National University of Computer and Emerging Sciences, Lahore Campus

Course:	Data Structures	Course Code: CS 2001

Program:	**BS(CS)**	**Semester:**	**FALL-2025**
Due Date:	**Friday Oct 3,2025**	**Total Marks:**	**150**
Type:	**Assignment 2**	**Page(s):**	**5**
Course Instructor	**Rana Waqas Ali**	**TA.**	**Syed Aoun Haider Sherazi**

Important Instructions:

1. Submit your .cpp files named as your roll number+QuestionNo+FileNo. Dont submit zip files.
2. You are not allowed to copy solutions from other students. We will check your code for plagiarism using plagiarism checkers. If any sort of cheating is found, negative marks will be given to all students involved.
3. Late submission of your solution is not allowed.
4. All questions / parts must be done keeping in mind the time constraint of that Data Structures or marks will be deducted.
5. In case of any confusion, feel free to discuss with Instructor or TA.
6. Do proper edge cases handling for all questions.

Question 1 [50 Marks]

Design a C++ program that uses the stack class for matching tags and quotes in XML extensive Markup Language. Your program will get an XML code in an input file, and it should figure out if tags and quotes are properly matched or not using stack. In case the tags are not matched, then your program should **report i) the error, ii) print the mismatched tag and iii) print the line number at which the starting tag occurred. And then continue parsing the input XML file till the XML file ends.**

What is XML? XML is a markup language somewhat like HTML. It was designed to store and transport data. XML is just information wrapped in user-defined tags which is both human- and machine-readable. The XML language has no predefined tags like HTML. The tags are "invented" by the author of the XML document. For details, see https://www.w3schools.com/xml/xml_whatis.asp

National University of Computer and Emerging Sciences, Lahore Campus

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<message>Don't forget me this weekend!</message>
</note>
```

In above example

<?xml version="1.0" encoding="UTF-8"?>

This is XML prolog (header). It starts with <? and ends with ?>. The header should come in the start of the document.

<note>, <from>, <heading> and <message> are user defined tags and each must have a corresponding ending tag.

Your program should handle the following features of XML:

1. xml prolog (xml header)
2. xml tags (xml elements). The xml tags are case-sensitive.
3. xml attribute
4. xml comments, start with <!-- and ends with -->

National University of Computer and Emerging Sciences, Lahore Campus

Consider another example

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
<book category="children">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
</bookstore>
```

In the example above:

<title>, <author>, <year>, and <price> have **text content** because they contain text (like 2005).
<bookstore> and <book> have **element contents**, because they contain elements.
<book> has an **attribute** (category="children").

Note that Attribute values must always be quoted. Either single or double quotes can be used.

Your program should keep track that the attributes have opening and closing quotes.

Your code will have a template-based Node class Stack class. Implement stack using singly linked list.

```
template<class T>
class Stack {
private:

class Node {
       T  data;
      Node<T> * next;
      Node<T> * top;

 };
public:
      Stack() ;
      ~Stack();
      bool IsEmpty();
      bool push(const T & val);
      bool pop(T & val);
};    bool top(T & val);
```

Create XMLData class
Think about its attributes carefully

```
void main(){

Stack<XMLData> S1;
```

National University of Computer and Emerging Sciences, Lahore Campus

Note: (Not Mandatory , it's your choice)

You can also made simple functions in XMLData class and call them in main() as:

```
XMLData d1;

d1.checkAttribute();
```

BUT for that case your XML class must inherit your Stack class.

Question 2 [20 marks]

('Invertible' Stack)

Add the following methodology to the Stack Class (You must create your own Stack Class).

Add a method called flipStack. This method should work in O(1). Its effect should be such that the whole stack should be logically inverted, i.e. the oldest element becomes the newest and vice versa. So the next pop will remove the element that was at the bottom of the stack before it had been flipped. Notice that the stack may be flipped again and again by using the flipStack method repeatedly. Make sure that no slot of the array is wasted (Hint: Circular). As always, when the stack fills up we double its capacity; and when more than half of it is empty, we halve the capacity. Once again, test your code extensively.

Question 3 [40 marks]

Josephus Problem

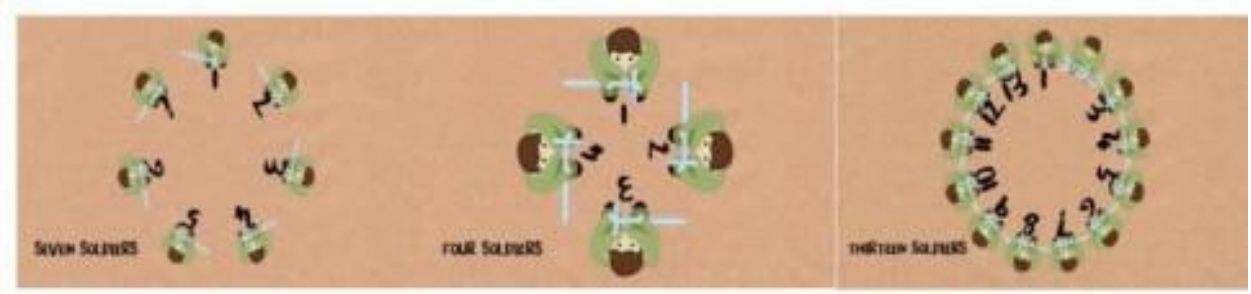
History: This problem is named after Flavius Josephus, a Jewish historian living in the 1st century. According to Josephus' account of the siege of Yodfat, he and his 40 soldiers were trapped in a cave byRoman soldiers. They chose suicide over capture and settled on a serial method of committing suicide by drawing lots. Josephus states that by luck or possibly by the hand of God, he and another man remained until the end and surrendered to the Romans rather than killing themselves. This is the story given in Book 3, Chapter 8, part 7 of Josephus' The Jewish War (writing of himself in the third person)

In computer science and mathematics, the Josephus problem (or Josephus permutation) is a theoretical problem related to a certain counting-out game.

National University of Computer and Emerging Sciences, Lahore Campus

https://en.wikipedia.org/wiki/Josephus_problem **1.** People (any number N >1) are standing in a circle waiting to be executed.

2 . Counting begins at a specified point (S selected randomly)in the circle and proceeds around the circle in a specified direction. You can assume that it is clockwise.



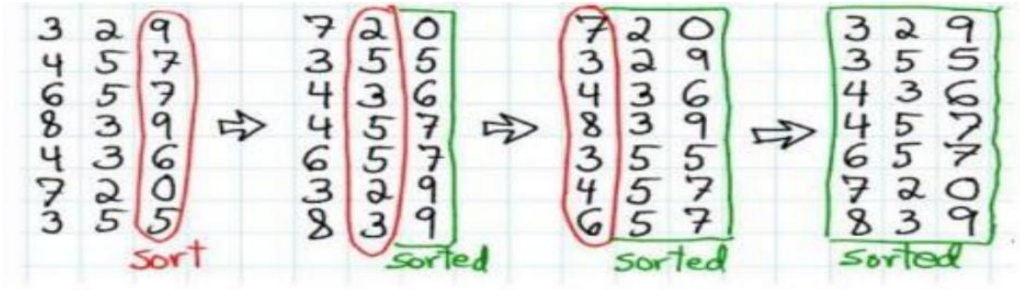
3. After a specified number of people are skipped, say (k-1), the next (kth) person is executed. The procedure is repeated with the remaining people, starting with the next person, going in the same direction and skipping the same number of people(k-1), until only one person remains, and is freed.

Write a C++ program that solves the Josephus problem by using a Queue data structure. Your program should take N and K input from the user. Design a graphical user interface that will update with every move as shown in figure above.

Question 4 [40 marks]

Problem 2: (Radix Sort)

Radix sort is a sorting algorithm, which sorts the keys based on the values of digits in keys. It takes a queue containing **n** keys to be sorted, where each key consists of **k**number of digits, and there could be **m** possible values for each digit **0** through **m-1**. Radix sort uses an array consisting of (**m**) queues for sorting of these keys. For example, if each key contains **k** = 3 digits and each individual digit has **m**=10 possible values 0 to 9, then it will use an array consisting of 10 queues 0-9in the sorting process of considering all digits one by one as follows.



You have to implement two Radix sorts which can sort **integer** and **string** keys with any value of **n** and **k**.

National University of Computer and Emerging Sciences, Lahore Campus