Codeferno 2025

Perseverantia's Coding Contest

Problem Set

brought to you by Bombay Scottish School

October 3rd, 2025

Welcome to Codeferno

General Information

- Contest Website: `192.xxx.xxx.xxx:[port]`
- Duration: **90 minutes**

Rules & Guidelines

1. You may use **C / C++ / Java / Python**.
2. For Java:
   - The filename must be the **problem ID**. Example: `A.java`.
   - Always include `public static void main(String[] args)`.
3. Do *not* print prompts such as `"Enter n:"`.
4. Follow the input/output format in the problem statement **exactly**. Any deviation will cause your submission to fail.

   For example: If the problem specifies output as

   ```
   1 2 3 4 5
   ```

   and you print

   ```
   1
   2
   3
   4
   5
   ```

   your answer will be marked wrong.

5. The exact required format will always be clearly specified in each problem.

Scoring

- Each problem carries a certain number of points.
- Partial scoring will be given for passing some subtasks.
- Ties are broken by **time of last correct submission**.

Allowed References

- Documentation provided locally during the contest:
  - Python: `[TODO link]`
  - C: `[TODO link]`
  - C++: `[TODO link]`
  - Java: `[TODO link]`
- You may freely refer to these official docs during the contest.

Prohibited Actions

- No internet access is allowed.
- Do **not** attempt to Google or consult any other online source.
- Do **not** attempt to use AI language models or external tools to solve the problems.
- Any violation will result in disqualification.

Good luck; may the best coder win!

Sample Problem Statement

Introduction

This document contains an example of a problem that might appear in Codeferno. It is provided to help you get familiar with the problem format, input/output style, subtasks, and solutions in different programming languages (C, C++, Python, and Java).

Problem A: Sum of Two Numbers

You are given two integers a and b. Your task is to compute their sum.

Input

Two integers a and b.

Output

Output a single integer, the value of $a + b$.

Constraints

Subtask	Constraints	Points
1	$-10^9 \leq a, b \leq 10^9$	30
2	$-10^{18} \leq a, b \leq 10^{18}$	70

Sample Input

```
5 7
```

Sample Output

```
12
```

Solutions

C (filename: any .c file)

```c
#include <stdio.h>

int main() {
  long long a, b;
  scanf("%lld %lld", &a, &b);
  printf("%lld\n", a + b);
}
```

C++ (filename: any .cpp file)

```cpp
#include <bits/stdc++.h>

using namespace std;

int main() {
  long long a, b;
  cin >> a >> b;
  cout << a + b << "\n";
}
```

Python (filename: any .py file)

```python
a, b = map(int, input().split())
print(a + b)
```

Java (filename: A.java)

```java
import java.util.*;

public class A {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    long a = sc.nextLong();
    long b = sc.nextLong();
    System.out.println(a + b);
  }
}
```

Notes

- In C, C++, and Python, the filename does not matter.
- In Java, the filename must match the problem ID. For example, for this problem (Problem A), the file should be named `A.java`.
- Always follow the input/output format strictly. Do not add prompts or extra text.

Neighbours

Problem ID: neighbours

Problem Statement

There are N houses built along a straight road. The position of the i-th house is given by an integer x_i. The houses are listed in order from left to right ($x_1 < x_2 < \cdots < x_N$).

Your task is to find the **minimum distance** between any two neighbouring houses.

Input

- The first line contains an integer N ($2 \leq N \leq 1000$), the number of houses.
- The second line contains N integers $x_1, x_2, \ldots, x_N$ ($0 \leq x_i \leq 10^6$), the positions of the houses. It is guaranteed that $x_1 < x_2 < \cdots < x_N$.

Output

Print a single integer: the minimum distance between two neighbouring houses.

Subtasks

Subtask	Constraints	Points
1	We will only test the provided sample inputs.	10
2	$N \leq 100$, $x_i \leq 1000$	30
3	$N \leq 1000$, $x_i \leq 10^6$	60

Sample Input 1

```
5
1 4 7 12 14
```

Sample Output 1

```
2
```

Explanation for Sample 1

The gaps between neighbours are: $4 - 1 = 3$, $7 - 4 = 3$, $12 - 7 = 5$, $14 - 12 = 2$. The minimum is 2.

Sample Input 2

```
4
10 20 25 40
```

Sample Output 2

```
5
```

Explanation for Sample 2

The gaps are 10, 5, and 15. The minimum is 5.

End of Problem 1

Bus Stops

Problem ID: bus

Problem Statement

A school bus travels along a straight road with N bus stops, numbered from 1 to N. At each stop:

- First, some students get off the bus.
- Then, some students get on the bus.

The bus starts empty before stop 1. Your task is to determine the **maximum number of students on the bus at any time**.

In some subtasks, additional rules apply (such as bus capacity). Read the constraints carefully.

Input

- The first line contains an integer N $(1 \leq N \leq 1000)$, the number of bus stops.
- The second line contains N integers $\mathrm{on}_1, \mathrm{on}_2, \ldots, \mathrm{on}_N$ $(0 \leq \mathrm{on}_i \leq 10^4)$, where on_i is the number of students boarding at stop i.
- The third line contains N integers $\mathrm{off}_1, \mathrm{off}_2, \ldots, \mathrm{off}_N$ $(0 \leq \mathrm{off}_i \leq 10^4)$, where off_i is the number of students getting off at stop i.
- The fourth line contains an integer C, the bus capacity. For subtasks where no capacity restriction applies, C will be very large (e.g. 10^9), so it does not affect the result.

Output

Print a single integer: the maximum number of students on the bus at any point in time. In subtasks requiring the stop index, print two integers: `max_students stop_index`.

Subtasks

Subtask	Constraints	Points
1	$N \leq 3$, $\text{on}_i, \text{off}_i \leq 10$, $C = 10^9$	10
2	$N \leq 100$, $\text{on}_i, \text{off}_i \leq 100$, $C = 10^9$	20
3	$N \leq 1000$, $\text{on}_i, \text{off}_i \leq 10^4$, $C = 10^9$	20
4	Same as Subtask 3, but $C \leq 10^4$. If more students attempt to board than seats available, only as many as possible get on. The rest are left behind permanently.	30
5	Same as Subtask 3. Additionally, print the **first stop index** at which the maximum occupancy occurs. Output format: two integers, `max_students stop_index`.	20

Sample Input 1

```
5
0 3 4 0 2
0 0 2 3 4
1000000000
```

Sample Output 1

```
5
```

Explanation for Sample 1

- Stop 1: 0 off, 0 on → 0
- Stop 2: 0 off, 3 on → 3
- Stop 3: 2 off, 4 on → 5
- Stop 4: 3 off, 0 on → 2
- Stop 5: 4 off, 2 on → 0

Maximum = 5.

Sample Input 2 (capacity example)

```
4
5 5 5 5
0 0 0 0
8
```

Sample Output 2

```
8
```

Explanation for Sample 2

- Stop 1: 5 board → 5
- Stop 2: 5 try to board, but capacity is 8 → 3 board, 2 left behind → total = 8
- Stops 3 and 4: bus remains full at 8

Maximum = 8.

Sample Input 3 (stop index example)

```
6
2 4 0 2 0 0
0 0 2 0 1 4
1000000000
```

Sample Output 3

```
6 2
```

Explanation for Sample 3

- Stop 1: 2 board → 2
- Stop 2: 4 board → 6
- Stop 3: 2 off → 4
- Stop 4: 2 on → 6
- Stop 5: 1 off → 5
- Stop 6: 4 off → 1

Maximum = 6 at stop 2, but maximum **after** that is 6 at stop 4. Since the first maximum is at stop 2, output is `6 2`.

End of Problem 2

Offthentic Feed

Problem ID: feed

Problem Statement

You are building a new social media platform called **Offthentic**. Users post content continuously, and the platform must display a live feed of posts.

There are n posts. Each post i has three attributes:

- u_i — the user ID of the author,
- t_i — the timestamp when the post was created (seconds since the start),
- l_i — the number of likes the post has.

The feed has the following rules:

- At each second, all posts created up to that time are available.
- The feed displays at most k posts at a time.
- Posts are ranked by:
  1. Higher likes (l_i),
  2. Later timestamp (t_i),
  3. Smaller user ID (u_i).
- When a new post enters the feed, it may push out an older one. Once a post leaves the feed, it does not reappear later.

It is guaranteed that no two posts have identical (u_i, t_i) pairs, so ties are always resolvable.

Your task is to simulate the feed and output the order in which posts first appeared. **Note:** Some posts may never appear if they are always out-ranked.

Input

- The first line contains two integers n and k ($1 \leq n, k \leq 10^5$).
- The next n lines each contain three integers u_i, t_i, and l_i ($1 \leq u_i \leq 10^9$, $0 \leq t_i \leq 10^9$, $0 \leq l_i \leq 10^5$).

Output

Print the indices of the posts (1-based, in input order) in the order they **first appear** in the feed. If a post never appears, it should not be printed.

Subtasks

Subtask	Constraints	Points
1	$n \leq 100$, $k = 1$	20
2	$n \leq 2000$, $k \leq 10$	20
3	$n \leq 10^5$, $k \leq 100$	30
4	No additional constraints	30

Sample 1

Input

```
5 2
10 1 5
7 2 5
3 2 8
5 3 5
2 5 10
```

Output

```
1 3 2 4 5
```

Explanation

- At $t = 1$, post 1 appears.
- At $t = 2$, posts 2 and 3 are available. Post 3 has more likes, so it enters; post 2 also enters while post 1 is pushed out.
- At $t = 3$, post 4 joins.
- At $t = 5$, post 5 (10 likes) pushes out post 2.

The order of first appearance is: 1, 3, 2, 4, 5.

Sample 2

Input

```
4 2
1 1 100
2 2 50
3 3 100
4 4 200
```

Output

```
1 2 3 4
```

Explanation

- At $t = 1$, post 1 appears.
- At $t = 2$, post 2 enters with lower likes but fills the empty slot.
- At $t = 3$, post 3 (likes = 100) replaces post 2 (likes = 50).
- At $t = 4$, post 4 (likes = 200) pushes out post 1.

Final order: 1, 2, 3, 4. Post 2 never reappears after being pushed out, even though newer posts arrive.

End of Problem 3