

# **C++ STANDARD TEMPLATE LIBRARY**

## **LAB MANUAL**

**Course Code** : **ACSB06**  
**Regulations** : **IARE - R18**  
**Semester** : **III**  
**Branch** : **CSE / IT**

**Prepared by**

**Ms. S Swarajya Laxmi, Assistant Professor**  
**Ms. B Ramya Sree, Assistant Professor**  
**Mr. N V Krishna Rao, Assistant Professor**  
**Mr. R M Noorullah, Assistant Professor**  
**Mr. G. Sunil Reddy, Assistant Professor**  
**Ms. K Laxmi Narayanamma, Assistant Professor**



**INSTITUTE OF AERONAUTICAL ENGINEERING**  
**(Autonomous)**  
**Dundigal, Hyderabad - 500 043**



**INSTITUTE OF AERONAUTICAL ENGINEERING**  
(Autonomous)  
Dundigal, Hyderabad - 500 043

**1. PROGRAM OUTCOMES:**

<b>PROGRAM OUTCOMES (POs)</b>	
<b>PO-1:</b>	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems ( <b>Engineering Knowledge</b> ).
<b>PO-2:</b>	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences ( <b>Problem Analysis</b> ).
<b>PO-3:</b>	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations ( <b>Design/Development of Solutions</b> ).
<b>PO-4:</b>	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions ( <b>Conduct Investigations of Complex Problems</b> ).
<b>PO-5:</b>	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations ( <b>Modern Tool Usage</b> ).
<b>PO-6:</b>	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice ( <b>The Engineer and Society</b> ).
<b>PO-7:</b>	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development ( <b>Environment and Sustainability</b> ).
<b>PO-8:</b>	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice ( <b>Ethics</b> ).
<b>PO-9:</b>	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings ( <b>Individual and Team Work</b> ).
<b>PO-10:</b>	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions ( <b>Communication</b> ).
<b>PO-11:</b>	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. ( <b>Project management and finance</b> ).

<b>PROGRAM OUTCOMES (POs)</b>	
<b>PO-12:</b>	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change ( <b>Life-long learning</b> ).

## **2. PROGRAM SPECIFIC OUTCOMES (PSOs):**

<b>PROGRAM SPECIFIC OUTCOMES (PSO'S)</b>	
<b>PSO – I</b>	<b>Professional Skills:</b> The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer - based systems of varying complexity.
<b>PSO – II</b>	<b>Software Engineering Practices:</b> The ability to apply standard practices and strategies in software service management using open-ended programming environments with agility to deliver a quality service for business success.
<b>PSO –III</b>	<b>Successful Career and Entrepreneurship:</b> The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

## **3. ATTAINMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:**

<b>Week No</b>	<b>Experiment</b>	<b>Program Outcomes Attained</b>	<b>Program Specific Outcomes Attained</b>
<b>1</b>	<b>CONTROL STRUCTURES</b>	PO1, PO2, PO5	PSO2
<b>2</b>	<b>VECTORS AND MAPS</b>	PO1, PO2, PO5	PSO2
<b>3</b>	<b>STACK AND QUEUE</b>	PO1, PO2, PO5	PSO2
<b>4</b>	<b>SETS AND STRINGS</b>	PO1, PO2, PO5	PSO2
<b>5</b>	<b>SORTINGS AND PAIRS</b>	PO1, PO2, PO5	PSO2
<b>6</b>	<b>ARRAYS AND LISTS</b>	PO1, PO2, PO3, PO4, PO12	PSO1, PSO2
<b>7</b>	<b>MULTISET AND MULTIMAPS</b>	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
<b>8</b>	<b>UNORDERED SETS</b>	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
<b>9</b>	<b>SET UNION AND INTERSECTION</b>	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2

10	<b>IMPLEMENTATION OF QUEUE USING LINKED LIST</b>	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
11	<b>PERMUTATIONS</b>	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2
12	<b>LEXICOGRAPHICAL</b>	PO1, PO2, PO3, PO4, PO5, PO12	PSO1, PSO2

**4. MAPPING COURSE OBJECTIVES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES:**

Course Objectives (COs)	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>I</b>	√	√	√										√		
<b>II</b>	√	√		√	√								√		
<b>III</b>			√		√							√	√	√	
<b>IV</b>	√	√	√									√		√	

## 5. SYLLABUS:

### C++ STANDARD TEMPLATE LIBRARY

VII Semester: CSE/IT								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
ACSB06	Core	L	T	P	C	CIA	SEE	Total
		0	0	3	1.5	30	70	100
<b>Contact Classes: Nil</b>	<b>Tutorial Classes: Nil</b>	<b>Practical Classes: 36</b>			<b>Total Classes: 36</b>			
<b>COURSE OBJECTIVES:</b> <b>The course should enable the students to:</b> I. Understand how C++ STL improves C with predefined libraries II. Learn how to implement C++ standard Template Libraries. III. Understand the concept of vectors, maps, stacks, queues and many more <b>COURSE LEARNING OUTCOMES (CLOs):</b> 1. To understand how C++ improves C with object-oriented features. 2. To learn how to write inline functions for efficiency and performance. 3. To learn the syntax and semantics of the C++ programming language. 4. To learn how to design C++ classes for code reuse. 5. To learn how to implement copy constructors and class member functions. 6. To understand the concept of data abstraction and encapsulation. 7. To learn how to overload functions and operators in C++. 8. To learn how containment and inheritance promote code reuse in C++. 9. To learn how inheritance and virtual functions implement dynamic binding with polymorphism. 10. To learn how to design and implement generic classes with C++ templates. 11. To learn how to use exception handling in C++ programs.								
<b>Week-1</b>	<b>CONTROL STRUCTURES</b>							
a. In this problem, you need to print the pattern of the following form containing the numbers from 1 to n. <pre> 4 4 4 4 4 4 4 3 3 3 3 4 4 3 2 2 2 3 4 4 3 2 1 2 3 4 4 3 2 2 2 3 4 4 3 3 3 3 3 4 4 4 4 4 4 4 4           </pre> b. Given a positive integer denoting n, do the following: If $1 \leq n \leq 9$ , then print the lowercase English word corresponding to the number (e.g., one for 1, two for 2, etc.). If $n > 9$ , print Greater than 9.								
<b>Week-2</b>	<b>VECTORS AND MAPS</b>							
a. A left rotation operation on a vector of size N shifts each of the array's elements 1 unit to the left. For example, if 2 left rotations are performed on array [1,2,3,4,5], then the array would become [3,4,5,1,3]. Given an vector of n integers and a number, d, perform d left rotations on the array. Then print the updated array as a single line of space-separated integers. Print a single line of n space-separated integers denoting the final state of the array after performing d left rotations.								

<p>b. Prasad is working as teacher in one school. He evaluated exam papers for all students. He decided to store their marks in his computer using their names. Can you please suggest best data structure . For example  Marks[Ramu]=98  Marks[Janu]=87</p>	
<b>Week-3</b>	<b>STACK AND QUEUE</b>
<p>a. You have an empty sequence, and you will be given queries. Each query is one of these three types:  1 x -Push the element x into the stack.  2 -Delete the element present at the top of the stack.  3 -Print the maximum element in the stack. For each type 3 query, print the maximum element in the stack on a new line.</p> <p>b. You must first implement a queue using two stacks. Then process queries, where each query is one of the 3 following types:  1 x: Enqueue element into the end of the queue.  2: Dequeue the element at the front of the queue.  3: Print the element at the front of the queue. For each query of type , print the value of the element at the front of the queue on a new line.</p>	
<b>Week-4</b>	<b>SETS AND STRINGS</b>
<p>a. You will be given Q queries. Each query is of one of the following three types:  1. x : Add an element x to the set.  2. x : Delete an element x from the set. (If the number is not present in the set, then do nothing).  3. x : If the number x is present in the set, then print "Yes"(without quotes) else print "No"(without quotes). For queries of type 3 print "Yes"(without quotes) if the number x is present in the set and if the number is not present, then print "No"(without quotes). Each query of type 3 should be printed in a new line.</p> <p>b. You are given a string containing characters A and B only. Your task is to change it into a string such that there are no matching adjacent characters. To do this, you are allowed to delete zero or more characters in the string. Your task is to find the minimum number of required deletions. For example, given the string s=AABAAB, remove an A at positions 0 and 3 to make s=ABAB in 2 deletions.</p>	
<b>Week-5</b>	<b>SORTINGS AND PAIRS</b>
<p>a. Raju and Ravi are friends. Raju asked Ravi to arrange the set of string in ascending order (Dictionary format). Please help the Ravi to put the strings in ascending order.</p> <p>b. Teacher given a task to students find the unvisited elements in the given matrix. The students are struggling to find the unvisited elements in the list. Please help them to solve.</p>	
<b>Week-6</b>	<b>ARRAYS AND LISTS</b>
<p>All friends are invited and they arrive at the party one by one in an arbitrary order. However, they have certain conditions for each valid i, when the i-th friend arrives at the party and sees that at that point, strictly less than Ai other people (excluding Chef) have joined the party, this friend leaves the party; otherwise, this friend joins the party. Help Chef estimate how successful the party can be find the maximum number of his friends who could join the party (for an optimal choice of the order of arrivals).</p>	
<b>Week-7</b>	<b>MULTISET AND MULTIMAPS</b>
<p>Kattapa, as you all know was one of the greatest warriors of his time. The kingdom of Maahishmati had never lost a battle under him (as army-chief), and the reason for that was their really powerful army, also called as Mahasena. Kattapa was known to be a very superstitious person. He believed that a soldier is "lucky" if the soldier is holding an even number of weapons, and "unlucky" otherwise. He considered the army as "READY FOR BATTLE" if the count of "lucky" soldiers is strictly greater than the count of "unlucky" soldiers, and "NOT READY" otherwise. Given the number of weapons each soldier is holding, your task is to determine whether the army formed by all these soldiers is "READY FOR BATTLE" or "NOT READY".</p>	
<b>Week-8</b>	<b>UNORDERED SETS</b>

You are given two lists of N distinct numbers. Sort both the list and print them alternatively starting with list one	
<b>Week-9</b>	<b>SET UNION AND INTERSECTION</b>
A class contains two subjects and students can take one or two subjects as there wish. Here, students opted subjects on there own interest. Now, your task is to print all the total students count and students names, and also print how many took two subjects and their names.	
<b>Week-10</b>	<b>IMPLEMENTATION OF QUEUE USING LINKED LIST</b>
A class contains two subjects and students can take one or two subjects as there wish. Here, students opted subjects on their own interest. Now your task to find the student names who are attending first subject but not second and vice versa.	
<b>Week-11</b>	<b>PERMUTATIONS</b>
IARE college has designed a new challenge called BuildIT Competitive Programming. In this game, each team contains N members and they are specialised in either Java Programming or Python Programming. The challenge contains n1 java questions and n2 Python questions. So, team members are decided to seat in all specialized members as one group. So that, number of ways the N members seat in the programming contest. For example: a team contains 'ab' java programmers and 'cde' python programmers (a, b) (c, d, e) (b,a) (c, e, d) (d, c, e) (d, e, c) (e, c, d) (e, d, c) So, total ways are = 12	
<b>Week-12</b>	<b>LEXICOGRAPHICAL</b>
Ravi and Raju are best friends. Ravi given a set of strings to Raju and ask him to find smaller string as per lexicographical order. Please help him to find.	
<b>Reference Books:</b>	
1. Bjarne Stroustrup , "Programming: Principles and Practice Using C++" 2 <sup>nd</sup> Edition, 2014. 2. Herbert Schildt, "C++: The Complete Reference, 4 <sup>th</sup> Edition, 2017.	

## 6. INDEX:

<b>S.NO</b>	<b>LIST OF EXPERIMENTS</b>	<b>PAGE NO</b>
	<b>CONTROL STRUCTURES</b>	
<b>WEEK-1</b>	In this problem, you need to print the pattern of the following form containing the numbers from 1 to n. 4 4 4 4 4 4 4 3 3 3 3 4 4 3 2 2 2 3 4 4 3 2 1 2 3 4 4 3 2 2 2 3 4 4 3 3 3 3 4 4 4 4 4 4 4	10

	b.	Given a positive integer denoting n, do the following: If $1 \leq n \leq 9$ , then print the lowercase English word corresponding to the number (e.g., one for, two for, etc.). If $n > 9$ , print Greater than 9.	
<b>WEEK-2</b>	<b>VECTORS AND MAPS</b>		
	a.	A left rotation operation on a vector of size N shifts each of the array's elements 1 unit to the left. For example, if 2 left rotations are performed on array [1,2,3,4,5], then the array would become[3,4,5,1,3] Given an vector of n integers and a number, d , perform d left rotations on the array. Then print the updated array as a single line of space-separated integers. Print a single line of n space-separated integers denoting the final state of the array after performing d left rotations.	17
	b.	Prasad is working as teacher in one school. He evaluated exam papers for all students. He decided to store their marks in his computer using their names. Can you please suggest best data structure . For example Marks["Ramu"]=98 Marks["Janu"]=87	
<b>WEEK-3</b>	<b>STACK AND QUEUE</b>		
	a.	You have an empty sequence, and you will be given queries. Each query is one of these three types: 1 x -Push the element x into the stack. 2 -Delete the element present at the top of the stack. 3 -Print the maximum element in the stack. For each type 3 query, print the maximum element in the stack on a new line.	24
	b.	You must first implement a queue using two stacks. Then process queries, where each query is one of the 3 following types: 1 x: Enqueue element into the end of the queue. 2: Dequeue the element at the front of the queue. 3: Print the element at the front of the queue. For each query of type , print the value of the element at the front of the queue on a new line	
<b>WEEK-4</b>	<b>SETS AND STRINGS</b>		
	a.	You will be given Q queries. Each query is of one of the following three types: 1. x : Add an element x to the set. 2. x : Delete an element x from the set. (If the number is not present in the set, then do nothing). 3. x : If the number x is present in the set, then print "Yes"(without quotes) else print "No"(without quotes). For queries of type 3 print "Yes"(without quotes) if the number x is present in the set and if the number is not present, then print "No"(without quotes). Each query of type 3 should be printed in a new line.	26
	b.	You are given a string containing characters A and B only. Your task is to change it into a string such that there are no matching adjacent characters. To do this, you are allowed to delete zero or more characters in the string. Your task is to find the minimum number of required deletions.	



	<b>SORTINGS AND PAIRS</b>		
<b>WEEK-5</b>	a	Raju and Ravi are friends. Raju asked Ravi to arrange the set of string in ascending order (Dictionary format). Please help the Ravi to put the strings in ascending order.	28
	b	Teacher given a task to students find the unvisited elements in the given matrix. The students are struggling to find the unvisited elements in the list. Please help them to solve.	
	<b>ARRAYS AND LISTS</b>		
<b>WEEK-6</b>	All friends are invited and they arrive at the party one by one in an arbitrary order. However, they have certain conditions for each valid $i$ , when the $i$ -th friend arrives at the party and sees that at that point, strictly less than $A_i$ other people (excluding Chef) have joined the party, this friend leaves the party; otherwise, this friend joins the party. Help Chef estimate how successful the party can be find the maximum number of his friends who could join the party (for an optimal choice of the order of arrivals).		31
	<b>MULTISET AND MULTIMAPS</b>		
<b>WEEK-7</b>	Kattapa, as you all know was one of the greatest warriors of his time. The kingdom of Maahishmati had never lost a battle under him (as army-chief), and the reason for that was their really powerful army, also called as Mahasena. Kattapa was known to be a very superstitious person. He believed that a soldier is "lucky" if the soldier is holding an even number of weapons, and "unlucky" otherwise. He considered the army as "READY FOR BATTLE" if the count of "lucky" soldiers is strictly greater than the count of "unlucky" soldiers, and "NOT READY" otherwise. Given the number of weapons each soldier is holding, your task is to determine whether the army formed by all these soldiers is "READY FOR BATTLE" or "NOT READY".		34
	<b>UNORDERED SETS</b>		
<b>WEEK-8</b>	a.	You are given two lists of $N$ distinct numbers. Sort both the list and print them alternatively starting with list one	37
	<b>SET UNION AND INTERSECTION</b>		
<b>WEEK-9</b>	a.	A class contains two subjects and students can take one or two subjects as there wish. Here, students opted subjects on there own interest. Now, your task is to print all the total students count and students names, and also print how many took two subjects and their names.	39
	<b>IMPLEMENTATION OF QUEUE USING LINKED LIST</b>		
<b>WEEK-10</b>	a.	A class contains two subjects and students can take one or two subjects as there wish. Here, students opted subjects on their own interest. Now your task to find the student names who are attending first subject but not second and vice versa.	41
	<b>PERMUTATIONS</b>		
<b>WEEK-11</b>	a.	IARE college has designed a new challenge called BuildIT Competitive Programming. In this game, each team contains $N$ members and they are specialised in either Java Programming or Python Programming.	43

		<p>The challenge contains n1 java questions and n2 Python questions. So, team members are decided to seat in all specialized members as one group. So that, number of ways the N members seat in the programming contest. For example: a team contains 'ab' java programmers and 'cde' python programmers</p> <p>(a, b) (c, d, e)  (b,a) (c, e, d)  (d, c, e)  (d, e, c)  (e, c, d)  (e, d, c)</p> <p>So, total ways are = 12</p>	
	<b>LEXICOGRAPHICAL</b>		
<b>WEEK-12</b>	a.	Ravi and Raju are best friends. Ravi given a set of strings to Raju and ask him to find smaller string as per lexicographical order. Please help him to find.	45

## WEEK-1 CONTROL STRUCTURES

### 1.1 OBJECTIVE:

- a. To print the pattern of the following form containing the numbers from 1 to n.

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

- b. Given a positive integer denoting n, do the following:  
If  $1 \leq n \leq 9$ , then print the lowercase English word corresponding to the number (e.g., one for, two for, etc.). If  $n > 9$ , print Greater than 9.

### 1.2 RESOURCES:

Dev C++

### 1.3 PROGRAM LOGIC:

- a) To print the pattern

```
#include<iostream>
using namespace std;
int main()
{
    int n,min,max,i,j,a[50][50],order;
    cout<<"enter n value : ";
    cin>>n;
    order=2*n-1;
    min=0,max=2*n-1;
    while(n>0)
    {
        for(i=min;i<max;i++)
        {
            for(j=min;j<max;j++)
```

```

        {
            if(i==min||i==max-1||j==min||j==max-1)
                a[i][j]=n;
        }
    }
    n--;
    min++;
    max--;
}
for(i=0;i<order;i++)
{
    for(j=0;j<order;j++)
    {
        cout<<a[i][j]<<" ";
    }
    cout<<"\n";
}
return 0;
}

```

**b) Program to print the number names**

```

#include<iostream>
using namespace std;
int main()
{
    int num;
    cout<<"enter a number:";
    cin>>num;
    switch(num)

    {
        case 1:cout<<"one";break;
        case 2:cout<<"two";break;
        case 3:cout<<"three";break;
        case 4:cout<<"four";break;
        case 5:cout<<"five";break;
        case 6:cout<<"six";break;
        case 7:cout<<"SEVEN";break;
        case 8:cout<<"eight";break;
        case 9:cout<<"nine";break;
        default:cout<<"number is greater than 9";
        break;
    }
    return 0;
}

```

**1.4 INPUT/OUTPUT**

**a) To print the pattern**

enter n value : 3

3 3 3 3  
3 2 2 3  
3 2 1 2 3  
3 2 2 2 3  
3 3 3 3 3

**b) Program to print the number names**

Enter a number: 4

Four

**1.5 PRE LAB VIVA QUESTIONS:**

1. What is the classification of control structures?
2. What is the syntax of for loop?
3. What is the difference between for loop and while loop?

**1.6 LAB ASSIGNMENT:**

1. Write a C++ program to Concatenate Two Strings
2. Write a C++ program to Remove all Characters in a String Except Alphabets.
3. Write a C++ program to Find Largest Element of an Array

**1.7 POST LAB VIVA QUESTIONS:**

1. What is conditional controls?
2. What is the syntax of break statement?
3. What are jumping statements?

## WEEK - 2 VECTORS AND MAPS

### 2.1 OBJECTIVE:

- a. A left rotation operation on a vector of size N shifts each of the array's elements 1 unit to the left. For example, if 2 left rotations are performed on array [1,2,3,4,5], then the array would become[3,4,5,1,3 . Given an vector of n integers and a number, d , perform d left rotations on the array. Then print the updated array as a single line of space-separated integers. Print a single line of n space-separated integers denoting the final state of the array after performing d left rotations.
- b. Prasad is working as teacher in one school. He evaluated exam papers for all students. He decided to store their marks in his computer using their names. Can you please suggest best data structure.

### 2.2 RESOURCES:

Dev C++

### 2.3 PROGRAM LOGIC:

#### a) Left rotation operation:

```
#include<iostream>
using namespace std;
int main()
{
    int a[100],i,n,d,y=1;
    cout<<"enter no.of elements";
    cin>>n;
    cout<<"enetr no.of rotations";
    cin>>d;
    for(i=1;i<n;i++)
    {
        cout<<i;
    }
    for(i=1;i<n;i++)
    {
        if(d<n)
        {
            a[i]=d+1;
            d++;
        }
        else
        {
            a[i]=y;
            y++;
        }
    }
    for(i=1;i<n;i++)
```

```

    {
        cout<<a[i];
    }
    return 0;
}

```

**c) To store their marks in his computer using their names:**

```

#include<iostream>
using namespace std;
class student
{
    public:
    char name[50];
    int marks;
}
s[10];
int main()
{
    int i,n;
    cout<<"enter n value:";
    cin>>n;
    for(i=0;i<n;i++)
    {
        cout<<"enter name:";
        cin>>s[i].name;
        cout<<"enter marks:";
        cin>>s[i].marks;
    }
    for(i=0;i<n;i++)
    {
        cout<<"name:"<<s[i].name<<endl;
        cout<<"marks:"<<s[i].marks<<endl;
        return 0;
    }
}

```

## 2.4 INPUT/OUTPUT:

**a) Left rotation operation:**

```

enter no.of elements :6
enetr no.of rotations :4
1234556123

```

**b) To store their marks in his computer using their names:**

```

enter n value: 2
enter name: swarajya
enter marks: 25
enter name: laxmi

```

enter marks: 24  
name: swarajya  
marks: 25  
name: laxmi  
marks: 24

**2.5 PRE LAB VIVA QUESTIONS:**

1. What are vectors used for in C++?
2. What are maps in C++?
3. What is the use of standard template library in C++?

**2.6 LAB ASSIGNMENT:**

1. Write a program to Reverse a Sentence Using Recursion
2. Write a Program to Find G.C.D Using Recursion
3. Write a Program to Check Whether a Number can be Express as Sum of Two Prime Numbers

**2.7 POST LAB VIVA QUESTIONS:**

1. How STL is different from the C++ Standard Library
2. What are vectors used for in C++?
3. What are maps in C++?



USING LINUX OPERATING SYSTEM

3.1 OBJECTIVE:

- a. You have an empty sequence, and you will be given queries. Each query is one of these three types:
  - i. Push the element x into the stack.
  - ii. Delete the element present at the top of the stack.
  - iii. Print the maximum element in the stack. For each type 3 query, print the maximum element in the stack on a new line.
  
- b. You must first implement a queue using two stacks. Then process queries, where each query is one of the 3 following types:
  - i. Enqueue element into the end of the queue.
  - ii. Dequeue the element at the front of the queue.
  - iii. Print the element at the front of the queue. For each query of type , print the value of the element at the front of the queue on a new line.

3.2 RESOURCES:

Dev C++

3.3 PROGRAM LOGIC:

a) Stack operations

```
#include <iostream>
using namespace std;
int stack[100], n=3, top=-1;
void push(int val) {
    if(top>=n-1)
        cout<<"Stack Overflow"<<endl;
    else {
        top++;
        stack[top]=val;
    }
}
void pop() {
    if(top<=0)
        cout<<"Stack Underflow"<<endl;
    else {
        cout<<"The popped element is "<< stack[top] <<endl;
        top--;
    }
}
void display() {
    if(top>=0) {
        cout<<"Stack elements are:";
```

```

        for(int i=top; i>=0; i--)
            cout<<stack[i]<<" ";
        cout<<endl;
    } else
        cout<<"Stack is empty";
}
int main() {
    int ch, val;
    cout<<"1) Push in stack"<<endl;
    cout<<"2) Pop from stack"<<endl;
    cout<<"3) Display stack"<<endl;
    cout<<"4) Exit"<<endl;
    do {
        cout<<"Enter choice: "<<endl;
        cin>>ch;
        switch(ch) {
            case 1: {
                cout<<"Enter value to be pushed:"<<endl;
                cin>>val;
                push(val);
                break;
            }
            case 2: {
                pop();
                break;
            }
            case 3: {
                display();
                break;
            }
            case 4: {
                cout<<"Exit"<<endl;
                break;
            }
            default: {
                cout<<"Invalid Choice"<<endl;
            }
        } }while(ch!=4);
    return 0; }

```

#### **b) Queue operations**

```

#include <iostream>
using namespace std;

```

```

int queue[100], n = 3, front = -1, rear = -1;
void Insert() {
    int val;
    if (rear == n - 1)
        cout<<"Queue Overflow"<<endl;
    else {
        if (front == 0)
            front = 0;
        cout<<"Insert the element in queue : "<<endl;
        cin>>val;
        rear++;
        queue[rear] = val;
    }
}
void Delete() {
    if(front==-1) {
        cout<<"Element deleted from queue is : "<< queue[front] <<endl;
        front++;
    }
}
void Display() {
    if (front == - 1)
        cout<<"Queue is empty"<<endl;
    else {
        cout<<"Queue elements are : ";
        for (int i = front; i <= rear; i++)
            cout<<queue[i]<<" ";
        cout<<endl;
    }
}
int main() {
    int ch;
    cout<<"1) Insert element to queue"<<endl;
    cout<<"2) Delete element from queue"<<endl;
    cout<<"3) Display all the elements of queue"<<endl;
    cout<<"4) Exit"<<endl;
    do
    {
        cout<<"Enter your choice : "<<endl;
        cin>>ch;
    }
}

```

```

        switch (ch)
        {
            case 1: Insert();
            break;
            case 2: Delete();
            break;
            case 3: Display();
            break;
            case 4: cout<<"Exit"<<endl;
            break;
            default: cout<<"Invalid choice"<<endl;
        }
    } while(ch!=4);
    return 0;
}

```

### 3.4 INPUT/OUTPUT:

#### a) Stack operations

- 1) Push in stack
- 2) Pop from stack
- 3) Display stack
- 4) Exit

Enter choice:

1

Enter value to be pushed:

12

Enter choice:

1

Enter value to be pushed:

52

Enter choice:

1

Enter value to be pushed:

45

Enter choice:

1

Enter value to be pushed:

52

Stack Overflow

Enter choice:

2

The popped element is 45

Enter choice:

3

Stack elements are:52 12

Enter choice:

2

The popped element is 52

Enter choice:

3

Stack elements are:12

**b) Queue operations**

- 1) Insert element to queue
- 2) Delete element from queue
- 3) Display all the elements of queue
- 4) Exit

Enter your choice :

1

Insert the element in queue :

12

Enter your choice :

1

Insert the element in queue :

45

Enter your choice :

1

Insert the element in queue:

65

Enter your choice :

1

Queue Overflow

Enter your choice :

2

Element deleted from queue is: 0

Enter your choice:

52

Invalid choice

Enter your choice:

**3.5 PRE-LAB VIVA QUESTIONS:**

1. What are stacks and queues?
2. What is peek in stack?
3. What is a stack coding?
4. What are the real time applications of stack?

**3.6 LAB ASSIGNMENT:**

1. Write a C++ Program to Check Armstrong Number
2. Write a C++ Program to Find ASCII Value of a Character
3. Write a C++ Program to Check Leap Year

**3.7 POST-LAB VIVA QUESTIONS:**

1. What are the applications of stacks and queues?
2. Why is top in stack?
3. What does top do in stack?
4. What is stack with example?

## WEEK-4

### SETS AND STRINGS

#### 4.1 OBJECTIVE:

- a. You will be given Q queries. Each query is of one of the following three types:
  - i. x : Add an element x to the set.
  - ii. x : Delete an element x from the set. (If the number is not present in the set, then do nothing).
  - iii. x : If the number x is present in the set, then print "Yes"(without quotes) else print "No"(without quotes). For queries of type 3 print "Yes"(without quotes) if the number x is present in the set and if the number is not present, then print "No"(without quotes). Each query of type 3 should be printed in a new line.
- b. You are given a string containing characters A and B only. Your task is to change it into a string such that there are no matching adjacent characters. To do this, you are allowed to delete zero or more characters in the string. Your task is to find the minimum number of required deletions.

#### 4.2 RESOURCES:

Dev C++

#### 4.3 PROGRAM LOGIC:

##### a) Set Operations

```
#include <iostream>
#include <set>
using namespace std;
int main()
{
int iCount;
set<int> ss;
cin >> iCount;
for (int i=0; i<iCount; ++i){
int type, query;
cin >> type >> query;
switch (type){
case 1:
ss.insert(query);
break;
case 2:
ss.erase(query);
break;
case 3:
cout << (ss.find(query) == ss.end() ? "No" : "Yes") << endl;
break;
}
}
return 0; }
```

### b) String Operation

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    int i,count=0;
    char s[100];
    cout<<"\nenter string : ";
    cin>>s;
    for(i=0;i<strlen(s)-1;i++)
    {
        if(s[i]==s[i+1])
            count++;
    }
    cout<<"\nThe minimum no of required deletions is : "<<count;
    cout<<s;
    return 0;
}
```

## 4.4 INPUT/OUTPUT:

### a) Set Operations

Enter the size of the array

5

enter the type and data with tab spaces

1 11

1 22

1 33

1 44

1 55

### b) String Operation

enter string : AABAAB

The minimum no of required deletions is : 2

## 4.5 PRE LAB VIVA QUESTIONS:

- 1.What is set in STL?
- 2.How many parameters can a resize method take?
- 3.How do you define a set in C++?
- 4.How do you define a set?

## 4.6 LAB ASSIGNMENT

1. Write a C++ program to Swap two numbers.
2. Write a C++ program to find factorial of a number.
3. Write a C++ program to find largest number among three numbers.



**4.7 POST LAB VIVA QUESTIONS:**

- 1.What is Auto Type C++?
- 2.Is set in C++ sorted?
- 3.How do you clear a set in C++?
- 4.What is set in CPP?

## WEEK-5 SORTING AND PAIRS

### 3.1 OBJECTIVE:

- a. Raju and Ravi are friends. Raju asked Ravi to arrange the set of string in ascending order (Dictionary format). Please help the Ravi to put the strings in ascending order.
- b. Teacher given a task to students find the unvisited elements in the given matrix. The students are struggling to find the unvisited elements in the list. Please help them to solve.

### 5.2 RESOURCES:

Dev C++

### 5.3 PROGRAM LOGIC:

#### a) Sorting strings in ascending order

```
#include <iostream>
using namespace std;
int main()
{
    string str[10], temp;
    cout << "Enter 10 words: " << endl;
    for(int i = 0; i < 10; ++i)
    {
        getline(cin, str[i]);
    }
    for(int i = 0; i < 9; ++i)
        for( int j = i+1; j < 10; ++j)
        {
            if(str[i] > str[j])
            {
                temp = str[i];
                str[i] = str[j];
                str[j] = temp;
            }
        }
    cout << "In ascending order: " << endl;
    for(int i = 0; i < 10; ++i)
    {
        cout << str[i] << endl;
    }
    return 0;
}
```

#### b) Find unvisited elements in the given matrix

```
#include<iostream>
#include<string.h>
using namespace std;
int main()
{
    int a[25][25],i,j,r,c;
```

```

        cout<<"\nenter no of rows and columns : ";
        cin>>r>>c;
        cout<<"\nenter matrix elements : ";
        for(i=0;i<r;i++)
        {
            for(j=0;j<c;j++)
            {
                cin>>a[i][j];
            }
        }
        cout<<"\nmatrix is : \n";
        for(i=0;i<r;i++)
        {
            for(j=0;j<c;j++)
            {
                cout<<a[i][j]<<"\t";
            }
            cout<<"\n";
        }
        cout<<"\nunvisited elements are : ";
        for(i=0;i<r;i++)
        {
            for(j=0;j<c;j++)
            {
                if(a[i][j]==0)
                    cout<<"\na["<<i<<"]["<<j<<"];
            }
        }
        return 0;
    }

```

#### 5.4 INPUT/OUTPUT:

##### a) Sorting strings in ascending order

Enter 10 words:

C

C++

Java

Python

Perl

R

Matlab

Ruby

JavaScript

PHP

In ascending order order:

C

C++

Java

JavaScript  
Matlab  
PHP  
Perl  
Python  
R  
Ruby

**b) Find unvisited elements in the given matrix**

enter no of rows and columns : 3

3

enter matrix elements : 2

0

5

9

1

0

0

3

7

matrix is :

2 0 5

9 1 0

0 3 7

unvisited elements are :

a[0][1]

a[1][2]

a[2][0]

**5.5 PRE-LAB VIVA QUESTIONS:**

1. How do you sort a pair in C++?
2. Which sorting is best and why?
3. What is merge sort and how it works?

**5.6 LAB ASSIGNMENT:**

1. Write a C++ program to check whether a number is palindrome or not.
2. Write a C++ program to create pyramid and pattern.
3. Write a C++ program to find all roots of a quadratic equation

**5.7 POST-LAB VIVA QUESTIONS:**

1. What are the types of sorting?
2. What is sorting with example?
3. Which sort is best?

## WEEK-6

### ARRAYS AND LISTS

#### 6.1 OBJECTIVE:

All friends are invited and they arrive at the party one by one in an arbitrary order. However, they have certain conditions — for each valid  $i$ , when the  $i$ -th friend arrives at the party and sees that at that point, strictly less than  $A_i$  other people (excluding Chef) have joined the party, this friend leaves the party; otherwise, this friend joins the party. Help Chef estimate how successful the party can be — find the maximum number of his friends who could join the party (for an optimal choice of the order of arrivals).

#### 6.2 RESOURCES:

Dev C++

#### 6.3 PROGRAM LOGIC:

```
#include <iostream>
#include <cmath>
#include <algorithm>
using namespace std;

int main()
{
    int t,i=0,n,l,j,f,a[100005];
    cin>>t;
    while(t)
    {
        f=0;
        cin>>n;
        l=n;
        i=0;
        while(l)
        {
            cin>>a[i];
            i++;
            l--;
        }
        sort(a,a+n);

        if(a[0]!=0)
        {
            cout<<0<<endl;
            t--;
            continue;
        }
    }
}
```

```

        j=0;
        while(a[j]<=f && j<n)
        {
            f++;
            j++;
        }
        cout<<f<<endl;
        t--;
    }
}

```

#### 6.4 INPUT/OUTPUT:

Enter the value of n

5

3 1 0 0 5 5

3

4

2 3 4 5 6 7

0

2 1 0 8 9 0

3

4 2 4 5 6 7

1

3 5 8 0 7 8

1

#### 6.5 PRE-LAB VIVA QUESTIONS:

1. What is the index number of the last element of an array with 9 elements?
2. What is array data structure in C++?
3. What is C++ language interview question?

#### 6.6 LAB ASSIGNMENT:

1. Write a C++ programs to find GCD of two numbers
2. Write a C++ programs to addition of two matrix
3. Write a C++ programs to reverse array element using function

#### 6.7 POST-LAB VIVA QUESTIONS:

1. What is array in data structure with example?

2. What is array with example?
3. What is array and its types?

## WEEK-7

### MULTISET AND MULTIMAPS

#### 7.1 OBJECTIVE:

Kattapa, as you all know was one of the greatest warriors of his time. The kingdom of Maahishmati had never lost a battle under him (as army-chief), and the reason for that was their really powerful army, also called as Mahasena. Kattapa was known to be a very superstitious person. He believed that a soldier is "lucky" if the soldier is holding an even number of weapons, and "unlucky" otherwise. He considered the army as "READY FOR BATTLE" if the count of "lucky" soldiers is strictly greater than the count of "unlucky" soldiers, and "NOT READY" otherwise. Given the number of weapons each soldier is holding, your task is to determine whether the army formed by all these soldiers is "READY FOR BATTLE" or "NOT READY".

#### 7.2 RESOURCES:

Dev C++

#### 7.3 PROGRAM LOGIC:

```
#include<iostream>
using namespace std;
int main()
{
    int n,i,a[100],even=0,odd=0;
    cout<<"enter no of soldiers : ";
    cin>>n;
    for(i=0;i<n;i++)
    {
        cout<<"enter no of weapons of soldier "<<i+1<<" : ";
        cin>>a[i];
        if(a[i]%2==0)
            even++;
        else
            odd++;
    }
    if(even>odd)
        cout<<"READY FOR BATTLE";
    else
        cout<<"NOT READY";
    return 0;
}
```



#### **7.4 INPUT/OUTPUT:**

enter no of soldiers : 5  
enter no of weapons of soldier 1 : 2  
enter no of weapons of soldier 2 : 35  
enter no of weapons of soldier 3 : 6  
enter no of weapons of soldier 4 : 88  
enter no of weapons of soldier 5 : 9  
READY FOR BATTLE

enter no of soldiers : 6  
enter no of weapons of soldier 1 : 9  
enter no of weapons of soldier 2 : 12  
enter no of weapons of soldier 3 : 13  
enter no of weapons of soldier 4 : 18  
enter no of weapons of soldier 5 : 20  
enter no of weapons of soldier 6 : 49  
NOT READY

#### **7.5 PRE-LAB VIVA QUESTIONS:**

- 1.What is the difference between map and set in C++?
- 2.What is the use of maps in C++?
- 3.What is the difference between map and Multimap?

#### **7.6 LAB ASSIGNMENT:**

- 1.Write a C++ program to Compare Two Strings.
- 2.Write a C++ program to Calculate Area and Perimeter of Rectangle.
- 3.Write a C++ program to swap two number with the help of third variable.

#### **7.7 POST-LAB VIVA QUESTIONS:**

- 1.What is a multiset in C++?
- 2.What is unordered set in C++?
- 3.What is the difference between set and multiset in C++?

## WEEK-8 UNORDERED SETS

### 8.1 OBJECTIVE:

You are given two lists of N distinct numbers. Sort both the list and print them alternatively starting with list one.

### 8.2 RESOURCES:

Dev C++

### 8.3 PROGRAM LOGIC:

```
#include <bits/stdc++.h>
using namespace std;

// Function to merge array in sorted order
void sortedMerge(int a[], int b[], int res[],
                 int n, int m)
{
    // Concatenate two arrays
    int i = 0, j = 0, k = 0;
    while (i < n) {
        res[k] = a[i];
        i += 1;
        k += 1;
    }
    while (j < m) {
        res[k] = b[j];
        j += 1;
        k += 1;
    }

    // sorting the res array
    sort(res, res + n + m);
}

// Driver code
int main()
{
    int a[] = { 10, 5, 15 };
    int b[] = { 20, 3, 2, 12 };
    int n = sizeof(a) / sizeof(a[0]);
    int m = sizeof(b) / sizeof(b[0]);

    // Final merge list
    int res[n + m];
    sortedMerge(a, b, res, n, m);
}
```

```
cout << "Sorted merged list :";  
for (int i = 0; i < n + m; i++)  
    cout << " " << res[i];  
cout << "n";  
  
return 0;  
}
```

#### **8.4 INPUT/OUTPUT:**

Sorted merged list : 2 3 5 10 12 15 20

#### **8.5 PRE-LAB VIVA QUESTIONS:**

1. What is unordered set in C++?
2. What is the difference between set and unordered set?
3. How do you check if an element is in a set C++?

#### **8.6 LAB ASSIGNMENT:**

1. Write a program in C++ to find the greatest number in two numbers with the help of if/else.
2. Write a C++ program for union and intersection of two sorted arrays
3. Write a C++ program for bubble sort

#### **8.7 POST-LAB VIVA QUESTIONS:**

1. What is the difference between ordered and unordered sets?
2. What is the difference between set and multiset in C++?
3. Explain whether set in C++ sorted?

## WEEK-9

### SET UNION AND INTERSECTION

#### 9.1 OBJECTIVE:

A class contains two subjects and students can take one or two subjects as there wish. Here, students opted subjects on there own interest. Now, your task is to print all the total students count and students names, and also print how many took two subjects and their names.

#### 9.2 RESOURCES:

Dev C++

#### 9.3 PROGRAM LOGIC:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int t,n,m,i,d,j,h,ma;
    cin>>n>>m;
    int a[n][2],b[m][2];
    for(i=0;i<n;i++)
    {
        cin>>a[i][0];
    }
    for(i=0;i<m;i++)
    {
        cin>>b[i][0];
        cin>>b[i][1];
        for(j=b[i][0];j<=b[i][1];j++)
            a[j][1]=i;
    }
    cin>>d;int x;
    for(i=0;i<n;i++)
    {
        int ma=a[i][0];h=0;
        if(i-d<0)
            j=0;
        else
            j=i-d;
        if(i+d<=n)
            x=i+d;
        else
            x=n;
        for(;j<=x;j++)
        {
```

```

        if(a[j][1]!=a[i][1])
        {
        if(a[j][0]>ma)
        {
                ma=a[j][0];h=0;
        }
        else if((a[j][0]==ma)&&(a[j][0]!=a[i][0]))
        {
                h++;
        }
        }
}
if(ma!=a[i][0])
cout<<ma<<" "<<h+1<<"\n";
else
cout<<-1<<"\n";
}
return 0;
}

```

**9.4 INPUT / OUTPUT :**

```

10 5
1 2 1 2 1 2 2 1 2 3
1 2
3 4
5 6
7 8
9 10
2

```

**9.5 PRE-LAB VIVA QUESTIONS:**

1. How do you find the union and intersection of two arrays in Java?
2. How do you find the kth largest element in an unsorted array?
3. What is Union in array?

**9.6 LAB ASSIGNMENT:**

1. Write a C++ program for insertion Sort
2. Write a C++ program to raise any number x to a positive power n
3. Write a C++ program to calculate square root of any number

**9.7 PRE-LAB VIVA QUESTIONS:**

1. How do you find the intersection of two sets?
2. What are the examples of intersection of sets?
3. What does  $\cap$  mean?

## WEEK-10

### IMPLEMENTATION OF QUEUE USING LINKED LIST

#### 10.1 OBJECTIVE:

A class contains two subjects and students can take one or two subjects as there wish. Here, students opted subjects on their own interest. Now your task to find the student names who are attending first subject but not second and vice versa.

#### 10.2 RESOURCES:

Dev C++

#### 10.3 PROGRAM LOGIC:

```
#include <stdio.h>
#include <stdlib.h>
struct Node
{
int data;           // integer data
struct Node* next; // pointer to the next node
}*rear = NULL, *front = NULL;
struct Node* newNode(int item)
{
struct Node* node = (struct Node*)malloc(sizeof(struct Node));
if (node != NULL)
{
node->data = item;
node->next = NULL;
return node;
}
else
{
cout<<"\nHeap Overflow";
exit(EXIT_FAILURE);
}
}
{
if (front == NULL)
{
cout<<"\nQueue Underflow";
exit(EXIT_FAILURE);
}

struct Node *temp = front;
cout<<"Removing %d\n", temp->data;
front = front->next;
if (front == NULL)
```

```

rear = NULL;

// deallocate the memory of removed node and
// optionally return the removed item
int item = temp->data;
free(temp);
return item;
}

// Utility function to add an item in the queue
void enqueue(int item) // insertion at the end
{
// Allocate the new node in the heap
struct Node* node = newNode(item);
cout<<"Inserting %d\n", item;

// special case: queue was empty
if (front == NULL)
{
// initialize both front and rear
front = node;
rear = node;
}
else
{
// update rear
rear->next = node;
rear = node;
}
}

// Utility function to return top element in a queue
int peek()
{
// check for empty queue
if (front != NULL)
return front->data;
else
exit(EXIT_FAILURE);
}

// Utility function to check if the queue is empty or not
int isEmpty()
{
return rear == NULL && front == NULL;
}

```

```

    }

    // main function
    int main()
    {
        enqueue(1);
        enqueue(2);
        enqueue(3);
        enqueue(4);

        cout<<"Front element is %d\n", peek();

        dequeue();
        dequeue();
        dequeue();
        dequeue();

        if (isEmpty())
            cout<<"Queue is empty";
        else
            cout<<"Queue is not empty";

        return 0;
    }

```

#### **10.4 INPUT / OUTPUT:**

Enter the elements in queue

1

2

3

4

The queue elements are 1 2 3 4

#### **10.5 PRE-LAB VIVA QUESTIONS:**

1. How is linked list implemented?
2. What type of memory allocation is referred for linked lists?
3. What is the need for linked representation of lists?

#### **10.6 LAB ASSIGNMENT:**

1. Write a C++ program to insert an element in an array
2. Write a C++ program to do linear search in an array
3. Write a C++ program to print three numbers in descending order



**10.7 POST-LAB VIVA QUESTIONS:**

1. How many pointers are required to implement a simple linked list?
2. What are linked lists good for?
3. What are the advantages of linked list?

## WEEK-11 PERMUTATIONS

### 11.1 OBJECTIVE:

IARE college has designed a new challenge called BuildIT Competitive Programming. In this game, each team contains N members and they are specialised in either Java Programming or Python Programming. The challenge contains n1 java questions and n2 Python questions. So, team members are decided to seat in all specialized members as one group. So that, number of ways the N members seat in the programming contest.

### 11.2 RESOURCES:

Dev C++

### 11.3 PROGRAM LOGIC:

```
#include <bits/stdc++.h>
using namespace std;
void permute1(string a, int l, int r, char x[])
{
    int p=0;
    if (l == r)
    {
        string x[p]=a;
        cout<<a<<x[p]<<endl;
        p++;
    }
    else
    {
        for (int i = l; i <= r; i++)
        {
            swap(a[l], a[i]);
            permute1(a, 0, r-1,x);
            swap(a[l], a[i]);
        }
    }
}

void permute2(string b, int l, int r, char y[])
{
    int q=0;
    if (l == r)
    {
        string y[q]=b;
        cout<<b<<y[q]<<endl;
        q++;
    }
}
```

```

else
{
    for (int i = l; i <= r; i++)
    {
        swap(b[l], b[i]);
        permute2(b, l+1, r,y);
        swap(b[l], b[i]);
    }
}
}
int main()
{
    string str1,str2;
    cin>>str1>>str2;
    char x[100], y[100];
    int n1 = str1.size();
    int n2 = str2.size();
    permute1(str1, 0, n1-1,x);
    permute2(str2, 0, n2-1,y);
    return 0;
}

```

#### 11.4 INPUT/OUTPUT: Enter the input: ab cde

Output is  
abcde  
abced  
abdce  
abdec  
abecd  
abedc  
bacde  
baced  
badce  
badec  
baecd  
baedc

#### 11.5 PRE-LAB VIVA QUESTIONS:

1. How do you find permutations?
2. What is permutations and combination?
3. Where permutation and combination is used?

#### 11.6 LAB ASSIGNMENT:

1. Write a C++ program to check whether a number is odd or even.
2. Write a C++ program to print fibonacci series.

3. Write a C++ program to Count no. of words in a string.

**11.7 POST-LAB VIVA QUESTIONS:**

1. How do you tell if a question is a permutation or combination?
2. What is permutation formula?
3. What is permutation example?

## WEEK-12

### LEXICOGRAPHICAL

#### 12.1 OBJECTIVE:

Ravi and Raju are best friends. Ravi given a set of strings to Raju and ask him to find smaller string as per lexicographical order. Please help him to find.

#### 12.2 RESOURCES:

Dev C++

#### 12.3 PROGRAM LOGIC:

```
#include <bits/stdc++.h>
using namespace std;

// str : Stores input string
// n : Length of str.
// curr : Stores current permutation
// index : Index in current permutation, curr
void permuteRec(string str, int n,
                int index = -1, string curr = "")
{
    // base case
    if (index == n)
        return;

    cout << curr << "\n";
    for (int i = index + 1; i < n; i++) {

        curr += str[i];
        permuteRec(str, n, i, curr);

        // backtracking
        curr = curr.erase(curr.size() - 1);
    }
    return;
}

// Generates power set in lexicographic
// order.
void powerSet(string str)
{
    sort(str.begin(), str.end());
    permuteRec(str, str.size());
}
```

```
// Driver code
int main()
{
    string str = "cab";
    powerSet(str);
    return 0;
}
```

**12.4 INPUT/OUTPUT:**

a  
ab  
abc  
ac  
b  
bc  
c

**12.5 PRE-LAB VIVA QUESTIONS:**

- 1.What is lexicographical order in string?
- 2.What does Lexicographically greater mean?
- 3.How do you compare two strings lexicographically?

**12.6 LAB ASSIGNMENT:**

- 1.Write a C++ program to find length of a string.
- 2.Write a C++ program to find sum of square of n natural numbers.
- 3.Write a C++ program to find sum of digits of a number.

**12.7 POST-LAB VIVA QUESTIONS:**

- 1.What is lexicographic order of numbers?
- 2.What is lexicographic order example?
- 3.What does lexicographical mean?