Alice goes first, for what values of N can she guarantee a win?

Note: 1 is considered a power of two.

Containment [Maths]

Jerry is trying to model the containment of a disease on an infinite grid of squares.

Initially, 2025 of these squares are “infected” while the rest are “normal”.

Then, after each second, each square that is “infected” and currently adjacent to two or more “normal” squares will be cured and turn back to “normal”. This process happens simultaneously across all squares.

Two squares are considered adjacent if they share an edge.

How long (in seconds) must it take to guarantee that all the squares have turned back to “normal”, regardless of which 2025 squares are initially “infected”?

You must also show that your answer is minimal.

Guardian [Maths]

You have found cpmsoc's secret lair, but you must answer the guardian snake's question before you can enter (or you will be eaten).

Find all pairwise coprime positive integers $a \geq b \geq c \geq d$ such that:

- $a \mid bc + bd + cd - 3$
- $b \mid ac + ad + cd - 3$
- $c \mid ab + ad + bd - 3$
- $d \mid ab + ac + bc - 3$

For full marks, you must also show that you have found all possible answers.

Snake Sharing [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

You have S (jelly) snakes to share between yourself and F other friends. You do this by splitting the snakes between $F + 1$ piles, and then letting each friend choose which pile they'd like. Your friends, being rather selfish, will always choose the piles with the most snakes, and leave you with one pile that has the least.

Having a sweet tooth yourself, you want to know how many snakes you can keep for yourself.

Input

The first and only line contains two integers, S, the total number of snakes, and F, the number of friends you are sharing with.

You should read from standard input.

In Python, you could use the line `S, F = map(int, input().split())`.

In C or C++, you could use the line `int S, K; scanf("%d%d", &N, &K);`.

Output

Output a single integer, the most snakes you can keep for yourself.

You should write to standard output.

In Python, you could use the line `print(answer)`.

In C or C++, you could use the line `printf("%d\n", answer);`.

Constraints

For all test cases:

- $0 \le S \le 1000$
- $0 \le F \le 10$

Sample Input 1

```
10 4
```

Sample Output 1

```
2
```

Explanation 1

With 10 snakes and 4 friends, you can make 5 piles, each with 2 snakes. You will be left with one of these piles, resulting in keeping 2 snakes for yourself. It can be shown that you cannot get 3 or more snakes for yourself, so the answer is 2.

Scoring

Your program will be run on the sample case and nine secret cases one after another, and if it produces the correct output for all test cases, it solves this problem. Recall that your final score on the task is the score of your highest scoring submission.

Serpent Stretching [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

Ssserpents ssslither through a winding cave to ssstretch to their full length, N. Due to the ssstrange ssshape of the cave, the snake will end up in the following ssshape.

```
###
..#
###
#..
###
..#
###
#..
```

and so on.

Given the length N, print the resulting shape of the snake. The snakes always begin in the top left corner of the grid and start moving towards the right.

Input

The first and only line of input contains one integer N, where N is the length of the snake.

You should read from standard input.

In Python, you could use the line `N = int(input())`.

In C or C++, you could use the line `int N; scanf("%d", &N);`.

Output

Output the shape of the snake, using `#` to represent spaces with the snake and `.` for spaces without. Each line of the output should have exactly three characters. Output lines if and only if the snake is on the line.

You should write to standard output.

In Python, you could use the following code.

```python
for row in answer:
    print(row)
```

In C or C++, you could use the following code.

```c
for (int i = 0; i < r; i++) {
    printf("%s\n", answer[i]);
}
```

Constraints

For all test cases:

- $3 \leq N \leq 1000$

Sample Input 1

```
14
```

Sample Output 1

```
###
..#
###
#..
###
..#
.##
```

Explanation 1

The snake follows the shape of the curve, and ends once there have been 14 #s.

Scoring

Your program will be run on the sample case and nine secret cases one after another, and if it produces the correct output for all test cases, it solves this problem. Recall that your final score on the task is the score of your highest scoring submission.

Rattlesnake Recursion [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

You are looking at rattlesnake population growth when you realise it follows a recursive trend. Indeed, if one year has R rattlesnakes, then the next year will have $X \times R + Y$ rattlesnakes. The coefficient X is constant, whereas Y can vary year to year between L and R, inclusive.

Given a starting population A, target population T, and the growth parameters X, L, and R, return the least number of years it will take for the population to reach exactly T, or return -1 if this is impossible.

Input

The first and only line contains five integers, A, T, X, L, R. Respectively, they represent:

- The starting population in year 0
- The target population which much be reached exactly
- The amount the population is multipled by each year
- The least amount added to the population each year
- The greatest amount added to the population each year

You should read from standard input.

In Python, you could use the line `A, T, X, L, R = map(int, input().split())`.

In C or C++, you could use the line `int A, T, X, L, R; scanf("%d %d %d %d %d", &A, &T, &X, &L, &R);`.

Output

Output a single integer. If T can be reached, output the least number of years it will take. If T cannot be reached, output `-1`.

You should write to standard output.

In Python, you could use the line `print(answer)`.

In C or C++, you could use the line `printf("%d\n", answer);`.

Constraints

For all test cases:

- $1 \leq A \leq T \leq 10^{18}$
- $1 \leq X \leq 10^6$
- $1 \leq L \leq R \leq 10^6$

for Subtask 1 (20% of points):

- $X = 1$

For subtask 2 (20% of points):

- $1 \leq A \leq T \leq 10^6$.

For subtask 3 (60% of points):

- There are no additional constraints.

You may need to use `long long` instead of `int` in C and C++.

Sample Input 1

```
3 53 4 1 3
```

Sample Output 1

```
2
```

Explanation 1

The fastest way is to go 3 → 13 → 53. This takes two years.

Sample Input 2

```
5 30 4 1 9
```

Sample Output 2

```
-1
```

Explanation 2

With the given parameters, it is impossible to have a population of 30.

Scoring

For each subtask (worth 20%, 20% and 60% of points, as per the Constraints section), your program will be run on multiple secret test cases one after another, and if it produces the correct output for **all** test cases, it solves that subtask. Your program will receive the points for each subtask it solves. Recall that your final score on the task is the score of your highest scoring submission.

Snake Housing [Programming]

Program time limit: 1.5 seconds

Program memory limit: 512 MB

Cpmsoc owns N snakes, with the i th snake having a weight of W_i.

Kyle is arranging miniature homes for these snakes. Each must house exactly two snakes.

Furthermore,

- The sum of the weights of the two snakes must be less than or equal to A, or they will be too big for the house
- The difference of the weights of two snakes must be less than or equal to B, or the larger snake will eat the smaller one

How many pairs of snakes can Kyle house?

Input

The first line contains three integers N, A and B.

The second line contains N integers $W_1, W_2, W_3, \cdots, W_N$.

Output

Output a single integer representing the maximum number of pairs of snakes Kyle can house.

Constraints

For all test cases:

- $2 \leq N \leq 10^6$
- $0 \leq A, B \leq 10^9$
- $0 \leq W_i \leq 10^9$ for all i

For Subtask 1 (30% of points):

- $N \leq 10^3$

For Subtask 2 (70% of points):

- There are no additional constraints.

Sample Input 1

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

Sample Output 1

```
1
```

Explanation 1

You can form a pair of snakes with weights 1 and 2. This pair is valid because $1 + 2 \leq 5$ and $2 - 1 \leq 2$.

There is no arrangement that houses two pairs of snakes.

Sample Input 2

```
7 15 4
4 5 6 7 8 9 11
```

Sample Output 2

```
3
```

Explanation 2

You can form three pairs of snakes: - the snakes with weights 4 and 5 - the snakes with weights 6 and 9 - the snakes with weights 7 and 8

Scoring

For each subtask (worth 30% and 70% of points, as per the Constraints section), your program will be run on multiple secret test cases one after another, and if it produces the correct output for **all** test cases, it solves that subtask. Your program will receive the points for each subtask it solves. Your final score on the task is the score of your highest scoring submission.

Friends Pairing [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

In the town of Snake-ville, there are N snakes, each with a unique value. These snakes are connected by $N - 1$ friendship bonds, forming a tree structure.

Jerry has now tasked you with performing the following operation on the tree any number of times and outputting the maximum value that you can achieve. The operation you can perform is:

- Pick two directly connected snakes u and v.
- They decide to move away together and live their best lives.
- You gain $A_u + A_v$ friendship points.
- These two friends leave the town, and all their friendship bonds are removed.

Input

The first line contains one single integers N.

The second line contains N integers $A_1, A_2, A_3, \cdots, A_N$, denoting the value of each snake.

For the following $N - 1$ lines, each one contains two integers B_i and C_i denoting that there is a friendship bond between B_i and C_i.

Output

Output a single integer representing the maximum value you can achieve.

Constraints

For all test cases:

- $1 \leq N \leq 10^6$
- $1 \leq B_i, C_i \leq N$
- $0 \leq A_i \leq 10^9$ for all i

For Subtask 1 (30% of points):

- $B_1 = 1$ and $C_{n-1} = n$ and $C_i - B_i = 1$.

For Subtask 2 (70% of points):

- There are no additional constraints.

Sample Input 1

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

Sample Output 1

```
12
```

Explanation 1

You can form two friendship pairs. The first one between the second and third snake for 9 points and the second between the fourth and fifth snake for 3 points.

Sample Input 2

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

Sample Output 2

```
14
```

Explanation 2

You can form two pairs of snakes:

- the third and fourth snake
- the first and second snake

Scoring

For each subtask (worth 30% and 70% of points, as per the Constraints section), your program will be run on multiple secret test cases one after another, and if it produces the correct output for **all** test cases, it solves that subtask. Your program will receive the points for each subtask it solves. Your final score on the task is the score of your highest scoring submission.

Most Common Skyline [Programming]

Program time limit: 9 seconds

Program memory limit: 1 GB

You are given an array A of size N. The elements inside this array are labelled as $A_1, A_2, \cdots, A_n$.

The skyline of this array is an array A' of size N where $A'_i = \max(A_1, A_2, \cdots, A_i)$.

For example, the skyline of $A = [3, 2, 4, 2, 9]$ would be $A' = [3, 3, 4, 4, 9]$.

You must support Q operations of the form:

- Given elements X and Y, swap the elements A_X and A_Y and output the maximum number of times an element appears in the skyline of the new array A.

Operations are given in order and affect future operations.

Input

The first line the integers N and Q.

The second line contains N integers $A_1, A_2, \cdots, A_N$.

The next Q lines contain two integers X_i and Y_i, the X and Y values for the i th operation.

Output

Output Q lines, with the i th line containing the output from the i th operation.

Constraints

For all test cases:

- $2 \leq N \leq 10^6$
- $1 \leq Q \leq 10^6$
- $1 \leq X_i, Y_i \leq N$, $X_i \neq Y_i$ for all i
- $1 \leq A_i \leq 10^9$ for all i

For Subtask 1 (10% of points):

- $N, Q \leq 10^3$

For Subtask 2 (30% of points):

- $N, Q \leq 10^5$

For Subtask 3 (60% of points):

- There are no additional constraints.

Sample Input

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

Sample Output

```
1
5
3
2
```

Explanation

After the first operation, the array A will be `1 2 3 4 5` and the skyline A' will be `1 2 3 4 5`. The maximum number of times an element appears is 1.

After the second operation, the array A will be `5 2 3 4 1` and the skyline A' will be `5 5 5 5 5`. The maximum number of times an element appears is 5.

After the third operation, the array A will be `4 2 3 5 1` and the skyline A' will be `4 4 4 5 5`. The maximum number of times an element appears is 3.

After the fourth operation, the array A will be `2 4 3 5 1` and the skyline A' will be `2 4 4 5 5`. The maximum number of times an element appears is 2.

Scoring

For each subtask (worth 10%, 30% and 60% of points, as per the Constraints section), your program will be run on multiple secret test cases one after another, and if it produces the correct output for **all** test cases, it solves that subtask. Your program will receive the points for each subtask it solves. Your final score on the task is the score of your highest scoring submission.