Launch Week Penguin Contest 2024 Problems

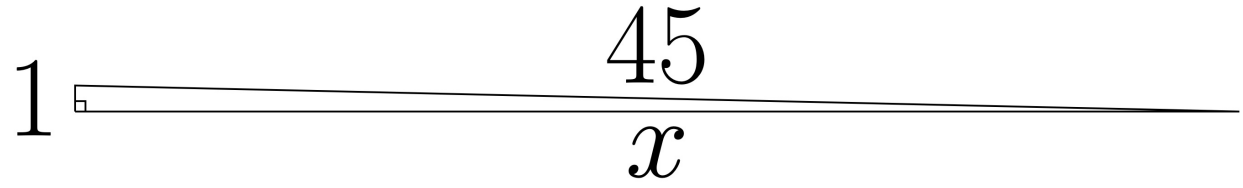
UNSW CPMSoc

27 May 2024

Contents

Trangle [Maths]

Pingu has a geometry question for you! There is a right-angled triangle whose longest side has length 45 and whose shortest side has length 1. If x is the length of the other side, what is x^2?



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

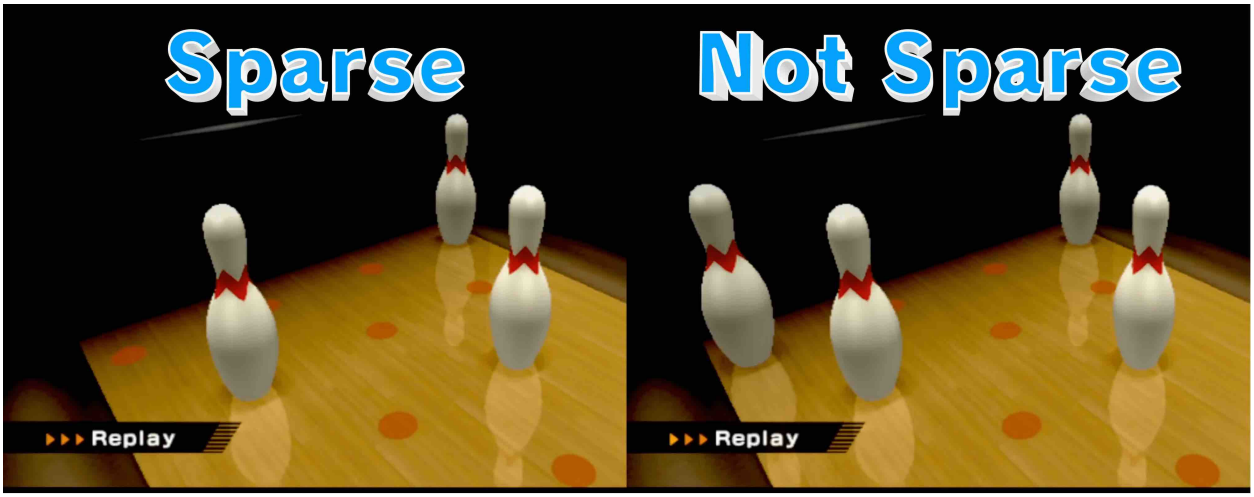
For all the tasks, including this one, you can submit as many times as you like with no penalty! Only your highest-scoring submission will be counted.

Sparse [Maths]

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

In ten-penguin bowling there are ten penguins arranged in a triangle.

After some penguins have been knocked over, the remaining state is called *sparse* if no two standing penguins are adjacent. Each penguin is considered to be distinct, so rotations and flips are counted as different states.



Subtask 1 (30% of points)

How many sparse states exist with 1 standing penguin?

Subtask 2 (30% of points)

How many sparse states exist with 2 standing penguins?

Subtask 3 (30% of points)

How many sparse states exist with 3 standing penguins?

Subtask 4 (10% of points)

How many sparse states exist with 4 standing penguins?

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, 3 and 4, you should submit "1,2,3,4". 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 "1,0,0,0".

Difference Closure [Maths]

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

Kowalski is performing some analysis on sets of integers.

Kowalski says a set S of integers is *difference-closed* if for all $a, b \in S$, we have $|a - b| \in S$.

Kowalski defines the *difference-closure* of a set as the smallest difference-closed set containing it.

Subtask 1 (40% of points)

Find the size of the difference-closure of $\{14, 49\}$.

Subtask 2 (60% of points)

Find the size of the difference-closure of $\{185, 111111, 111111111111\}$.

Submission

Submit your answer to the subtasks as a comma-separated list of integers. For example, if your answers to the subtasks are 1 and 2, you should submit "1,2". 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".

Skibidi Progression [Maths]

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

Wheezy is a student at Skibidi Penguin School. Wheezy likes integers; integers are nice. A *Skibidi* sequence is a sequence of integers such that there exist positive integers a and b such that for all terms x (except the last term), the next term is $ax + b$.

Subtask 1 (3 points)

How many Skibidi sequences exist beginning with 7 and ending in 606?

Subtask 2 (4 points)

How many Skibidi sequences exist beginning with 7 and ending in 607?

Submission

Submit your answer to the subtasks as a comma-separated list of integers. For example, if your answers to the subtasks are 1 and 2, you should submit "1,2". 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".

Skibidi Explanation [Maths]

Explain why your answer to Subtask 2 of Skibidi Progression is correct.

Scoring

Your submission will be manually scored based on merit.

Divisors [Maths]

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

Let $\text{penguin}(n, m) = \sum_{d|n} \text{penguin}(d, m-1)$ for positive integers n and m, and let $\text{penguin}(n, 0) = 1$.

Here, $\sum_{d|n}$ denotes that we sum over all positive integers d which are divisors of n.

Subtask 1 (10% of points)

Evaluate $\text{penguin}(2024, 1)$.

Subtask 2 (20% of points)

Evaluate $\text{penguin}(2024^5, 1)$.

Subtask 3 (30% of points)

Evaluate $\text{penguin}(2024^5, 2)$.

Subtask 4 (40% of points)

Evaluate $\text{penguin}(2024^5, 2024)$.

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, 3 and 4, you should submit "1,2,3,4". 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 "1,0,0,0".

Lachlam's Problem [Maths]

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

Lachlam's a problem, but he's also *got* a problem. You see, he's got a rotating, 16-digit combination lock on his bike where each digit can cycle between the values from 0 to 9. Right now, all the digits are set to 0. Lachlam remembers that, from the correct lock combination, he only shifted the numbers up, and that the least he shifted a digit was by exactly X spots, and the most he shifted a digit was by exactly $X + D$ spots.

Note that digits wrap around, so shifting 9 up by 1 turns it into 0. For example, if the correct lock combination is 5678876555556767 (and Lachlam shifted all the digits up to 0 in the least number of required moves), then $X = 2$ and $D = 3$. Unfortunately, Lachlam can't remember X nor the exact amount every digit was shifted by, because a penguin fell on his head, but he does remember D. Help show Lachlam how futile his efforts are by answering the following questions.

Subtask 1 (30% of points)

How many possible correct lock combinations exist for $D = 1$?

Subtask 2 (40% of points)

How many possible correct lock combinations exist for $D = 3$?

Subtask 3 (30% of points)

How many possible correct lock combinations exist for $D = 5$?

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".

Triangle Formation [Maths]

Let $n \geq 3$ be an integer. Independently and uniformly at random, n real numbers between 0 and 1 are chosen. Find in terms of n the probability that three of them can be chosen to form the side lengths of a triangle. You should explain why your answer is correct.

Scoring

Your submission will be manually scored based on merit.

Click B8 [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

As the temperature drops, Pip the Penguin and her seven friends are preparing to build an igloo. Each of the eight penguins, including Pip, has a favourite integer B_i.

Pip decides that the number of ice blocks in the igloo should be the sum of $B_i \times B_j$ over all indices i and j such that $i < j$. Determine the total number of ice blocks Pip needs.

Input

The first and only line of input contains eight integers $B_1, B_2, \ldots, B_8$, which are the favourite integers of the penguins.

You should read from standard input.

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

In C or C++, you could use the line `int B[8]; for (int i = 0; i < 8; i++) scanf("%d", &B[i]);`.

Constraints

For all test cases:

- $1 \leq B_i \leq 1000$ for all i.

Output

Output a single integer, the sum of $B_i \times B_j$ over all indices i and j such that $i < j$.

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);`.

Sample Input 1

```
8 3 100 0 0 0 0 0
```

Sample Output 1

```
1124
```

Explanation 1

In this case, the penguins with favourite integer 0 can be ignored since they do not contribute to the answer. So the answer is $8 \times 3 + 8 \times 100 + 3 \times 100 = 24 + 800 + 300 = 1124$.

Sample Input 2

```
1 1 1 1 1 1 1 1
```

Sample Output 2

```
28
```

Explanation 2

In this case, all numbers are 1. Thus, the answer is the number of pairs of indices such that $i < j$, which is 28.

Scoring

Your program will be run on the 2 sample cases and 8 secret cases one after another, each worth 10% of the points. Recall that your final score on the task is the score of your highest scoring submission.

TV Remote [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

Mumble has a minimalist TV remote with two buttons: `+` to increase the volume by 1, and `-` to decrease the volume by 1. The volume is capped between 1 and K, so any button press is ignored if the volume would go outside that range.

You are given a sequence of N button presses. Help Mumble by determining how many distinct ending volumes are possible after performing the sequence of button presses, over all choices of starting volume from 1 to K inclusive.

Input

- The first line of input contains two integers N and K, where N is the number of button presses and K is the maximum volume.
- The second line contains one string of N characters, each either `+` or `-`, denoting the sequence of button presses.

You should read from standard input.

In Python, you could use the following code.

```
N, K = map(int, input().split())
buttons = input().strip()
```

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

```
int N;
scanf("%d", &N);
char buttons[N+1];
scanf("%s", buttons);
```

Constraints

For all test cases:

- $1 \leq N, K \leq 100\,000$.
- Each character of the string is either `+` or `-`.

Additionally:

- For Subtask 1 (60% of points), $N, K \leq 1000$.
- For Subtask 2 (40% of points), there are no additional constraints.

As a hint, to solve Subtask 2, you may need a more efficient algorithm than one which simulates all N button presses for all K possible starting volumes. This would take (very roughly) $100\,000 \times 100\,000 = 10^{10}$ steps in the worst case, but you can expect a computer to perform up to (very very roughly) 10^7 steps in one second, which is many orders of magnitudes too slow.

Output

Output a single integer, the number of distinct possible ending numbers.

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);`.

Sample Input 1

```
5 4
+-++-
```

Sample Output 1

```
2
```

Explanation 1

Here the maximum volume is 4.

If the starting volume is 1, then the sequence of volumes is $1 \xrightarrow{+} 2 \xrightarrow{-} 1 \xrightarrow{+} 2 \xrightarrow{+} 3 \xrightarrow{-} 2$.

If the starting volume is 2, then the sequence of volumes is $2 \xrightarrow{+} 3 \xrightarrow{-} 2 \xrightarrow{+} 3 \xrightarrow{+} 4 \xrightarrow{-} 3$.

If the starting volume is 3, then the sequence of volumes is $3 \xrightarrow{+} 4 \xrightarrow{-} 3 \xrightarrow{+} 4 \xrightarrow{+} 4 \xrightarrow{-} 3$.

If the starting volume is 3, then the sequence of volumes is $4 \xrightarrow{+} 4 \xrightarrow{-} 3 \xrightarrow{+} 4 \xrightarrow{+} 4 \xrightarrow{-} 3$.

Thus there are 2 possible ending volumes.

Sample Input 2

```
5 8
++---
```

Sample Output 2

```
5
```

Scoring

For each subtask (worth 60% and 40% 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.

Penguin Army [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

You wake up ready to go diving for fish, but find that you're in a picturesque meadow, a long way from home! You quickly piece together that in order to escape, you'll need to create the largest possible army you can, but luckily you have some abilities to help you do just that!

You start your journey with an army of N penguins, and must complete K missions in order. Within each mission you must use each of your 5 abilities exactly once, but you can choose the order to use them in.

The ith mission has five integer ability strengths associated with it, which are a_i, b_i, c_i, d_i and e_i. The five abilities work as follows.

- Apparition: Add a_i penguins to your army.
- Bewitchment: Subtract b_i penguins from your army. If your army size is now negative, you immediately fail your journey.
- Consolidation: Divide your army size by c_i, rounding down.
- Duplication: Multiply your army size by d_i.
- Examination: If your army size is strictly greater than e_i, your army increases by e_i. Otherwise you immediately fail your journey.

Can you complete your journey? If so, output the size of the largest army you can have after all missions have been completed.

Input

- The first line of input contains two integers N and K, where N is the starting number of penguins and K is the total number of missions.
- Then follow K lines, the ith of which contains five integers a_i, b_i, c_i, d_i and e_i, which are the ability strengths of the ith mission.

You should read from standard input.

In Python, you could use the following code.

```
N, K = map(int, input().split())
missions = [list(map(int, input().split())) for _ in range(K)]
```

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

```
int N, K;
scanf("%d%d", &N, &K);
int a[K], b[K], c[K], d[K], e[K];
for (int i = 0; i < K; i++) {
    scanf("%d%d%d%d%d", &a[i], &b[i], &c[i], &d[i], &e[i]);
}
```

Constraints

For all test cases:

- $0 \leq N \leq 10^9$.
- $1 \leq K \leq 1000$.
- $1 \leq a_i, b_i, c_i, d_i, e_i \leq 10^9$ for all i.
- It is impossible for the current army size to exceed 10^9 at any point.

Additionally:

- For Subtask 1 (50% of points): $K = 1$.

- For Subtask 2 (50% of points): there are no additional constraints.

Output

If you cannot complete your journey, output `-1`. Otherwise, output a single integer, the size of the largest army you can have after all missions have been completed.

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);`.

Sample Input 1

```
7 2
2 9 4 7 9
2 8 2 1 10
```

Sample Output 1

```
15
```

Explanation 1

On the first mission, we can perform the abilities in the following order.

- Apparition of strength 2: we now have $7 + 2 = 9$ penguins.
- Duplication of strength 7: we now have $9 \times 7 = 63$ penguins.
- Bewitchment of strength 9: we now have $63 - 9 = 54$ penguins.
- Consolidation of strength 4: we now have $54 \div 4 = 13$ penguins.
- Examination of strength 9: we now have $13 + 9 = 22$ penguins.

On the second mission, we can perform the abilities in the following order.

- Consolidation of strength 2: we now have $22 \div 2 = 11$ penguins.
- Examination of strength 10: we now have $11 + 10 = 21$ penguins.
- Bewitchment of strength 8: we now have $21 - 8 = 13$ penguins.
- Apparition of strength 2: we now have $13 + 2 = 15$ penguins.
- Duplication of strength 1: we now have $15 \times 1 = 15$ penguins.

Our army size is 15 at the end, which turns out to be the maximum possible.

Scoring

For each subtask (worth 50% and 50% 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.

Forensics [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

You had an array, but someone came along and made segtree operations on it! Being the forensics expert you are, you decide to investigate deeper.

Initially, every element of the array was 0. Then, some number of operations were made to the array. The operations that were made were of the form (l, r, x), where x was added to every array element from index l to index r inclusive. It is known that $1 \leq l \leq r \leq N$, and that x is a positive integer.

Also, the set of operations made is known to be laminar, meaning that for any two operations (l_1, r_1, x_1) and (l_2, r_2, x_2), their ranges either do not share an index, or one range is included in the other. More precisely, it is never the case that $l_1 < l_2 \leq r_1 < r_2$.

Investigating deeper, you realise the operations were made in a rush, which means that whoever did the operations had limited time to operate on the array. Thus, you conclude that they must have used the minimum number of operations possible to generate this particular array. Now, you must write a program to find that minimum number and close the case once and for all.

Input

- The first line of input contains one integer N, the number of elements in the array.
- The next line contains N integers, the ith of which is A_i, the ith element in the final array.

Output

Output a single integer, the minimum possible number of laminar segtree operations to generate the given array.

Constraints

For all test cases:

- $1 \leq N \leq 100\,000$.
- $0 \leq A_i \leq 10^9$ for all i.

Additionally:

- For Subtask 1 (20% of total points): $A_i = A_j$ for all i and j.
- For Subtask 2 (60% of total points): $A_i \neq 0$ for all i.
- For Subtask 3 (20% of total points): there are no additional constraints.

Sample Input 1

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

Sample Output 1

```
3
```

Explanation 1

In this case, 3 segtree operations could have been made: $(1, 5, 1)$, $(2, 4, 1)$ and $(3, 3, 1)$. This is the minimum number of operations possible.

Sample Input 2

```
3
0 0 0
```

Sample Output 2

```
0
```

Explanation 2

In this case no segtree operations were made on the array.

Scoring

For each subtask (worth 20%, 60% and 20% 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.

President [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

Tux, the President of GridLand, has decided to some build roads between some neighbouring pairs of cities. The cities form a grid with N rows and M columns. In order to be a good president, he needs to build roads so that there is exactly one path between any pair of cities, and so that the longest such path contains exactly K roads.

Is it possible for Tux to build the roads in such a way? If so, tell him which roads to build.

Input

The first and only line of input contains three integers N, M, and K, where N is the number of rows, M is the number of columns, and K is the length of the longest path.

Constraints

For all test cases:

- $1 \leq N, M \leq 1\,000$.
- $0 \leq K \leq 1\,000\,000$.

Additionally:

- For Subtask 1 (20% of points): $N = 2$.
- For Subtask 2 (20% of points): $M = 3$.
- For Subtask 3 (30% of points): $1 \leq N, M \leq 20$.
- For Subtask 4 (30% of points): there are no additional constraints.

Output

If there is no way for Tux to build roads in a valid way, then output `IMPOSSIBLE`.

Otherwise, output NM lines. The first line contains the word `POSSIBLE`. Each of the following lines contains four integers x_1, y_1, x_2, y_2, denoting a road between the neighbouring cities at (x_1, y_1) and (x_2, y_2), where $1 \leq x_1, x_2 \leq N$ and $1 \leq y_1, y_2 \leq M$.

If there are multiple correct outputs, any one of them will be accepted.

Sample Input 1

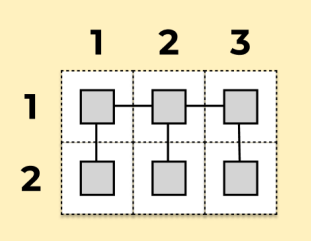
```
2 3 4
```

Sample Output 1

```
POSSIBLE
1 1 1 2
1 1 2 1
1 2 1 3
1 2 2 2
1 3 2 3
```

Explanation 1

This sample case is illustrated below.



Sample Input 2

```
2 3 2
```

Sample Output 2

```
IMPOSSIBLE
```

Explanation 2

In this sample case, it is impossible for the longest path to contain less than 3 roads.

Scoring

For each subtask (worth 20%, 20%, 30% and 30% 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.

Training [Programming]

Program time limit: 2 seconds

Program memory limit: 512 MB

For the upcoming ICPC training season, UNSW has decided to put the teams into groups, to increase the efficiency of the training.

There are M rivalries between pairs of teams. Every team has a rivalry with at most K other teams, where $K \leq 3$. However, the university wants the teams in the same group to work together and not compete with each other. Thus, they decided to create the groups in a way such that every team has a rivalry with at most one team in the same group.

Team IDLE is wondering how to form groups so that the number of groups is minimised.

Input

- The first line of input contains three integers N, M, and K, where N is the number of teams, M is the number of rivalries, and K is the maximum number of rivalries a team can have.
- Then follow M lines, the ith of which contains two integers a_i and b_i, denoting a rivalry between team a_i and team b_i.

Output

- The first line of output contains one integer G, the minimum number of groups.
- The second line contains N integers $x_1, x_2, \ldots x_n$, where $1 \leq x_i \leq G$ and x_i is the group number of the ith team for all i.

If there are multiple correct outputs, any one of them will be accepted.

Constraints

For all test cases:

- $1 \leq N \leq 100\,000$.
- $0 \leq M \leq KN/2$.
- $1 \leq K \leq 3$.
- $1 \leq a_i, b_i \leq N$ for all i.
- Each team has a rivalry with at most K other teams.

Additionally:

- For Subtask 1 (20% of points): $1 \leq N \leq 20$.
- For Subtask 2 (20% of points): $1 \leq N \leq 100$.
- For Subtask 3 (30% of points): $K \leq 2$.
- For Subtask 4 (30% of points): there are no additional constraints.

Sample Input 1

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

Sample Output 1

```
1
1 1 1 1
```

Explanation 1

Since we can put everyone in the same group the minimum number of groups is 1.

Sample Input 2

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

Sample Output 2

```
2
1 1 2 2
```

Explanation 2

In this case, we cannot put everyone in the same group because if we did, then team 2 would have two rivals in the same group.

Scoring

For each subtask (worth 20%, 20%, 30% and 30% 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.

Real Life Problem [Other]

There is a graph embossed on a plaque underneath the sundial at UNSW's Quadrangle Building.

What are the numbers of the months containing the turning points of the curve?

Submission

Submit your answer as a comma-separated list of integers in increasing order, representing the months. For example, if the months are January, February and March, you should submit "1,2,3".

Feedback [Other]

Receive the answer to this task by completing the contest feedback form here.