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.

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`.

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 Mar 27, at 23:59 PM (UTC+8) would be considered as LATE. A late submission contest will open after deadline.

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: I cannot access to Online Judge.

A: First, please ensure that you are using the campus network. If you are not on campus, please use the university VPN. Second, please delete cookies and refresh browser or use other browser. If you still cannot access to Online Judge, try to visit it via the IP address `10.26.200.13`.

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

A: Refer to OJ Wiki Q&A

Authors

If you have questions for the problems below, please contact:

- Problem 1. Xuanzhuo Liu: xuanzhuoliu@link.cuhk.edu.cn
- Problem 2. Yingli Zhou: yinglizhou@link.cuhk.edu.cn
- Problem 3. Yohandi: yohandi@link.cuhk.edu.cn

CSC3100 Data Structures Spring 2024

Programming Assignment 2

Due: Mar 27 2024 23:59:00

Assignment Link: http://oj.cuhk.edu.cn/contest/csc310024spa2

Access code: Waaagh

Question 1, question 2, and question 3 weigh 40%, 25%, and 25% of the total grade, respectively.

Please note that you are not recommended to use AI tools such as chatGPT to complete your assignment for potential plagiarism issue.

1 Find the winner (40% of this assignment)

1.1 Description

Suppose you are playing a game with n people (labeled from 0 to n-1) and each person is allowed to leave a secret note to one or more people in this game. There may exist one winner. The definition of a winner is that all the other (n-1) people leave a note to him/her but he/she does not leave a note to any of them.

Now you want to figure out who the winner is or verify that there is not one. The only thing you are allowed to do is to ask questions like “Hello, A. Did you leave a note to B?” to get information of whether A leaves a note to B. You need to find out the winner by asking as few questions as possible.

You are given a square N*N matrix M[ ][ ] which is used to represent people in the game. If an element of row i and column j is set to 1 (M[i][j] = 1) it means ith person leaves a note to jth person. Otherwise it is set to 0. There will be exactly one winner or no winner in the game. Return the winner’s label if there is a winner in the game. If there is no winner, return -1.

1.2 Input

The first line of input contains an integer n, representing the number of players. The next n lines follow. The i-th line represents i-th player. If i-th player leaves a note to j-th player, the j-th entry is 1.

1.3 Output

For each test case, output the winner’s label if there is a winner in the game. If there is no winner, return -1.

Sample Input 1

```
4
0 0 1 0
0 0 1 0
0 0 0 0
0 0 1 0
```

Sample Output 1

```
2
```

Sample Input 2

```
4
0 0 1 0
0 0 1 0
0 1 0 0
0 0 1 0
```

Sample Output 2

```
-1
```

Problem Scale & Subtasks

There are 5 tests in total, with the same weight.

Test Case No.	Constraints
1	$1 \leq n \leq 5$
2-4	$1 \leq n \leq 100$
5-6	$1 \leq n \leq 1000$
7-8	$1 \leq n \leq 10^4$
9-10	$1 \leq n \leq 10^5$

Hint

You may try to find the winner in O(N) time.

2 Find out the tallest house (25% of this assignment)

2.1 Description

Imagine you're living in a small town called Arrayville, where every house is represented by a number that indicates its height. The town has a tradition: every year, they organize a parade. As a member of the parade committee, your task is to find out the tallest house visible from the float at each stop, to ensure the float's decorations do not obstruct the view of the spectators. Due to the float's width, at each stop, you can only see k houses located consecutively, including the house right in front of the float. Given an array *houses* representing the heights of the houses in the order they appear along the main street, and an integer k representing the width of the float, your challenge is to return an array that lists the tallest house visible from the float at each stop during the parade.

2.2 Input

The first line of input contains an integer n, representing the number of houses, and an integer k. The second line of input contains an array *houses* with length n.

2.3 Output

An array that lists the tallest house visible.

Sample Input 1

```
8 3
2 3 4 2 6 2 5 1
```

Sample Output 1

```
4
4
6
6
6
5
```

Sample Input 2

```
8 3
1 3 5 7 9 11 13 15
```

Sample Output 2

```
5
7
9
11
13
15
```

Sample Input 3

```
8 4
10 20 30 40 50 60 70 80
```

Sample Output 3

```
40
50
60
70
80
```

Problem Scale & Subtasks

There are 16 tests in total, with the same weight, we use n to denote the length of “houses”, and $a[i]$ to denote the element in “houses”.

Test Case No.	Constraints
1-3	$100 \leq a[i], n \leq 1000$, $1 \leq k \leq 50$
4-8	$10^3 \leq a[i], n \leq 10^4$, $250 \leq k \leq 3000$
9-12	$10^4 \leq a[i], n \leq 10^5$, $10^3 \leq k \leq 10^4$
13-16	$10^5 \leq a[i], n \leq 10^6$, $10 \leq k \leq 10^4$

3 Encrypting the Numeral Scroll (25% of this assignment)

3.1 Description

Being trusted by your master, you are to encrypt an artifact. This artifact, known as ”Numeral Scroll”, features a sequence of digits that forms the key to unlock a vault.

However, you know that the sequence has exactly n figures, whereas the input in the vault only accepts $n-m$ digits to be filled with. That’s when you realize there is another note in the artifact stating that ”create the smallest possible number without changing the original sequence’s order”. This means you must take out m digits away from the sequence to construct the smallest possible number.

3.2 Additional Note

If m is exactly n, you have to return 0 as the answer. Moreover, a sequence that has leading zero(s) is still considered a valid answer, for example, 05; however, you will still need to print it as 5 (without the leading zero(s)).

3.3 Input

You are given a sequence of n digits and an integer m.

3.4 Output

The smallest possible number after taking out m digits away from the given sequence.

Sample Input 1

```
1234 4
```

Sample Output 1

```
0
```

Sample Input 2

```
1942056387 5
```

Sample Output 2

```
5387
```

The smallest possible number is 05387; hence, 5387 is the answer.

Problem Scale & Subtasks

For all test cases:

- $1 \leq n \leq 1\ 000\ 000$
- $1 \leq m \leq n$

Test Case No.	Constraints
1-4	$n \leq 10$
5-6	$n \leq 10\ 000$, $m = 1$
7-8	$m = 1$
9-10	$m = n - 1$
11-14	All digits in the sequence are either 1 or 2.
15-20	No additional constraints.