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 `A2_<Student ID>.zip` :

```
A2_<Student ID>.zip
|-- A2_P1_<Student ID>.java/py/c/cpp
|-- A2_P2_<Student ID>.java/py/c/cpp
|-- A2_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 `A2_123456789.zip`:

```
A2_123456789.zip
|-- A2_P1_123456789.py
|-- A2_P2_123456789.java
|-- A2_Report_123456789.pdf
```

C.3 Late Submissions

Submissions after Oct. 24 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

Authors

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

- Problem 1. Chunxu Lin: 221012033@link.cuhk.edu.cn
- Problem 2. Zirong Zeng: zirongzeng@link.cuhk.edu.cn

CSC3100 Data Structures Fall 2024

Programming Assignment 2

Zirong Zeng: zirongzeng@link.cuhk.edu.cn
Chunxu Lin: 221012033@link.cuhk.edu.cn

Due: Oct. 24 2024 23:59:00

Assignment Link: https://oj.cuhk.edu.cn/d/csc3100_2024_fall/homework/6706a6d32579925ba8060510

1 Queue (40% of this assignment)

1.1 Description

Given an n-length list, the value a_i in the list satisfies that $a_i \in \{1, ..., n\}$ and $i = 1, ..., n$. Then, numbering each element a_i in the list as b_i, $b_i = i$.

Please find the minimum number of elements that should be deleted so that the list of remaining elements has the same elements as the list of their corresponding numbers.

For example, given a list $A = \{2, 1, 2, 2, 4\}$, and its number list $N = \{1, 2, 3, 4, 5\}$. We need to delete the 3-rd and 5-th elements from lists A and N because numbers 3 and 5 are not in the list A. Then, the list $A = \{2, 1, 2\}$ and its number $N = \{1, 2, 4\}$. However, since 4 is not in the list A, we need to delete the 3-rd element from lists A and N. Finally, we get the list $A = \{2, 1\}$ and its number list $N = \{1, 2\}$, where the list $A = \{2, 1\}$ has the same elements as its number list $N = \{1, 2\}$. Conclusively, we delete 3 elements from the list A.

1.2 Input

- The first line contains a positive integer n.
- The second line contains the n-length list.

1.3 Output

- The minimum number of deleted elements.

Sample Input 1

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

Sample Output 1

```
3
```

Sample Input 2

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

Sample Output 2

```
0
```

For a list $A = \{1, 3, 4, 2, 5\}$, its number list is $N = \{1, 2, 3, 4, 5\}$. We do not need to delete any element from lists A and N since the list A has the same elements as its number list N.

Sample Input 3

```
See attached sample.in
```

Sample Output 3

```
See attached sample.out
```

Problem Scale & Subtasks

For 100% of the test cases, $1 \leq n \leq 2 * 10^5$.

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

Hint

Hint1: You can use a queue to store elements that are not in a given list but are in its list of numbers.

2 Time Complexity (50% of this assignment)

2.1 Description

John is learning a new programming language called A++. Having just mastered loops, he is excitedly writing many programs that contain only loop structures. However, he does not know how to compute the time complexity of his programs. So he turns to you for help. What you need to do is to write a program to calculate the time complexity for each program that John writes.

The loop structure in A++ is as follows:

```
F i x y
    ... // code block to be executed
E
```

Here "F i x y" indicates creating a variable i initialized as x, then compare i with y, and enter the loop if and only if i is smaller or equal to y. Each time the execution of the code block inside the loop ends, i will be changed to $i + 1$. Then i will be compared with y to determine whether to enter the loop again. The loop ends when i becomes larger than y.

Both x and y can be positive integers or the variable n. n represents the size of the data and must be retained during time complexity calculations. It cannot be treated as a constant. The value of n is much greater than 100. x is guaranteed to be smaller or equal to y.

"E" indicates the end of the loop. When the loop ends, any variables created within that loop are also destroyed.

Note: For the sake of convenience in writing, the uppercase letter O is used to represent the standard notion of Θ when describing time complexity in this problem.

2.2 Input

The first line contains a positive integer t, indicating that John writes t ($1 \leq t \leq 10$) programs in total (the programs contain only loop structures). Note that the loop structures are allowed to be nested. The following lines describe the t programs in order.

For each program, the first line contains a positive integer L ($1 \leq L \leq 20000$), indicating the number of lines in this program. The next L lines describe the program in detail: each line is either in the form of "F i x y" or "E", where i is a lowercase letter (it is guaranteed that i will not be letter "n"), x and

y are either variable n or positive integers smaller than 100, and x is guaranteed to be smaller or equal to y. Since John has mastered loops quite well, there will be no syntax errors in his programs. i.e. F and E are guaranteed to match each other and he will not use variables that haven't been destroyed in a loop.

2.3 Output

Output t lines, each indicating the time complexity of a program, in the order of the input. In this problem, time complexity is either in the form of O(1) or O(n^w), where w is a positive integer smaller than 20000. O(1) means constant time complexity and O(n^w) means that the time complexity is $O(n^w)$

Sample Input 1

```
6
2
F i 1 1
E
2
F x 1 n
E
4
F x 5 n
F y 10 n
E
E
4
F x 9 n
E
F y 2 n
E
4
F x 9 n
F y n n
E
E
4
F y 1 99
F x n n
E
E
```

Sample Output 1

```
O(1)
O(n^1)
O(n^2)
O(n^1)
O(n^1)
O(1)
```

Problem Scale & Subtasks

Test Case No.	Constraints	Properties
1-3	$L \leq 10$	A
4-6	$L \leq 500$	
7-10	$L \leq 20000$	

A: For each program, the first $L/2$ lines are in the form of "F i x y", and the lines from $L/2+1$ to L are in the form of "E".