# A. Requirements

## Code (90%)

You can write your code in Java, Python, C, or C++. The *time limit* may vary among different languages, depending on the performance of the language. Your code must be a complete excutable program instead of only a function. We guarantee test data strictly compliance with the requirements in the description, and you do not need to deal with cases where the input data is invalid.

**No AI Assistance or Plagiarism:** All code must be your own. The use of AI tools (e.g., ChatGPT, GitHub Copilot) or copying from external sources or peers is **strictly forbidden**.

Violations of the plagiarism rules will result in 0 points or even **failure** of this course.

**Libraries in this assignment:**

- For C/C++, you can only include standard library.
- For Java, you can only `import java.util.*`
- For Python, you can only import standard library. In other words, you cannot import libraries such as `numpy`.

We provide an example problem to illustrate the information above better.

## Report (10%)

You also need to write a report in `pdf` type to explain the following:

- What are the possible solutions for the problem?
- How do you solve this problem?
- Why is your solution better than others?

Please note that the **maximum** number of pages allowed for your report is **5 pages**.

Remember that the report is to illustrate your thinking process. Keep in mind that your report is supposed to show your ideas and thinking process. We expect clear and precise textual descriptions in your report, and we do not recommend that you over-format your report.

# B. Example Problem: A + B Problem

## Description

Given 2 integers A and B, compute and print A + B

## Input

Two integers in one line: A, and B

## Output

One integer: A + B

## Sample Input 1

```
1 2
```

## Sample Output 1

```
3
```

### Problem Scale & Subtasks

For 100% of the test cases, $0 \leq A, B \leq 10^6$

### Solutions

#### Java

```java
import java.util.*;

public class Example {
    public static void main(String[] args) {
        int a, b;
        Scanner scanner = new Scanner(System.in);
        a = scanner.nextInt();
        b = scanner.nextInt();
        scanner.close();
        System.out.println(a + b);
    }
}
```

#### Python

```python
AB = input().split()
A, B = int(AB[0]), int(AB[1])
print(A + B)
```

#### C

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
  int A, B;
  scanf("%d%d", &A, &B);
  printf("%d\n", A + B);
  return 0;
}
```

#### C++

```cpp
#include <iostream>

int main(int argc, char *argv[])
{
  int A, B;
  std::cin>> A >> B;
  std::cout<< A + B << std::endl;
  return 0;
}
```

## C. Submission

After finishing this assignment, you are required to submit your code to the Online Judge System (OJ), and upload your `.zip` package of your code files and report to BlackBoard.

### C.1 Online Judge

Once you have completed one problem, you can submit your code on the page on the Online Judge platform (oj.cuhk.edu.cn, campus only) to gain marks for the code part. You can submit your solution of one problem for **no more than 80 times**.

After you have submitted your program, OJ will test your program on all test cases and give you a grade. The grade of your latest submission will be regarded as the final grade of the corresponding problem. Each problem is tested on multiple test cases of different difficulty. You will get a part of the score even if your algorithm is not the best.

**Note:** The program running time may vary on different machines. Please refer to the result of the online judge system. OJ will show the time and memory limits for different languages on the corresponding problem page.

If you have other questions about the online judge system, please refer to OJ wiki (campus network only). If this cannot help you, feel free to contact us.

### C.2 BlackBoard

You are required to upload your **source codes and report** to the BlackBoard platform. You need to name your files according to the following rules and compress them into `A1_<Student ID>.zip` :

```
A1_<Student ID>.zip
|-- A1_P1_<Student ID>.java/py/c/cpp
|-- A1_P2_<Student ID>.java/py/c/cpp
|-- A1_Report_<Student ID>.pdf
```

For Java users, **you don't need to consider the consistency of class name and file name**.

For example, suppose your ID is 123456789, and your problem 1 is written in `Python`, problem 2 is written in `Java` then the following contents should be included in your submitted `A1_123456789.zip`:

```
A1_123456789.zip
|-- A1_P1_123456789.py
|-- A1_P2_123456789.java
|-- A1_Report_123456789.pdf
```

### C.3 Late Submissions

Submissions after Sept. 29 2024 23:59:00(UTC+8) would be considered as LATE.

The LATE submission page will open after deadline on OJ.

Submisson time = max{latest submisson time for every problem, BlackBoard submisson time}

There will be penalties for late submission:

- 0–24 hours after deadline: final score = your score×0.8
- 24–72 hours after deadline: final score = your score×0.5
- 72+ hours after deadline: final score = your score×0

## FAQs

**Q:** My program passes samples on my computer, but not get AC on OJ.

**A:** Refer to OJ Wiki Q&A

# CSC3100 Data Structures Fall 2024

**Programming Assignment 1**

Yige Jiang: yigejiang@link.cuhk.edu.cn
Chunxu Lin: 221012033@link.cuhk.edu.cn

Due: Sept. 29 2024 23:59:00

**Assignment Link:** https://oj.cuhk.edu.cn/d/csc3100_2024_fall/homework/66e704de6605d3c4e7f63c35

## 1 Array Problem (40% of this assignment)

### 1.1 Description

You are given a sequence of integers $a_i$ of length $n$. Additionally, you are given $m$ operations to perform on this sequence. Each operation is one of the following:

- Given $k, x, y, c$, update the value of $a_k$ using the formula:

$$a_k = \left((x^2 + ky + 5x) \ mod \ P\right) * c$$

Obviously, the resulting value will be between $[1 - P, P - 1]$, where $c = \pm 1$.

- Query the sum of all elements in the sequence, i.e., compute:

$$\sum_{i=1}^{n} a_i$$

- Query the maximum number of distinct values in the sequence if each element is multiplied by either 1 or $-1$ (you can flip the sign of some elements and count the maximum number of distinct numbers).

Your task is to process these operations efficiently.

### 1.2 Input

The first line contains three integers $n$, $m$ and $P$ ($1 \leq n, m \leq 10^6, 1 \leq P \leq 10^6$) — the length of the sequence, the number of operations and the divisor in modulo operation, respectively.

The second line contains $n$ integers, representing the original value of the array $a$, denoted as $a_1, a_2, ..., a_n$ ($-P < a[i] < P$).

Each of the next $m$ lines contains a description of one of the following types of operations:

- For update operations, the line will contain five integers $1, k, x, y, c$ ($1 \leq k \leq n, 0 \leq x, y < \min(P, 2000), c \in \{-1, 1\}$).

- For sum queries, the line will contain a single integer 2.

- For distinct value queries, the line will contain a single integer 3.

### 1.3 Output

For each sum query, output the sum of all elements in the array.

For each distinct value query, output the maximum number of distinct values that can be obtained by multiplying each element by either 1 or $-1$.

### Sample Input 1

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

### Sample Output 1

```
3
3
4
```

### Sample Input 2

```
10 10 5
-1 -2 2 -3 2 0 -4 3 3 -3
3
1 2 4 4 -1
2
1 3 1 2 -1
3
2
3
1 2 4 4 1
3
1 4 4 0 -1
```

### Sample Output 2

```
7
-5
8
-9
8
8
```

### Sample Input 3

```
in 'array_sampleinput3.in'
```

### Sample Output 3

```
in 'array_sampleinput3.ans'
```

### Problem Scale & Subtasks

For about 60% test cases, distinct value queries are not evolved.

| Test Case No. | Constraints |
|---|---|
| 1-4 | $n, m \leq 20$ |
| 5-7 | $n, m \leq 5 \times 10^3$ |
| 8-10 | $n, m \leq 10^6$ |

### Hint

If you encounter a **TLE**, and your algorithm's time complexity is efficient, try optimizing your I/O operations.

## 2 List (50% of this assignment)

### 2.1 Description

Given an array, which is a permutation of size n (an array of size n where every integer from 1 to n appears exactly once), we perform q operations. During the i-th operation, we perform the following:

- Choose any subarray that contains at least 2 elements.

- Split it into two non-empty arrays.
- Obtain two integers $l_i$ and $r_i$, where $l_i$ is the **left most element** in the **left part** of the split, and $r_i$ is the **right most element** in the **right part** of the split.

For example, if the initial array is [6, 3, 4, 1, 2, 5], we perform the following operations:

1. Choose the array [6, 3, 4, 1, 2, 5] and split it into [6, 3] and [4, 1, 2, 5]. Then, $l_1 = 6$ and $r_1 = 5$.
2. Choose the array [4, 1, 2, 5] and split it into [4, 1, 2] and [5]. Then, $l_2 = 4$ and $r_2 = 5$, resulting in the arrays [6, 3], [4, 1, 2], and [5].
3. Choose the array [4, 1, 2] and split it into [4] and [1, 2]. Then, $l_3 = 4$ and $r_3 = 2$, resulting in the arrays [6, 3], [4], [1, 2], and [5].

**Objective.** Given two integers n and q, along with two sequences $[l_1, l_2, ..., l_q]$ and $[r_1, r_2, ..., r_q]$, a permutation is called valid if we can perform q operations and generate the given sequences $[l_1, l_2, ..., l_q]$ and $[r_1, r_2, ..., r_q]$.

**Problem**. Determine whether a given permutation with q operations is valid.

## 2.2 Input

1. The first line contains two integers n and q ($1 \leq q < n \leq 10^6$).
2. The second line contains a permutation of size n.
3. The third line contains q integers, $l_1, l_2, ..., l_q$ ($1 \leq l_i \leq n$).
4. The fourth line contains q integers, $r_1, r_2, ..., r_q$ ( $1 \leq r_i \leq n$ )

## 2.3 Output

Output 1 if the given permutation is valid, otherwise output 0.

### Sample Input 1

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

### Sample Output 1

```
1
```

### Sample Input 2

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

### Sample Output 2

```
0
```

### Sample Input 3

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

### Sample Output 3

```
0
```

## Problem Scale & Subtasks

For 100% of the test cases, $1 \leq q < n \leq 10^6$.

| Test Case No. | Constraints |
|---|---|
| 1-2 | $n \leq 10$ |
| 3-5 | $n \leq 10^3$ |
| 6-10 | $n \leq 10^6$ |

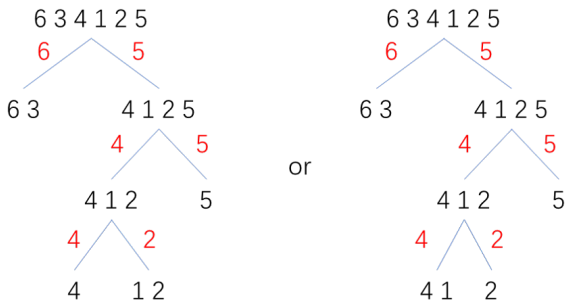


Figure 1: *Hint2*

## Hint

*Hint1*: For C/C++ and Java users, an `int` type stores integers range from -2,147,483,648 to 2,147,483,647. It may be too small for this problem. You need other data types, such as `long long` for C/C++ and `long` for Java. They store integers ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807. Use `scanf("%lld",&n)` for C, `cin>>n` for C++ and `n = scanner.nextLong()` for Java to get the input $n$. And the other operations for `long` and `long long` are quite same as `int`.

*Hint2*: The process of **Sample Input 1** can be described as in the Figure 1.

*Hint3*: This problem can be easily solved by following the above process from bottom to top.

*Hint4*: Consider how using the data structure, such as dictionary or list, to store the indices of $l_i$ and $r_i$ can help solve the problem.