



Graphic Era
HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

Term work

(Midsem)

on
OOPs with C++

(PCS 307)

2021-22

Submitted to:

Akash Chauhan

Assistant Professor

GEHU, D.Dun

Submitted by:

Vishal Joshi

University Roll. No.: 2018861

Class Roll. No./Section: 63/A

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

ACKNOWLEDGMENT

I would like to particularly thank my OOPs with C++ Lab Faculty **Mr. Akash Chauhan** for his patience, support and encouragement throughout the completion of this Term work.

At last but not the least I greatly indebted to all other persons who directly or indirectly helped me during this course.

VISHAL JOSHI

University. Roll No.- 2018861

B.Tech CSE-B-III Sem

Session: 2021-22

GEHU, Dehradun



Table of Contents

Program No.	Program Name	Page
1	Run all four compilation units individually for any sample program using C++.	3
2	Write an Efficient code to check if a number is prime or not.	5
3	<p>Write C++ code for below mentioned tasks?</p> <p>Task1: How the preprocessor will react when you try to use, #include<iostream.h> instead of #include<iostream>?</p> <p>Task2: How the preprocessor will react when you use cout but don't include #include<iostream> in your code?</p> <p>Task3: Take a char variable and use cin to take its value from the user, cout it and it will only return one character and loss rest of the data you have entered!</p> <p>Task4: How to resolve above issue? [Hint: by using, getline(cin, line);]</p>	8-15
4	<p>Write C++ code for below mentioned tasks?</p> <p>add(int,int), add(float, float), both of these methods are in two different namespaces First and Second respectively.</p> <p>Task1: Access these methods using scope resolution operator [::](SRO) from main method?</p> <p>Task2: Access these methods using "using" keyword for main method?</p> <p>Task3: Try to access these methods without using, (SRO) and "using" keyword and check how the compiler will react to it?</p> <p>Task4: Try to access these methods for Mixed Values [Int, Float] and see how the compiler will react to it?</p>	16-23
5	<p>Write C++ code for below mentioned tasks?</p> <p>Primary Data Types related questions in C++:</p> <p>Task1: Initialise all primary data types, assign their values and print them all? [char, bool, short, int, long, float, double, long double, wide char]</p> <p>Task2: Apply sizeof operator on all above operators and their variables?</p>	24-30

	<p>Example</p> <pre>Int a = 10; Cout << Sizeof(int); Cout << Sizeof(a);</pre>	
6	<p>Write C++ code for below mentioned tasks?</p> <p>String related Questions in C++:</p> <p>Task1: What happens if we add integer with a string, how the compiler would react to it?</p> <pre>String str = "ABC"; Int a = 1; String str2 = str + a;</pre> <p>Task2: Check the entered string is Palindrome or not?</p> <p>String str = "75457"</p> <p>Output: Yes it is a palindrome or No it is not a Palindrome.</p> <p>(Use, getline(sin, str1) and reverse_iterator of string to check this)</p> <p>Task3: Make a reverse of a string using reverse method and reverse_iterator of string class?</p> <p>Task4: String Compare: Check if the strings are equal or not? (do not use str1.compare(str2), do it manually)</p> <p>Task5: String Compare: Check the possible values string.compare() function will return?</p> <p>(Create cases in which compare function would return below values)</p> <p>X>0 X<0 X==0 X = -4 X = 5 X = -2104040...</p> <p>Also check the ASCII difference between two characters?(use int type cast)</p> <p>Task6: Check if string is mutable in C++ or not?</p> <pre>String a = "Hello"; Cout << &a; a[0] = 'J'; Cout << &a; Cout << a;</pre> <p>What is the output?</p>	31-45
7	Write C++ code for below mentioned tasks?	46-76

Array and 2D Array related Questions in C++:

Task1: Create a switch statement [Manual], In Which:

- a. When you pass 1 your program would print current year
- b. When you pass 2 your program would print current month
- c. When you pass 3 your program would print current day
- d. When you pass 4 your program would print Not applicable

Task2: Create a switch statement [Using ctime], In Which:

- a. When you pass 1 your program would print current year
- b. When you pass 2 your program would print current month
- c. When you pass 3 your program would print current day
- d. When you pass 4 your program would print Not applicable

Task3:

v1. Print using reverse method:

1 2 3 9 8 7
4 5 6 ==> 6 5 4
7 8 9 3 2 1

v2. Print using (10- arr[i][j]) method:

1 2 3 9 8 7
4 5 6 ==> 6 5 4
7 8 9 3 2 1

v3. Restore using reverse method [without creating new array]:

1 2 3 9 8 7
4 5 6 ==> 6 5 4
7 8 9 3 2 1

v4. Restore using (10- arr[i][j]) method [without creating new array]:

1 2 3 9 8 7
4 5 6 ==> 6 5 4
7 8 9 3 2 1

Task4: Restore the same values in the same array, arr[3][3]:

1 2 3 1 1 1
4 5 6 ==> 2 2 2
7 8 9 3 3 3

- v1. Use row loop [int i, for all j]
 v2. Use arr[i][N-1]/3, at each place
 v3. Use, arr[i][j]-(2*i+j)

Task5: Store these in an array[4][4] in given fashion and then print:

```
*  
* *  
* * *  
* * * *
```

Task6: Store these in an array[4][4] in given fashion and then print:

```
* * * *  
* * *  
* *  
*
```

Task7: Store these in an array[4][4] in given fashion and then print:

```
*  
* *  
* * *  
* * * *
```

Task8: Store these in an array[4][4] in given fashion and then print:

```
* * * *  
* * *  
* *  
*
```

8

Write C++ code for below mentioned tasks?

77-96

Pointer, Function, Inline Function, Recursion in C++:

Task1: Will the program through an error and if yes then why?

```
int *p = {10,20,20};  
cout << *p;  
p++;  
cout << *p;
```

Task2: Output of this program?

V1. Issue?

```
int arr[] = {10,20,30};  
cout << *arr;  
cout << arr;  
arr++;  
cout << *arr;
```

V2. How to resolve above issue?

```
int arr[] = {10,20,30};  
cout << *arr;  
cout << arr;  
cout << *(?);
```

Task3: Output of this program?

V1. Output?

```
int a = 10;  
int *p;  
int **q;  
p = &a;  
q = &p;  
cou << *p;  
cou << **q;
```

V2. Change the value of a using q pointer to pointer.

Task4: Find factorial of a number using function but not recursion

Task5: Find factorial of a number using recursion

Task6: Series Problem using recursion for n series

2, (2^2 + 2), (3^3 + 3), (4^4 + 4), (5^5 + 5),

Hint:

$n * ((n-1)^(n-1) + (n-1))$

Task7: Perform Call by value, call by Address for swapping value of a and b:

```
int a = 10;  
int b = 20;
```

V1. Swap(a,b); //call by Value [void swap(int a, int b){}]

V2. Swap(a,b); //call by Value [void swap(int &a, int &b){}]

V3. Swap(&a,&b); //call by Address

	<p>Class, Object, Constructor, Static Data Members, friend function in C++:</p> <p>Task1: Class and Object in C++</p> <p>a. WAP to assign and print the roll number, phone number and address of two students having names "Sam" and "John" respectively by creating two objects of the class 'Student'.</p> <p>b. WAP which would contain array of objects [many objects], of a class Student [Name, Age, Year, section, marks], the section would be A,B,C and D. Your program would be able to return the total marks of students in the college.</p> <p>Hint [Make a Matrix or Tabular diagram to understand the problem], all the rows will differ each other by different objects of Student class [Student s1,s2,s3,s4].</p> <p>Task2: Constructor in C++</p> <p>WAP to create a class to print the area of a square and a rectangle. The class has two functions with the same name but different number of parameters. The function for printing the area of rectangle has two parameters which are its length and breadth respectively while the other function for printing the area of square has one parameter which is the side of the square. Use multiple constructors to for the initialization.</p> <p>Task3: Static Data Members in C++</p> <p>WAP to count the total number of calls for a member function from more than one objects. [Lets say, from 3 such Objects]</p> <p>Task4: Friend Function in C++</p> <p>WAP in which you create a Student class having basic information for each student, like name, age and marks. By using friend function add marks of all the students [lets say 3 objects] and print it.</p> <p>Task5: Structure in C++</p> <p>WAP to create a College class and Student Structure in C++ in one program. By providing such suitable examples write at least 5 differences between class and struct code your have written above.</p> <p>Hint [Access Specifiers, Heap and Stack, large and small memory, etc.]</p> <p>Task6: Extra Questions:</p> <p>WAP which would perform these tasks of your data:</p> <ul style="list-style-type: none"> a. Come to next line b. set minimum field width c. fill string with (*) after setw(15) function *****1234 <p>by using endl, setw, and setfill [Manipulators in C++]</p>	
10	<p>Write C++ code for below mentioned tasks?</p> <p>Array of Objects, Pointer to Object, This pointer, Operator Overloading in C++</p> <p>Task1: Array of Objects in C++</p> <p>WAP to create a directory that contains the following information.</p> <ul style="list-style-type: none"> (a) Name of a person (b) Address (c) Telephone Number (if available with STD code) 	113-123

	<p>(d) Mobile Number (if available) (e) Head of the family</p> <p>Task2: Pointer to Object in C++ WAP to create print or display Student information containing in Student class by using pointers to object.</p> <p>Task3: This pointer in C++ WAP to pass two variables in a parameterized constructor during object creation and have same names variables as class member data and constructor parameters. Your job is to calculate the remainder of those two numbers.</p> <p>Task4: Operator Overloading in C++ a). WAP, in which you write a friend function to overload a less than '<' operator in C++. b). WAP in which you can add two objects [every object would have 1 integer value] by overloading + operator, which eventually would add the data values of those two object by adding the objects.</p>	
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

11	<p>Write C++ code for below mentioned tasks?</p> <p>Task-11.1 Inheritance Basics: WAP in C++ to create a Parent and Child interaction using inheritance. With this Parent and Child Interaction try to perform these tasks:</p> <ul style="list-style-type: none"> a. Call Parent class method in child class function without creating an object of parent class b. Call Parent class method in main method by child class object <p>Task-11.2 Multiple Inheritance in C++: Create two classes named Mammals and MarineAnimals. Create another class named BlueWhale which inherits both the above classes. Now, create a function in each of these classes which prints "I am mammal", "I am a marine animal" and "I belong to both the categories: Mammals as well as Marine Animals" respectively. Now, create an object for each of the above class and try calling</p> <p>1 - function of Mammals by the object of Mammal 2 - function of MarineAnimal by the object of MarineAnimal 3 - function of BlueWhale by the object of BlueWhale 4 - function of each of its parent by the object of BlueWhale</p> <p>Task-11.3 Dimond Problem in multiple inheritance using C++:</p> <ul style="list-style-type: none"> a. WAP to illustrate Dimond Problem in multiple inheritance b. its solution using Virtual base classes. Write separate programs if required. c. What else multiple inheritance can cause in a program, explain it by providing proper solution 	124-135
----	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------

12	<p>Write C++ code for below mentioned tasks?</p> <p>Task-12.1 WAP to illustrate the role of Access Modifiers [private, public, protected] separately in:</p> <ul style="list-style-type: none"> a. Accessing base class elements in derived class or Inheritance b. Accessing base class elements through object <p>Task-12.2 Execution flow of Constructors and Destructors in C++:</p> <ul style="list-style-type: none"> a. WAP to illustrate the calling and execution flow of Constructors in inheritance. [L-2 Inheritance] b. WAP to illustrate the calling and execution flow of Destructors in inheritance. [L-2 Inheritance] c. Pass parameters to base class through derived class constructor. [L-1 Inheritance] 	136-145
----	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------

13	<p>Write C++ code for below mentioned tasks?</p> <p>Task 13.1 To overload add method for two parameters with int and float data types in Base class. Along with it create a Derived class from the Base class named as child. The class child should override one of the overloaded method from base class.</p> <p>Perform following tasks:</p> <ol style="list-style-type: none"> 1. Try calling overriding method from child class object. 2. Write name of the method which is not seen by the child class object <p>Create two versions:</p> <p>version 01: without 'using' keyword</p> <p>version 02: with 'using' keyword</p> <p>Task 13.2 Base class having a virtual and a pure virtual function. Derived class having same copy of virtual function with changed logic and definition of pure virtual function.</p> <p>Perform following tasks:</p> <ol style="list-style-type: none"> 1. Try to call child class overriding method from base class pointer. 2. Try to call child class definition of pure virtual function in child class. 3. Find out the abstraction in above implementation. 	146-157
14	<p>Write C++ code for below mentioned tasks?</p> <p>Task 14.1 Illustrate the compile time and run time binding using base class pointer, which holds the address of child class</p> <p>Task 14.2 Perform the following:</p> <ol style="list-style-type: none"> 1. Call base class destructor from base class pointer which is holding the child class object. 2. Call child class destructor from base class pointer which is holding the child class object. 	158-163
15	<p>Write C++ code for below mentioned tasks?</p> <p>Task 15.1 WAP in C++ to read and write from and to a file using ifstream and ofstream.</p> <p>Task 15.2 WAP in C++ to perform these tasks:</p> <ol style="list-style-type: none"> a. Read from a file using fstream [char by char] b. Write into a file using fstream c. Append into a file using fstream d. Count total number of characters, words and lines in a file <p>Task 15.3 WAP in C++ for IO manipulators mentioned below:</p> <ol style="list-style-type: none"> a. IOS: hex,dec,skipws,noskipws b. Istream: ws 	164-179

	<p>c. Ostream: endl, ends, flush d. Iomanip: setW, setPrecision</p>	
16	<p>Write C++ code for below mentioned tasks?</p> <p>Task 16.1 WAP in C++ to create a generic add function for given tasks:</p> <ul style="list-style-type: none"> a. Perform add over two integers and return integer b. Perform add over one int and one float and return double <p>Task 16.2 WAP in C++ to perform these tasks:</p> <ul style="list-style-type: none"> a. Catch a Divide by zero exception in $z = x/y$ using "throw runtime_error" b. What will be the output of this program and why? <pre>#include <iostream> using namespace std; int main() { try { throw 'a'; } catch (int x) { cout << "Caught " << x; } catch (...) { cout << "Default Exception\n"; } return 0; }</pre> <ul style="list-style-type: none"> c. What will be the output of this program and why? <pre>#include <iostream> using namespace std; int main() { try { throw 'a'; } catch (int x) { cout << "Caught "; } return 0; }</pre> <p>d. Rethrow and catch an exception by creating a separate user defined divide function for condition divide by zero.</p>	180-191

17	<p>WAP in C++ with the help of STL:</p> <p>a. List:</p> <ol style="list-style-type: none"> 1. Iterate a int list using iterator and print it 2. Find size of a list 3. Sort a list 4. Reverse a list <p>b. Vector:</p> <ol style="list-style-type: none"> 1. Insert elements into a int vector 2. Iterate this vector using iterator and print it 3. Find size of a capacity and max size of a vector 4. Resize a vector 5. checks if the vector is empty or not <p>c. Map:</p> <ol style="list-style-type: none"> 1. Insert elements into a <code><int, string></code> map 2. insert elements in random order 3. Iterate this map using iterator and print its keys and values 4. Find an element as key from this map 5. assigning the elements from map1 to map2 6. remove all elements with key = x (any key present in map) 7. Find size, max size of a map 8. checks if this map is empty or not 9. Clear a map <p>d. Algorithm:</p> <ol style="list-style-type: none"> 1. Convert an Array into a Vector 2. Sort a Vector 3. Reverse a vector 4. Max element in a Vector 5. Min element in a Vector 6. Occurrences of x in a vector 7. Sort an Array 8. Binary Search in an Array 	192-255
----	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------



DEPARTMENT OF CSE
B.Tech. CSE
STUDENT LAB REPORT SHEET

Name of Student Mob.No.....

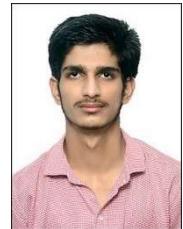
Address Permanent

Father's Name Occupation MoNo.....

Mother's Name Occupation.....MoNo.....

SectionBranch.....Semester.....Class Roll No..... Grade A B C

Local Address.....Email..... Marks 5 3 1



S.N o.	Practical	D.O.P.	Date of Submissi on	Gra de (Viv a)	Grade (Report File)	Total Marks (out of 10)	Student's Signature	Teacher's Signature
1	Practical -01	23.09.21	23.09.21				VISHAL	
2	Practical -02	23.09.21	23.09.21				VISHAL	
3	Practical -03	02.10.21	02.10.21				VISHAL	
4	Practical -04	02.10.21	02.10.21				VISHAL	
5	Practical -05	07.10.21	07.10.21				VISHAL	
6	Practical -06	07.10.21	07.10.21				VISHAL	
7	Practical -07	17.10.21	18.10.21				VISHAL	
8	Practical -08	28.10.21	30.10.21				VISHAL	
9	Practical -09	28.10.21	30.10.21				VISHAL	
10	Practical -10	28.10.21	30.10.21				VISHAL	
11	Practical -11	14.12.21	14.12.21				VISHAL	
12	Practical -12	14.12.21	14.12.21				VISHAL	
13	Practical -13	27.12.21	27.12.21				VISHAL	

14	Practical -14	27.12.21	27.12.21				VISHAL	
15	Practical -15	03.01.22	03.01.22				VISHAL	
16	Practical-16	03.01.22	03.01.22				VISHAL	
17	Practical-17	06.01.22	06.01.22				VISHAL	

Program 1

Source Code:

```
#include <iostream>
using namespace std;
#define MAX 50
int main()
{
    cout <<"Sum of First 50 Natural Number is "<<MAX*(MAX+1)/2;
    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with the following windows:

- Terminal Window:** Located at the bottom, it displays the command-line history and the source code for `process.cpp`.
- File Manager Window:** Located in the center, it shows the directory structure and files related to the C++ program.
- Code Editor Window:** Located at the top, it shows the `process.cpp` file with its code.

Terminal Window Content (process.cpp):

```
#include <iostream>
using namespace std;
#define MAX 50
int main()
{
    cout << "Sum of First 50 Natural Number is "<<MAX*(MAX+1)/2;
    return 0;
}
```

File Manager Window Content:

```
knockcat@VICKY:~/Documents/OOPS in C++/practical 1- process involving while compiling a c++ program$ ls
knockcat@VICKY:~/Documents/OOPS in C++/practical 1- process involving while compiling a c++ program$ g++ -E process.cpp >process.i
knockcat@VICKY:~/Documents/OOPS in C++/practical 1- process involving while compiling a c++ program$ g++ -S process.i
knockcat@VICKY:~/Documents/OOPS in C++/practical 1- process involving while compiling a c++ program$ g++ -c process.s
knockcat@VICKY:~/Documents/OOPS in C++/practical 1- process involving while compiling a c++ program$ g++ -o process.exe process.cpp
knockcat@VICKY:~/Documents/OOPS in C++/practical 1- process involving while compiling a c++ program$ ./process
knockcat@VICKY:~/Documents/OOPS in C++/practical 1- process involving while compiling a c++ program$
```

Code Editor Window Content (process.cpp):

```
#include <iostream>
using namespace std;
#define MAX 50
int main()
{
    cout << "Sum of First 50 Natural Number is "<<MAX*(MAX+1)/2;
    return 0;
}
```

Program 2

Source Code:

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

int sqr(int n)
{
    int ans;
    if(n==0)
        return 0;
    int start=1,end=n;
    while(start<=end)
    {
        int mid=start+(end-start)/2;
        if(mid<=n/mid)
        {
            ans=mid;
            start=start+1;
        }
        else end=mid-1;
    }
    return ans;
}

bool isPrime(int n)
{
    if (n <= 1)
        return false;

    for (int i = 2; i <= sqr(n); i++)
        if (n % i == 0)
            return false;

    return true;
}
```

```
}
```

```
int main()
{
    int n;
    cout<<"Enter Number : ";
    cin>>n;
    isPrime(n) ? cout <<n<<" is prime" : cout <<n<<" is notprime";

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a file manager window.

Terminal Window:

```

Activities   Files ▾
knockcat@VICKY:~/Documents/OOPS in C++/Practical 2- Prime Number$ ls
knockcat@VICKY:~/Documents/OOPS in C++/Practical 2- Prime Number$ gedit primesqr.cpp
[1]+  Stopped                 gedit primesqr.cpp
primesqr.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 2- Prime Number$ ls
knockcat@VICKY:~/Documents/OOPS in C++/Practical 2- Prime Number$ ./primesqr
Enter Number : 97
97 is prime
knockcat@VICKY:~/Documents/OOPS in C++/Practical 2- Prime Number$ ./primesqr
Enter Number : 11
Knockcat@VICKY:~/Documents/OOPS in C++/Practical 2- Prime Number$ ./primesqr
Enter Number : 102
102 is not prime
knockcat@VICKY:~/Documents/OOPS in C++/Practical 2- Prime Number$ ./primesqr
Enter Number : 100023
100023 is not prime
knockcat@VICKY:~/Documents/OOPS in C++/Practical 2- Prime Number$ ./primesqr
Enter Number : 473
knockcat@VICKY:~/Documents/OOPS in C++/Practical 2- Prime Number$ gedit explanation.txt
[1]+  Killed                  gedit primesqr.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 2- Prime Number$ 
  
```

File Manager Window:

The file manager shows the following files in the current directory:

- Recent
- Starred
- Home
- Desktop
- Documents
- Downloads
- Music

Files listed in the current directory:

- explanation.txt
- primesqr
- primesqr.cpp

Program 3

Task 1:
Source Code:

```
#include<iostream.h>
using namespace std;
#define pi 3.14
int main()
{
    int area , r = 6;
    area = pi * r* r;
    cout<<"area = "<<area;
}

/*WE are not suppose to use the .h header file
as the preprocessor will not be able to find
the respective file and show fatal error
*/
```

Output

The screenshot shows a Linux desktop environment with a dark theme. On the left is a vertical dock with icons for a terminal, file manager, and other applications. Three windows are open:

- task1.cpp**: A code editor window showing C++ code. The code includes a preprocessor directive #include<iostream.h> instead of #include<iostream>. The code calculates the area of a circle with radius 6.
- Terminal**: A terminal window showing the command line. It shows the user navigating to a directory, opening the file with gedit, listing files, and then compiling it with g++. The compilation fails with a fatal error because the system does not have a file named iostream.h.
- Conclusion.txt**: A text editor window containing a single line of text: "If we include <iostream.h> in place of include<iostream> than the compiler will throw a fatal error that no such file or directory exist as #include<iostream.h>".

Program 3

Task 2:

Source Code:

```
using namespace std;
```

```
int main()
{
    int ind = 100;
    cout<<ind;
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a terminal window titled 'task1' with the command 'ls' running. The output shows a directory structure with files like 'task2.cpp', 'task2.i', and 'task2.o'. Below the terminal is another terminal window with the command 'g++ -E task2.cpp >task2.i' running, showing the preprocessed code. In the bottom left, there is a text editor window titled 'Conclusion.txt' containing a list of 7 points about the preprocessor. On the left side of the screen, there is a dock with various icons.

```

Oct 2 16:26
task1@VICKY: ~/Documents/OOPS in C++/Practical 3 - Preprocessor$ ls
knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor$ task2.cpp task2.i task2.o
knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor$ g++ -E task2.cpp >task2.i
# 1 "task2.cpp"
# 1 "<built-in>"
# 1 "<command-line>" 1 "/usr/include/stdc-predef.h" 1 3 4
# 1 "<command-line>" 2
# 1 "task2.cpp"

int main()
{
    int ind = 100;
    cout<<ind;
}
Knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor$ g++ -E task2.cpp >task2.i
Knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor$ g++ -S task2.i
task2.cpp: In function `int main()':
task2.cpp:9:2: error: `cout' was not declared in this scope
  9 |     cout<<ind;
   |     ^
Knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor$ knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor$ 
Knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor$ 
without including Iostream Header$ 

```

Conclusion.txt content:

- 1 How the preprocessor will react when you use cout but don't include #include<iostream> in your code?
- 2
- 3 The program will able to pass the preprocessing step , but it will not be able to pass the assembly step as the compiler will throw an error that
- 4
- 5 'cout' was not declared in the scope.
- 6
- 7 Hence the program will fail to perform further steps.....

Program 3

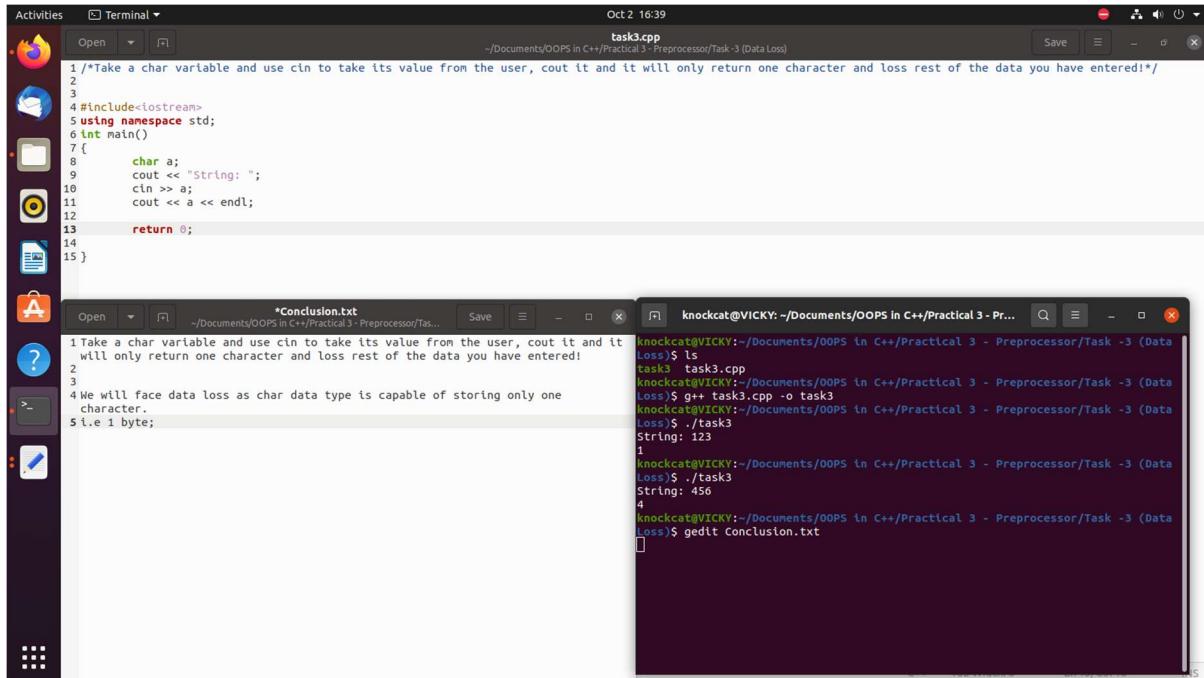
Task 3:

Source Code:

```
#include<iostream>
using namespace std;
int main()
{
    char a;
    cout << "String: ";
    cin >> a;
    cout << a << endl;

    return 0;
}
```

Output



```
Activities Terminal Oct 2 16:39 task3.cpp ~/Documents/OOPS in C++/Practical 3 - Preprocessor/Task-3 (Data Loss)
Open Save
1 /*Take a char variable and use cin to take its value from the user, cout it and it will only return one character and loss rest of the data you have entered!
2
3
4 #include<iostream>
5 using namespace std;
6 int main()
7 {
8     char a;
9     cout << "String: ";
10    cin >> a;
11    cout << a << endl;
12
13    return 0;
14
15 }
```

```
*Conclusion.txt Open Save
1 Take a char variable and use cin to take its value from the user, cout it and it will only return one character and loss rest of the data you have entered!
2
3
4 We will face data loss as char data type is capable of storing only one character.
5 i.e 1 byte;
```

```
knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor/Task -3 (Data Loss)$ ls
task3.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor/Task -3 (Data Loss)$ g++ task3.cpp -o task3
knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor/Task -3 (Data Loss)$ ./task3
String: 123
1
knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor/Task -3 (Data Loss)$ ./task3
String: 456
4
knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor/Task -3 (Data Loss)$ gedit Conclusion.txt
```

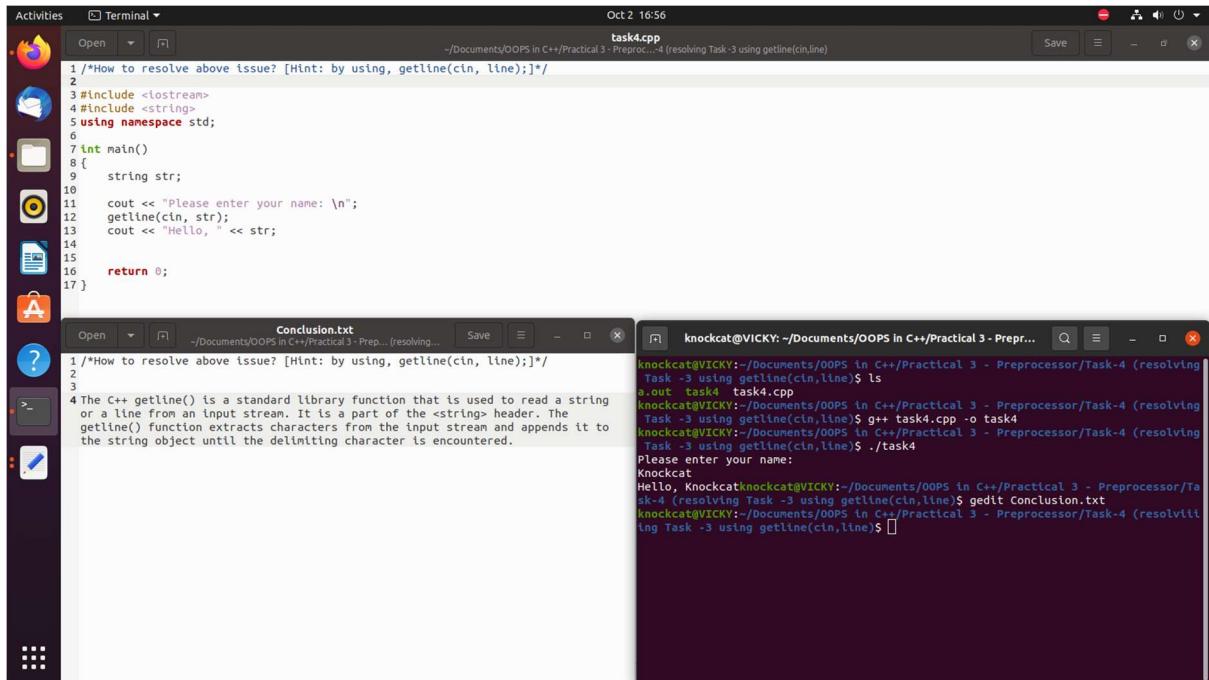
Program 3

Task 4:
Source Code:

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string str;

    cout << "Please enter your name: \n";
    getline(cin, str);
    cout << "Hello, " << str;
    return 0;
}
```

Output



The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Terminal Window:

```
Oct 2 16:56
task4.cpp
~/Documents/OOPS in C++/Practical 3 - Preproc... (resolving Task-3 using getline(cin,line))
Save   X
```

Code Editor Window:

```
Activities Terminal
Open task4.cpp
1 /*How to resolve above issue? [Hint: by using, getline(cin, line);]*/
2
3 #include <iostream>
4 #include <string>
5 using namespace std;
6
7 int main()
8 {
9     string str;
10    cout << "Please enter your name: \n";
11    getline(cin, str);
12    cout << "Hello, " << str;
13
14
15
16    return 0;
17 }
```

Conclusion.txt:

```
Conclusion.txt
Save
1 /*How to resolve above issue? [Hint: by using, getline(cin, line);]*/
2
3
4 The C++ getline() is a standard library function that is used to read a string or a line from an input stream. It is a part of the <string> header. The getline() function extracts characters from the input stream and appends it to the string object until the delimiting character is encountered.
```

Terminal Output:

```
knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor/Task-4 (resolving Task -3 using getline(cin,line)$ ls
a.out task4.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor/Task-4 (resolving Task -3 using getline(cin,line)$ g++ task4.cpp -o task4
Knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor/Task-4 (resolving Task -3 using getline(cin,line)$ ./task4
Please enter your name:
Knockcat
Hello, Knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor/Task-4 (resolving Task -3 using getline(cin,line)$ gedit Conclusion.txt
knockcat@VICKY:~/Documents/OOPS in C++/Practical 3 - Preprocessor/Task-4 (resolving Task -3 using getline(cin,line)$
```

Program 4

Task 1:

Source Code:

```
#include<iostream>
using namespace std;
namespace first{
    int add(int a, int b)
    {
        return (a+b);
    }
}
namespace second{
    float add(float a, float b)
    {
        return (a+b);
    }
}
int main()
{
    cout << "Sum of integers: " << first::add(2,3) << endl;
    cout << "Sum of float: " << second::add(2.2,3) << endl;

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. On the left is a dock with icons for a browser, file manager, terminal, and other applications. Three windows are open:

- Terminal:** Shows the command line interface with the user's session history.
- Code Editor:** Shows the C++ source code for `task1.cpp`. It contains two namespaces: `first` and `second`, each with an `add` function. The `main` function demonstrates the scope resolution operator `::` to call these functions.
- Text Editor:** Shows a text file named `Conclusion.txt` containing notes about the scope resolution operator.

```

Oct 2 17:21
task1.cpp
1 /*Access these methods using scope resolution operator [::](SRO) from main method?*/
2
3 #include<iostream>
4 using namespace std;
5 namespace first{
6     int add(int a, int b)
7     {
8         return (a+b);
9     }
10 }
11 namespace second{
12     float add(float a, float b)
13     {
14         return (a+b);
15     }
16 }
17 int main()
18 {
19     cout << "Sum of integers: " << first::add(2,3) << endl;
20     cout << "Sum of float: " << second::add(2.2,3) << endl;
21
22     return 0;
23 }

Conclusion.txt
1 Access these methods using scope resolution operator [::](SRO) from main method?
2
3
4 Namespace can be included by including using keyword with method or it can be
   included using method::funcname
5
6 :: Scope Resolution Operator

```

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 1(Acess Me
thods using Scope Resolution Operator)$ gedit task1.cpp
^Z
[1]+  Stopped                  gedit task1.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 1(Acess Me
thods using Scope Resolution Operator$ ls
task1.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 1(Acess Me
thods using Scope Resolution Operator$ g++ task1.cpp -o task1
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 1(Acess Me
thods using Scope Resolution Operator$ ./task1
Sum of integers: 5
Sum of float: 5.2
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 1(Acess Me
thods using Scope Resolution Operator$ gedit Conclusion.txt

```

Program 4

Task 2:

Source Code:

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

namespace first
{
    void saysomething()
    {
        cout<<"1.. first namespace      Hello world\n";
    }
}

using namespace first;

namespace second
{
    void saysomething()
    {
        cout<<"2.. seccond namespace      Hello world\n";
    }
}

int main()
{
    first::saysomething();
    saysomething();
    second::saysomething();
    return 0;
}
```

Output

The screenshot shows a Linux desktop interface with three open windows:

- Terminal Window:** Shows the command line output of a C++ program named "task2". The program prints "Hello world" twice, once from each namespace. The terminal window title is "task2.cpp".
- File Browser Window:** Shows a file tree under "/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 2 (Access methods by including "using" keyword)". It contains files "task2.cpp", "task2.o", and "task2".
- Text Editor Window:** Shows two files: "task2.cpp" and "Conclusion.txt". "task2.cpp" contains the C++ code for the namespaces and their methods. "Conclusion.txt" contains notes about the "using" keyword.

```

task2.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 2 (Access methods by including "using" keyword)$ ls
task2 task2.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 2 (Access methods by including "using" keyword)$ g++ task2.cpp -o task2
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 2 (Access methods by including "using" keyword)$ ./task2
1.. first namespace Hello world
1.. first namespace Hello world
2.. second namespace Hello world
2.. second namespace Hello world
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 2 (Access methods by including "using" keyword)$ gedit Conclusion.txt
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 2 (Access methods by including "using" keyword)$

Conclusion.txt
*Conclusion.txt
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 2 (Access methods by including "using" keyword)$ cat Conclusion.txt
1 Access these methods using "using" keyword for main method?
2
3 On using "using" keyword we do not need to include namespace by help of scope resolution operator.
4
5 using namespace name_of_namespace
6
7

```

Program 4

Task 3:

Source Code:

```
#include<iostream>
using namespace std;
namespace first{
    int add(int a, int b)
    {
        return (a+b);
    }
}
namespace second{
    float add(float a, float b)
    {
        return (a+b);
    }
}

using namespace first;
using namespace second;

int main()
{
    cout << add(2,3) << endl;
    cout << add(22,3) << endl;
    cout << add(9.3f,10.1f)<< endl;
    cout << add(5.5f,4.1f)<< endl;
    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Task3.cpp):

```

1 /*Try to access these methods without using, (SRO) and "using" keyword and check how the compiler will react to it?*/
2
3 #include<iostream>
4 using namespace std;
5 namespace first{
6     int add(int a, int b);
7     {
8         return (a+b);
9     }
10 }
11 namespace second{
12     float add(float a, float b)
13     {
14         return (a+b);
15     }
16 }
17
18 using namespace first;
19 using namespace second;
20
21 int main()
22 {
23     cout << add(2,3) << endl;
24     cout << add(2.2,3) << endl;
25     cout << add(9.3f,10.1f)<< endl;
26     cout << add(5.5f,4.1f)<< endl;
27     return 0;
28 }
29

```

Terminal (knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 3 (Access Method Without using (sro))\$ ls

```

Task3 Task3.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 3 (Access Method Without using (sro))$ ls
Task3 Task3.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 3 (Access Method Without using (sro))$ g++ Task3.cpp -o Task3
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 3 (Access Method Without using (sro))$ ./Task3
5
25
19.4
9.6
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 3 (Access Method Without using (sro))$ gedit Conclusion.txt
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)/Task - 3 (Access Method Without using (sro))$ 

```

Conclusion.txt:

```

1 Try to access these methods without using, (SRO) and "using" keyword and check
2 how the compiler will react to it?
3 The Compiler Worked Fine!

```

Program 4

Task 4:

Source Code:

```
#include<iostream>
using namespace std;
namespace first{
    int add(int a, int b)
    {
        return (a+b);
    }
}
namespace second{
    float add(float a, float b)
    {
        return (a+b);
    }
}

using namespace first;
using namespace second;

int main()
{
    cout << add(2,3) << endl;
    cout << add(2.2f,3) << endl;
    cout << add(0.3f,10)<< endl;
    cout << add(5.5f,4.1f)<< endl;
    return 0;
}
```

Output

```

Oct 2 18:47
Task4.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical - 4 (Namespace)~$ g++ Task4.cpp -o Task4
Task4.cpp: In function 'int main()':
Task4.cpp:24:20: error: call of overloaded 'add(float, int)' is ambiguous
  24 |     cout << add(2.2f,3) << endl;
               ^~~~~
Task4.cpp:6:6: note: candidate: 'int first::add(int, int)'
  6 |     int add(int a, int b)
               ^~~~
Task4.cpp:12:8: note: candidate: 'float second::add(float, float)'
 12 |     float add(float a, float b)
               ^~~~
Task4.cpp:25:21: error: call of overloaded 'add(float, int)' is ambiguous
 25 |     cout << add(0.3f,10) << endl;
               ^~~~~
Task4.cpp:6:6: note: candidate: 'int first::add(int, int)'
  6 |     int add(int a, int b)
               ^~~~
Task4.cpp:12:8: note: candidate: 'float second::add(float, float)'
 12 |     float add(float a, float b)
               ^~~~

*Conclusion.txt*
try to access these methods for Mixed Values [Int, Float] and see how the compiler will react to it?
Error Occur because no declared method for float , int parameter, and hence ambiguity occurs
If you use a name or qualified name that does not refer to a unique function or object. This is called Ambiguity
OR
When you derive classes, ambiguities can result if base and derived classes have members with the same names. Access to a base class member is ambiguous if you use a name or qualified name that does not refer to a unique function or object.

```

Program 5

Task 1:

Source Code:

```
#include<iostream>
using namespace std;
/*char short int long float double long double wide char*/

void char_d()
{
    char c = 67;
    cout<<c<<endl;
}

void bool_d()
{
    bool a = true;
    cout<<a<<endl;
}

void short_d()
{
    short a = 34;
    cout<<a<<endl;
}

void int_d()
{
    int b = 90;
    cout<<b<<endl;
}

void long_d()
{
    long int a = 89;
    cout<<a<<endl;
}

void float_d()
```

```

{
    float f = 32.78;
    cout<<f<<endl;
}

void double_d()
{
    double c = 78.43;
    cout<<c<<endl;
}

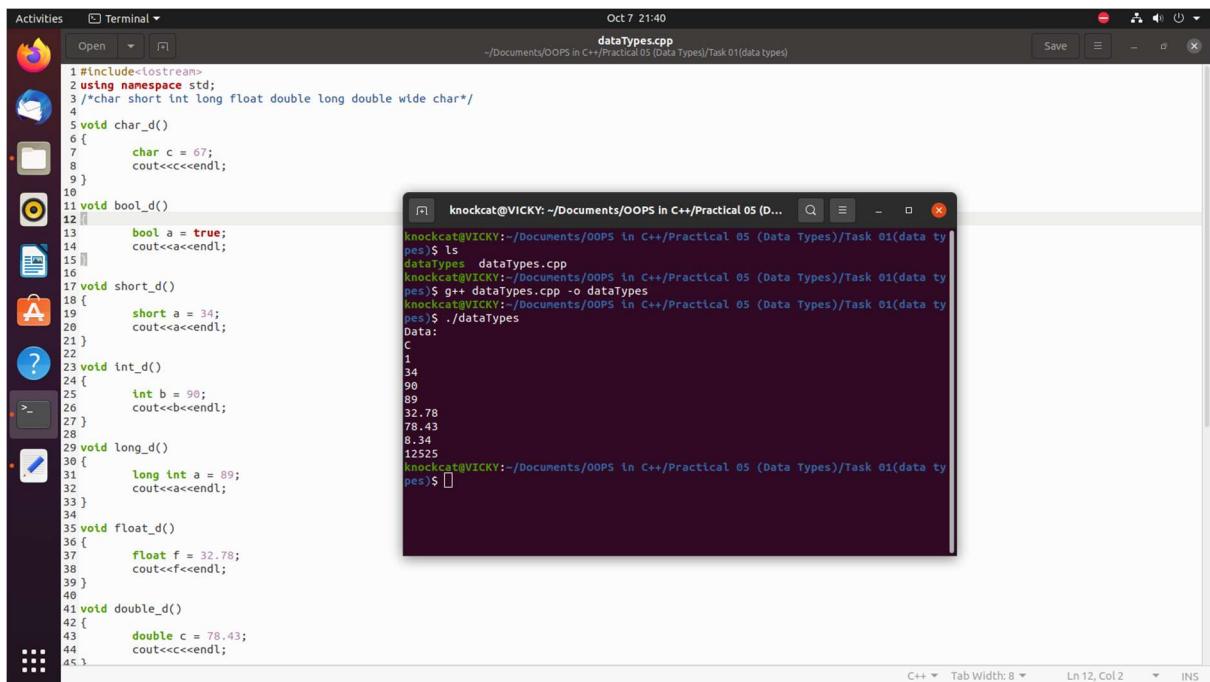
void long_double_d()
{
    long double a = 8.34;
    cout<<a<<endl;
}

void wide_char()
{
    wchar_t g = L'¤';
    cout<<g<<endl;
}

int main()
{
    cout<<"Data:"<<endl;
    char_d();
    bool_d();
    short_d();
    int_d();
    long_d();
    float_d();
    double_d();
    long_double_d();
    wide_char();
    return 0;
}

```

Output



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "dataTypes.cpp" and it displays the following output:

```

knockcat@VICKY: ~/Documents/OOPS in C++/Practical 05 (Data Types)/Task 01(data types)
knockcat@VICKY:~/Documents/OOPS in C++/Practical 05 (Data Types)/Task 01(data types)$ ls
dataTypes  dataTypees.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 05 (Data Types)/Task 01(data types)$ g++ dataTypees.cpp -o dataTypees
knockcat@VICKY:~/Documents/OOPS in C++/Practical 05 (Data Types)/Task 01(data types)$ ./dataTypees
Data:
C
1
34
90
89
32.78
78.43
8.34
12525
knockcat@VICKY:~/Documents/OOPS in C++/Practical 05 (Data Types)/Task 01(data types)$ 

```

The terminal window is part of a desktop environment with a dock containing various icons like a file manager, terminal, and system monitor.

Program 5

Task 2:
Source Code:

```
#include<iostream>
using namespace std;
/*char short int long float double long double wide char*/

void char_d()
{
    char c = 67;
    cout<<c<<"  "<<sizeof(char)<<"\t\t";
    cout<<sizeof(c)<<endl;
}

void bool_d()
{
    bool a = true;
    cout<<a<<"  "<<sizeof(bool)<<"\t\t";
    cout<<sizeof(a)<<endl;
}

void short_d()
{
    short a = 34;
    cout<<a<<"  "<<sizeof(short)<<"\t\t";
    cout<<sizeof(a)<<endl;
}

void int_d()
{
    int b = 90;
    cout<<b<<"  "<<sizeof(int)<<"\t\t";
    cout<<sizeof(b)<<endl;
}
```

```

}

void long_d()
{
    long int a = 89;
    cout<<a<<"  "<<sizeof(long int)<<"\t\t";
    cout<<sizeof(a)<<endl;
}

void float_d()
{
    float f = 32.78;
    cout<<f<<"  "<<sizeof(float)<<"\t\t";
    cout<<sizeof(f)<<endl;
}

void double_d()
{
    double c = 78.43;
    cout<<c<<"  "<<sizeof(double)<<"\t\t";
    cout<<sizeof(c)<<endl;
}

void long_double_d()
{
    long double a = 8.34;
    cout<<a<<"  "<<sizeof(long double)<<"\t\t";
    cout<<sizeof(a)<<endl;
}

void wide_char()
{
    wchar_t g = L'□';
    cout<<g<<"  "<<sizeof(wchar_t)<<"\t\t";
    cout<<sizeof(g)<<endl;
    wcout<<g<<endl;
}

```

```
int main()
{
    cout<<"Data Size(in bytes)\tSize (var)"<<endl;
    char_d();
    bool_d();
    short_d();
    int_d();
    long_d();
    float_d();
    double_d();
    long_double_d();
    wide_char();
    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Activities Terminal):

```

1 #include<iostream>
2 using namespace std;
3 /*char short int long float double long double wide char*/
4
5 void char_d()
6 {
7     char c = 67;      "<<sizeof(char)<<\t\t";
8     cout<<c<<"\t\t";
9     cout<<sizeof(c)<<endl;
10 }
11
12 void bool_d()
13 {
14     bool a = true;   "<<sizeof(bool)<<\t\t";
15     cout<<a<<"\t\t";
16     cout<<sizeof(a)<<endl;
17 }
18
19 void short_d()
20 {
21     short a = 34;   "<<sizeof(short)<<\t\t";
22     cout<<a<<"\t\t";
23     cout<<sizeof(a)<<endl;
24 }
25
26 void int_d()
27 {
28     int b = 90;     "<<sizeof(int)<<\t\t";
29     cout<<b<<"\t\t";
30     cout<<sizeof(b)<<endl;
31 }
32
33 void long_d()
34 {
35     long int a = 89;    "<<sizeof(long int)<<\t\t";
36     cout<<a<<"\t\t";
37     cout<<sizeof(a)<<endl;
38 }
39
40 void float_d()
41 {
42     float f = 32.78;  "<<sizeof(float)<<\t\t";
43     cout<<f<<"\t\t";
44     cout<<sizeof(f)<<endl;
45 }

```

Terminal (knockcat@VICKY: ~/Documents/OOPS in C++/Practical 05 (Data Types)/Task 02 (sizeof)):

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 05 (Data Types)/Task 02 (sizeof)
$ ls
sizeof_data_types sizeof_data_types.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 05 (Data Types)/Task 02 (sizeof)
$ g++ sizeof_data_types.cpp -o sizeof_data_types
knockcat@VICKY:~/Documents/OOPS in C++/Practical 05 (Data Types)/Task 02 (sizeof)
$ ./sizeof_data_types
Data  Size(in bytes)  Size (var)
C       1                  1
1       1                  1
34      2                  2
90      4                  4
89      8                  8
32.78   4                  4
78.43   8                  8
8.34    16                 16
12525   4                  4
*
knockcat@VICKY:~/Documents/OOPS in C++/Practical 05 (Data Types)/Task 02 (sizeof)
$ []

```

Program 6

Task 1:
Source Code:

```
#include<iostream>

void clear(void)
{
    while(getchar() != '\n'); //To clear the buffer when system not taking string as
input.....
}

using namespace std;
int main()
{
    int a,b;
    float c,d;
    string s1, s2;

    cout << "Enter int values: " << endl;
    cin >> a >> b;

    cout << "Enter float values: " << endl;
    cin >> c >> d;

    cout << "Enter first string: " << endl;
    clear();
    getline(cin,s1);

    cout << "Enter second string: " << endl;
    getline(cin,s2);

    cout << "Int + Int = " << a + b << endl;
    cout << "float + float = " << c + d << endl;
    cout << "string + string = " << s1 + " " + s2 << endl;

    cout << "Int + float =" << a + c << endl;
    cout << "Int + String =" << a + s1 << endl;
    return 0;}
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Terminal Window:

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 01 (string + int (ERROR))
Task01.cpp
Task01.cpp:36:32: error: no match for 'operator+' (operand types are 'int' and 'std::string' (aka 'std::cxx11::basic_string<char>'))
  36 |     cout << "Int + String =" << a >> s1 << endl;
      |           ^ ~~~~~
      |           int std::string (aka std::cxx11::basic_st
ringchar>)
In file included from /usr/include/c++/9/bits/stl_algobase.h:67,
                 from /usr/include/c++/9/bits/char_traits.h:39,
                 from /usr/include/c++/9/ios:40,
                 from /usr/include/c++/9/ostream:38,
                 from /usr/include/c++/9/ostream:39,
                 from Task01.cpp:3:
/usr/include/c++/9/bits/stl_iterator.h:423:5: note: candidate: 'template<class _Iterator> std::reverse_iterator<_Iterator>::operator+(typename std::reverse_iterator<_Iterator>::difference_type, const std::reverse_iterator<_Iterator>::difference_type __n,
423 |     operator+(typename reverse_iterator<_Iterator>::difference_type __n,

```

Code Editor Window:

```

1 //To check the output when int + string is given...
2 //">#include<bits/stdc++.h> this can be used for all the libraries..
3 #include<iostream>
4
5 void clear(void)
6 {
7     while(getchar() != '\n'); //To clear the buffer when system not taking string as input...
8 }
9
10 using namespace std;
11 int main()
12 {
13     int a,b;
14     float c,d;
15     string s1, s2;
16
17     cout << "Enter int values: " << endl;
18     cin >> a >> b;
19
20     cout << "Enter float values: " << endl;
21     cin >> c >> d;
22
23     cout << "Enter first string: " << endl;
24     clear();
25     getline(cin,s1);
26
27     cout << "Enter second string: " << endl;
28     getline(cin,s2);
29
30     cout << "Int + Int = " << a + b << endl;
31     cout << "float + float = " << c + d << endl;
32     cout << "String + string = " << s1 + " " + s2 << endl;
33
34     cout << "Int + float =" << a + c << endl;
35     cout << "Int + String =" << a + s1 << endl;
36
37     return 0;
38 }

```

Program 6

Task 2:

Source Code:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    string s,temp = "";
    cout << "Enter a string: ";
    getline(cin,s);

    string :: iterator it;
    for(it = s.begin(); it != s.end(); it++)
    {
        temp = *it + temp;
    }
    if(temp == s)
        cout << "String is Palindrome...." << endl;
    else
        cout << "String is not Palindrome...." << endl;
    return 0;
}
```

Output

The screenshot shows a terminal window titled "Task02.cpp" running on a Linux desktop. The terminal displays the source code for a C++ program that checks if a string is a palindrome using reverse iterators. The user enters three strings: "knockcat", "Malayala", and "ARORA". The program outputs "String is Palindrome..." for "knockcat" and "ARORA", and "String is not Palindrome..." for "Malayala".

```

Activities Terminal
Open ... Save ...
Oct 7 21:48
Task02.cpp
~/Documents/OOPS in C++/Practical 06 (String)...Task 02 (Palindrome (using reverse_iterator))
Task02.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 02 (Palindrome (using reverse_iterator))$ ls
Task02.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 02 (Palindrome (using reverse_iterator))$ g++ Task02.cpp -o Task02
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 02 (Palindrome (using reverse_iterator))$ ./Task02
Enter a string: knockcat
String is not Palindrome...
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 02 (Palindrome (using reverse_iterator))$ ./Task02
Enter a string: Malayala
String is not Palindrome...
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 02 (Palindrome (using reverse_iterator))$ ./Task02
Enter a string: ARORA
String is Palindrome...
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 02 (Palindrome (using reverse_iterator))$ 
```

Program 6

Task 3:

Source Code:

```
//To reverse the string....  
#include<iostream>  
using namespace std;  
int main()  
{  
    string s,rev;  
    cout << "\nEnter String: ";  
    getline(cin,s);  
  
    string :: iterator it;  
    for(it = s.begin(); it != s.end(); it++)  
        rev = *it + rev;  
  
    cout << "\nReversed string is: " << rev << endl;  
    return 0;  
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. On the left is a vertical dock containing icons for various applications like a file manager, terminal, and system settings. The main area has two windows: a terminal window titled 'knockcat@VICKY:' and a code editor window titled 'Task03.cpp'.

Code Editor (Task03.cpp):

```

1 //To reverse the string...
2 #include<iostream>
3 using namespace std;
4 int main()
5 {
6     string s,rev;
7     cout << "\nEnter String: ";
8     getline(cin,s);
9
10    string :: iterator it;
11    for(it = s.begin(); it != s.end(); it++)
12        rev = *it + rev;
13
14    cout << "\nReversed string is: " << rev << endl;
15    return 0;
16 }

```

Terminal (knockcat@VICKY: ~/Documents/OOPS in C++/Practical 06 (String)/Task 03 (Reverse of string (using reverse_iterator))\$ ls

```

Task03.cpp

```

Terminal (knockcat@VICKY: ~/Documents/OOPS in C++/Practical 06 (String)/Task 03 (Reverse of string (using reverse_iterator))\$ g++ Task03.cpp -o Task03

Terminal (knockcat@VICKY: ~/Documents/OOPS in C++/Practical 06 (String)/Task 03 (Reverse of string (using reverse_iterator))\$./Task03

Enter String: VISHAL

Reversed string is: LAHSIV

Program 6

Task 4:
Source Code:

```
//Comparing two strings....
#include<bits/stdc++.h>
using namespace std;
void clear(void)
{
    while(getchar() != '\n');
}

int main()
{
    string s1, s2;
    cout << "Enter first string: ";
    //clear();
    getline(cin,s1);
    cout << "Enter second string: ";
    getline(cin,s2);

    int size1 = s1.size();
    int size2 = s2.size();

    if( size1 != size2)
    {
        cout << "Given Strings are not equal...." << endl;
        return 0;
    }
    for(int i = 0; i < size1; i++)
        if(s1[i] != s2[i])
    {
        cout << "Given Strings are not equal...." << endl;
        return 0;
    }
    cout << "Given Strings are equal...." << endl;
    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. On the left is a dock with various icons. In the center, there's a terminal window titled "Task04.cpp" showing the source code for a C++ program to compare two strings manually. Below it is a code editor window titled "knockcat@VICKY: ~/Documents/OOPS in C++/Practical 06 (String)/Task 04 (compare string manually)" displaying the same code and its execution output.

```

Activities Terminal
Oct 7 21:54
Task04.cpp
~/Documents/OOPS in C++/Practical 06 (String)/Task 04 (compare string manually)
Save

knockcat@VICKY: ~/Documents/OOPS in C++/Practical 06 (String)/Task 04 (compare string manually)$ ls
Task04.cpp
knockcat@VICKY: ~/Documents/OOPS in C++/Practical 06 (String)/Task 04 (compare string manually)$ g++ Task04.cpp -o Task04
knockcat@VICKY: ~/Documents/OOPS in C++/Practical 06 (String)/Task 04 (compare string manually)$ ./Task04
Enter first string: Hello
Enter second string: World
Given Strings are not equal...
knockcat@VICKY: ~/Documents/OOPS in C++/Practical 06 (String)/Task 04 (compare string manually)$ ./Task04
Enter first string: knockcat
Enter second string: knockcat
Given Strings are equal...
knockcat@VICKY: ~/Documents/OOPS in C++/Practical 06 (String)/Task 04 (compare string manually)$

```

```

1 //Comparing two strings...
2 #include<iostream>
3 using namespace std;
4 void clear(void)
5 {
6     while(getchar() != '\n');
7 }
8
9 int main()
10 {
11     string s1, s2;
12     cout << "Enter first string: ";
13     //clear();
14     getline(cin,s1);
15     cout << "Enter second string: ";
16     getline(cin,s2);
17
18     int size1 = s1.size();
19     int size2 = s2.size();
20
21     if( size1 != size2 )
22     {
23         cout << "Given Strings are not equal...." << endl;
24         return 0;
25     }
26     for(int i = 0; i < size1; i++)
27     {
28         if(s1[i] != s2[i])
29         {
30             cout << "Given Strings are not equal...." << endl;
31             return 0;
32         }
33     }
34     cout << "Given Strings are equal...." << endl;
35     return 0;
36 }

```

Program 6

Task 5:

Source Code:

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

int main()
{
    string a;
    string b;
    cout << "Enter first string: ";
    getline(cin,a);
    cout << "Enter second string: ";
    getline(cin,b);

    cout << "On comparing String 1 and String 2, we get: " << a.compare(b) <<
    endl;

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. On the left is a dock with various icons. In the center, there's a terminal window titled "Task05.cpp" showing the source code for a C++ program that compares two strings using `string.compare()`. Below it is a terminal window showing the execution of the program, where the user inputs "Squery" and "Swift" and gets a result of -1. The terminal also shows the compilation command: `g++ Task05.cpp -o Task05`.

```

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6
7     string a;
8     string b;
9     cout << "Enter first string: ";
10    getline(cin,a);
11    cout << "Enter second string: ";
12    getline(cin,b);
13
14    cout << "On comparing String 1 and String 2, we get: " << a.compare(b) << endl;
15
16
17
18    return 0;
19 }

```

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 05 (string compare using string.compare())$ ls
Task05.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 05 (string compare using string.compare())$ g++ Task05.cpp -o Task05
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 05 (string compare using string.compare())$ ./Task05
Enter first string: Squery
Enter second string: Swift
On comparing String 1 and String 2, we get: -1
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 05 (string compare using string.compare())$ ./Task05
Enter first string: Sql
Enter second string: Sql
On comparing String 1 and String 2, we get: 0
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 05 (string compare using string.compare())$ ./Task05
Enter first string: C++
Enter second string: C
On comparing String 1 and String 2, we get: 2
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 05 (string compare using string.compare())$ 

```

Program 6

Task 6:

Source Code:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    string str;
    str = "Hello";
    cout<<&str<<endl;
    cout << str << endl;

    str[0] = 'J';
    cout<<&str<<endl;
    cout << str << endl;
    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Task06.cpp" and the path is "/Documents/OOPS in C++/Practical 06 (String)/Task 06 (Check if string is mutable or not)". The code in the terminal is:

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int main()
4 {
5     string str;
6     str = "Hello";
7     cout<<str<<endl;
8     cout << str << endl;
9
10    str[0] = 'J';
11    cout<<&str<<endl;
12    cout << str << endl;
13
14    return 0;
}
```

The terminal output shows the execution of the program:

```
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 06 (Check if s
tring is mutable or not)$ ls
Task06.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 06 (Check if s
tring is mutable or not)$ g++ Task06.cpp -o Task06
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 06 (Check if s
tring is mutable or not)$ ./Task06
0xffffe6123d1e0
Hello
0xffffe6123d1e0
Jello
knockcat@VICKY:~/Documents/OOPS in C++/Practical 06 (String)/Task 06 (Check if s
tring is mutable or not)$
```

The terminal interface includes a file manager sidebar on the left and various status indicators at the bottom.

Program 7

Task 1:
Source Code:

```
//Time and date Manually
#include<iostream>
using namespace std;
int main()
{
    int n;
    char ch;

    do{

        cout << "\nEnter choice you want to print: \n" << endl;
        cout << "1.Year 2.Month 3.Day" << endl;
        cin >> n;

        switch(n)
        {
            case 1:
                cout << "\n2021\n";
                break;
            case 2:
                cout << "\nOctober\n";
                break;
            case 3:
                cout << "\nThursday\n";
                break;
            default:
                cout << "\nInvalid choice....Enter valid data\n";
                break;
        }
    } while(ch != 'q');
}
```

```
    }  
    cout<<"Do you want to enter more (Y) or (N) :";  
    cin>>ch;  
}while(ch == 'Y' || ch == 'y');  
  
return 0;  
}
```

Output

The screenshot shows a terminal window with the following details:

- Title Bar:** Activities Terminal
- Time:** Tue Oct 19 23:13
- Path:** -/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 01 (Time and Date Manual)
- Code Content:**

```
1 //Time and date Manually
2 #include<iostream>
3 using namespace std;
4
5 int main()
6 {
7     int n;
8     char ch;
9
10    do{
11
12        cout << "\nEnter choice you want to print: \n" << endl;
13        cout << "1.Year 2.Month 3.Day" << endl;
14        cin >> n;
15
16        switch(n)
17        {
18            case 1:
19                cout << "\n2021\n";
20                break;
21            case 2:
22                cout << "\nOctober\n";
23                break;
24            case 3:
25                cout << "\nThursday\n";
26                break;
27            default:
28                cout << "\nInvalid choice...Enter valid data\n";
29                break;
30
31        }
32
33        cout<<"Do you want to enter more (Y) or (N) :";
34        cin>>ch;
35    }while(ch == 'Y' || ch == 'y');
36
37    return 0;
38
39
40 }
```
- Terminal Output:**

```
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 01 ( Time and Date Manual )$ ls
time_and_date.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 01 ( Time and Date Manual )$ g++ time_and_date.cpp -o time_and_date
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 01 ( Time and Date Manual )$ ./time_and_date
Enter choice you want to print:
1.Year 2.Month 3.Day
1
2021
Do you want to enter more (Y) or (N) :
You want to enter more (Y) or (N) :
Enter choice you want to print:
1.Year 2.Month 3.Day
2
October
Do you want to enter more (Y) or (N) :
Enter choice you want to print:
1.Year 2.Month 3.Day
3
Thursday
Do you want to enter more (Y) or (N) :
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 01 ( Time and Date Manual )$
```

Program 7

Task 2:
Source Code:

```

/*Create a switch statement [Using ctime]*/

#include <iostream>
#include <ctime>
using namespace std;

int main()
{
    time_t curtime;

    time(&curtime);
    int choice;
    cout << "1. Print current year." << endl;
    cout << "2. Print current month." << endl;
    cout << "3. Print current day." << endl;
    cout << "4. Print current Time." << endl;
    cout << "Enter your choice : ";
    cin >> choice;

    string day[7] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday"};
    string month[12] =
{"January", "February", "March", "April", "May", "June", "July", "August", "September",
"October", "November", "December"};
    time_t res = time(0);
    struct tm *t = localtime(&res);

    switch(choice)
    {
        case 1:   cout << "Current year is " << t->tm_year + 1900 << endl;

```

```
break;

case 2: cout << "Current month is " << month[t->tm_mon] << endl;
break;

case 3:
    cout << "Current day is " << day[t->tm_wday] << endl;
    break;

case 4:
    printf("Current time = %s", ctime(&curtime));
    break;

default: cout << "Invalid Input !!!" << endl;
break;
}

return 0;
}
```

Output

```

Activities  Text Editor ▾ Tue Oct 19 23:15
Open  *time_and_date.cpp  Save  ...
~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 02 (Time and Date ...
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 02 (Tl
me and Date Automatic)$ ls
time_and_date  time_and_date.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 02 (Tl
me and Date Automatic)$ g++ time_and_date.cpp -o time_and_date
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 02 (Tl
me and Date Automatic)$ ./time_and_date
1. Print current year.
2. Print current month.
3. Print current day.
4. Print current Time.
Enter your choice : 1
Current year is 2021
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 02 (Tl
me and Date Automatic)$ ./time_and_date
1. Print current year.
2. Print current month.
3. Print current day.
4. Print current Time.
Enter your choice : 2
Current month is October
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 02 (Tl
me and Date Automatic)$ ./time_and_date
1. Print current year.
2. Print current month.
3. Print current day.
4. Print current Time.
Enter your choice : 3
Current day is Tuesday
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 02 (Tl
me and Date Automatic)$ ./time_and_date
1. Print current year.
2. Print current month.
3. Print current day.
4. Print current Time.
Enter your choice : 4
Current time = Tue Oct 19 23:14:13 2021
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 02 (Tl
me and Date Automatic$ ./time_and_date
1. Print current year.
2. Print current month.
3. Print current day.
4. Print current Time.
Enter your choice :

```

Program 7

Task 3 v1:
Source Code:

```

/* v1. Print using reverse method:*/

#include<iostream>
using namespace std;
int main()
{
    int arr[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            cout<<arr[i][j]<<" ";
        }
        cout<<endl;
    }

    cout<<"\nReversed Array\n\n";

    for(int i = 2; i >= 0; i--)
    {
        for(int j = 2; j >= 0; j--)
        {
            cout<<arr[i][j]<<" ";
        }
        cout<<endl;
    }

    return 0;
}

```

Output

The screenshot shows a terminal window with the following details:

- Title Bar:** Activities Terminal ▾ Tue Oct 19 23:16 -/Documents/OOPS in C++/Practical 07 (Array s...g 10)/Task V01 (Print Using Reverse Method)
- Code Area (Left):**

```

1 // v1. Print using reverse method*/
2
3 #include<iostream>
4 using namespace std;
5 int main()
6 {
7     int arr[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
8     for(int i = 0; i < 3; i++)
9     {
10         for(int j = 0; j < 3; j++)
11         {
12             cout<<arr[i][j]<< " ";
13         }
14         cout<<endl;
15     }
16     cout<<"\nReversed Array\n\n";
17     for(int i = 2; i >= 0; i--)
18     {
19         for(int j = 2; j >= 0; j--)
20         {
21             cout<<arr[i][j]<< " ";
22         }
23         cout<<endl;
24     }
25     return 0;
26 }
27
28 }
```
- Output Area (Right):**

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V01 ( Print Using Reverse Method )$ ls
v01 V01.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V01 ( Print Using Reverse Method )$ g++ V01.cpp -o V01
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V01 ( Print Using Reverse Method )$ ./V01
1 2 3
4 5 6
7 8 9

Reversed Array

9 8 7
6 5 4
3 2 1
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V01 ( Print Using Reverse Method )$ 
```
- Bottom Status Bar:** C++ ▾ Tab Width: 8 ▾ Ln 30, Col 1 ▾ INS

Program 7

Task 3 v2:
Source Code:

```

//Rever array by subtracting 10

#include<iostream>
using namespace std;
int main()
{
    int arr[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            cout<<arr[i][j]<<" ";
        }
        cout<<endl;
    }

    cout<<"\nReversed Array\n\n";

    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            cout<<10-arr[i][j]<<" ";
        }
        cout<<endl;
    }

    return 0;
}

```

Output

The screenshot shows a terminal window with the following content:

```

Activities Terminal ▾ Tue Oct 19 23:18
Open ... V02.cpp ~/Documents/OOPS in C++/Practical 07 (Array &...sk V02 ( Print using (10 - arr[i][j]) method )
1 //Reverse array by subtracting 10
2
3 #include<iostream>
4 using namespace std;
5 int main()
6 {
7     int arr[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
8     for(int i = 0; i < 3; i++)
9     {
10         for(int j = 0; j < 3; j++)
11         {
12             cout<<arr[i][j]<<" ";
13         }
14         cout<<endl;
15     }
16     cout<<"\nReversed Array\n";
17     for(int i = 0; i < 3; i++)
18     {
19         for(int j = 0; j < 3; j++)
20         {
21             cout<<10-arr[i][j]<<" ";
22         }
23         cout<<endl;
24     }
25     return 0;
26 }
27
28
29 }
```

The terminal output shows the execution of the program and its results:

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V02 ( Print using (10 - arr[i][j]) method )$ ls
V02 V02.cpp
knockcat@VICKY:~/documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V02 ( Print using (10 - arr[i][j]) method )$ g++ V02.cpp -
o V02
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V02 ( Print using (10 - arr[i][j]) method )$ ./V02
1 2 3
4 5 6
7 8 9

Reversed Array
9 8 7
6 5 4
3 2 1
```

At the bottom right of the terminal window, there is a configuration menu with the following options:

- Display line numbers
- Display right margin
- Highlight current line
- Text wrapping

Below the configuration menu, the status bar shows "C++" and "Tab Width: 8". The cursor position is "Ln 18, Col 9" and the mode is "INS".

Program 7

Task 3 v3:
Source Code:

/*v3. Restore using reverse method [without creating new array]:*/

```
#include<iostream>
using namespace std;
int main()
{
    int n = 3, m = 3;
    int arr[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            cout<<arr[i][j]<<" ";
        }
        cout<<endl;
    }

    cout<<"\nReversed Array\n\n";

    int res = n % 2 == 0 ? n/2 - 1 : n/2;
    for(int i = 0; i <= res; i++)
    {
        for(int j = 0; j < n; j++)
        {
            if(i == n-1-i && j == n-1-j)
                break;
            int temp = arr[i][j];
            arr[i][j] = arr[n-1-i][n-1-j];
            arr[n-1-i][n-1-j] = temp;
        }
    }
}
```

```
for(int i = 0; i < n; i++)  
{  
    for(int j = 0; j < n; j++)  
        cout << arr[i][j] << " ";  
    cout << endl;  
}  
  
return 0;  
}
```

Output

The screenshot shows a terminal window with two panes. The left pane displays the source code for a C++ program named V03.cpp. The right pane shows the terminal output where the program is run and its output is displayed.

```

Activities Terminal ▾ Tue Oct 19 23:19
Open - /Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
Save ...
```

```

1 //V3. Restore using reverse method [without creating new array]*/
2
3 #include<iostream>
4 using namespace std;
5 int main()
6 {
7     int n = 3, m = 3;
8     int arr[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
9     for(int i = 0; i < 3; i++)
10    {
11        for(int j = 0; j < 3; j++)
12        {
13            cout<<arr[i][j]<<" ";
14        }
15        cout<<endl;
16    }
17    cout<<"\nReversed Array\n";
18
19    int res = n % 2 == 0 ? n/2 : 1 : n/2;
20    for(int i = 0; i <= res; i++)
21    {
22        for(int j = 0; j < n; j++)
23        {
24            if(i == n-1-i && j == n-1-j)
25                break;
26            int temp = arr[i][j];
27            arr[i][j] = arr[n-1-i][n-1-j];
28            arr[n-1-i][n-1-j] = temp;
29        }
30    }
31
32
33    for(int i = 0; i < n; i++)
34    {
35        for(int j = 0; j < n; j++)
36            cout << arr[i][j] << " ";
37        cout << endl;
38    }
39
40    return 0;
41
42 }
```

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V03 ( Print using reverse method, wthout creating new arr
y )$ ls
V03 V03.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V03 ( Print using reverse method, wthout creating new arr
y )$ g++ V03.cpp -o V03
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V03 ( Print using reverse method, wthout creating new arr
y )$ ./V03
1 2 3
6 5 4
3 2 1
Reversed Array
9 8 7
6 5 4
3 2 1
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V03 ( Print using reverse method, wthout creating new arr
y )$ |
```

C++ ▾ Tab Width: 8 ▾ Ln 43, Col 1 ▾ INS

Program 7

Task 3 v4:

Source Code:

```
#include<iostream>
using namespace std;
int main()
{
    int arr[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            cout<<arr[i][j]<<" ";
        }
        cout<<endl;
    }

    cout<<"\nReversed Array\n\n";

    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            arr[i][j] = 10 - arr[i][j];
            cout<<arr[i][j]<<" ";
        }
        cout<<endl;
    }

    return 0;
}
```

Output

The screenshot shows a terminal window with the following details:

- Title Bar:** Activities Terminal ▾ Tue Oct 19 23:21
- File Path:** ~/Documents/OOPS in C++/Practical 07 (Array &... 10 - arr[i][j] , without creating new array)
- Code (V04.cpp):**

```

1 /*v3. Restore using reverse method [without creating new array]:*/
2
3 #include<iostream>
4 using namespace std;
5 int main()
6 {
7     int arr[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
8     for(int i = 0; i < 3; i++)
9     {
10         for(int j = 0; j < 3; j++)
11         {
12             cout<<arr[i][j]<<" ";
13         }
14         cout<<endl;
15     }
16     cout<<"\nReversed Array\n";
17     for(int i = 0; i < 3; i++)
18     {
19         for(int j = 0; j < 3; j++)
20         {
21             arr[i][j] = 10 - arr[i][j];
22             cout<<arr[i][j]<<" ";
23         }
24         cout<<endl;
25     }
26     return 0;
27 }
28
29
30 }
```
- Output:**

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V04 ( Restore using ( 10 - arr[i][j] ) , without creating ne
w array )$ ls
V04 V04.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V04 ( Restore using ( 10 - arr[i][j] ) , without creating ne
w array )$ g++ V04.cpp -o V04
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V04 ( Restore using ( 10 - arr[i][j] ) , without creating ne
w array )$ ./V04
1 2 3
4 5 6
7 8 9
Reversed Array
9 8 7
6 5 4
3 2 1
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 03 ( Reverse o
f 2-D Array by Subtracting 10 )/Task V04 ( Restore using ( 10 - arr[i][j] ) , without creating ne
w array )$ |
```
- Bottom Status Bar:** C++ ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

Program 7

Task 4 v1:
Source Code:

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

int main()
{
    int arr[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

    cout << "Array is : " << endl;
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
            cout << arr[i][j] << " ";
        cout << endl;
    }

    cout << "\nModified Array is : " << endl;
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
            cout << i+1 << " ";
        cout << endl;
    }
    return 0;
}
```

Output

The screenshot shows a terminal window with the following details:

- Title Bar:** Activities Terminal ▾
- File Path:** ~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 04 (Restore values in Same array after performing certain Operations)/Task V01 (Restore Values, Use row loop [int i, for all j])
- Code (V01.cpp):**

```

1 /**
2 * Restore the same values in the same array, arr[3][3];
3 *
4     1 2 3           1 1 1
5     4 5 6   ===>  2 2 2
6
7     7 8 9           3 3 3
8
9
10    V1. Use row loop [int i, for all j]
11
12 */
13
14 #include <iostream>
15 using namespace std;
16
17 int main()
18 {
19     int arr[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
20
21     cout << "Array is : " << endl;
22     for(int i = 0; i < 3; i++)
23     {
24         for(int j = 0; j < 3; j++)
25             cout << arr[i][j] << " ";
26         cout << endl;
27     }
28
29     cout << "\nModified Array is : " << endl;
30     for(int i = 0; i < 3; i++)
31     {
32         for(int j = 0; j < 3; j++)
33             cout << i+1 << " ";
34         cout << endl;
35     }
36
37     return 0;
38 }
```
- Terminal Output:**

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 04 ( Restore v
values in Same array after performing certian Operations )/Task V01 ( Restore Values, Use row l
oop [int i, for all j] )$ ls
V01 V01.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 04 ( Restore v
values in Same array after performing certian Operations )/Task V01 ( Restore Values, Use row l
oop [int i, for all j] )$ g++ V01.cpp -o V01
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 04 ( Restore v
values in Same array after performing certian Operations )/Task V01 ( Restore Values, Use row l
oop [int i, for all j] )$ ./V01
Array is :
1 2 3
4 5 6
7 8 9
Modified Array is :
1 1 1
2 2 2
3 3 3

```
- Bottom Status Bar:** C++ ▾ Tab Width: 8 ▾ Ln 13, Col 5 ▾ INS

Program 7

Task 4 v2:
Source Code:

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

int main()
{
    int arr[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

    cout << "Array is : " << endl;
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
            cout << arr[i][j] << " ";
        cout << endl;
    }

    cout << "\nModified Array is : " << endl;
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
            cout << arr[i][j]/3 << " ";
        cout << endl;
    }
    return 0;
}
```

Output

The screenshot shows a terminal window with the following details:

- Title Bar:** Activities Terminal ▾ Tue Oct 19 23:23
- File Path:** -/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 04 (Restore v alues in Same array after performing certian Operations)/Task V02 (Restore Values, Use arr[i] [N-1]by3, at each place)\$ ls
- Code (V02.cpp):**

```

1 /**
2 
3 Restore the same values in the same array, arr[3][3];
4 
5      1 2 3          1 1 1
6      4 5 6    ==>  2 2 2
7      7 8 9          3 3 3
8 
9 V2. Use arr[i][N-1]/3, at each place
10 
11 */
12 
13 #include <iostream>
14 using namespace std;
15 
16 int main()
17 {
18     int arr[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
19 
20     cout << "Array is : " << endl;
21     for(int i = 0; i < 3; i++)
22     {
23         for(int j = 0; j < 3; j++)
24             cout << arr[i][j] << " ";
25         cout << endl;
26     }
27 
28     cout << "\nModified Array is : " << endl;
29     for(int i = 0; i < 3; i++)
30     {
31         for(int j = 0; j < 3; j++)
32             cout << arr[i][j]/3 << " ";
33         cout << endl;
34     }
35 
36     return 0;
37 }
38 }
```
- Output:**

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 04 ( Restore v alues in Same array after performing certian Operations )/Task V02 ( Restore Values, Use arr[i] [N-1]by3, at each place )$ ls
V02_V02.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 04 ( Restore v alues in Same array after performing certian Operations )/Task V02 ( Restore Values, Use arr[i] [N-1]by3, at each place )$ g++ V02.cpp -o V02
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 04 ( Restore v alues in Same array after performing certian Operations )/Task V02 ( Restore Values, Use arr[i] [N-1]by3, at each place )$ ./V02
Array is :
1 2 3
4 5 6
7 8 9
Modified Array is :
1 1 1
2 2 2
3 3 3
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 04 ( Restore v alues in Same array after performing certian Operations )/Task V02 ( Restore Values, Use arr[i] [N-1]by3, at each place )$
```
- Bottom Status Bar:** C++ ▾ Tab Width: 8 ▾ Ln 13, Col 5 ▾ INS

Program 7

Task 4 v3:
Source Code:

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

int main()
{
    int arr[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

    cout << "Array is : " << endl;
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
            cout << arr[i][j] << " ";
        cout << endl;
    }

    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
            arr[i][j] = arr[i][j] - (2*i+j);
    }

    cout << "\nModified Array is : " << endl;
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
            cout << arr[i][j] << " ";
        cout << endl;
    }

    return 0;
}
```

Output

The screenshot shows a terminal window with the following details:

- Title Bar:** Activities Terminal ▾
- Time:** Tue Oct 19 23:30
- File Path:** ~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task V03 (Use, arr[i][j]-(2^i+j))
- Code (Content of V03.cpp):**

```

1 /*
2
3 Restore the same values in the same array, arr[3][3]:
4
5      1 2 3          1 1 1
6      4 5 6  ==>    2 2 2
7      7 8 9          3 3 3
10
11 v3. Use, arr[i][j]-(2*i+j)
12 */
13
14 #include <iostream>
15 using namespace std;
16
17 int main()
18 {
19     int arr[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
20
21     cout << "Array is : " << endl;
22     for(int i = 0; i < 3; i++)
23     {
24         for(int j = 0; j < 3; j++)
25             cout << arr[i][j] << " ";
26         cout << endl;
27     }
28
29     for(int i = 0; i < 3; i++)
30     {
31         for(int j = 0; j < 3; j++)
32             arr[i][j] = arr[i][j] - (2*i+j);
33     }
34
35     cout << "\nModified Array is : " << endl;
36     for(int i = 0; i < 3; i++)
37     {
38         for(int j = 0; j < 3; j++)
39             cout << arr[i][j] << " ";
40         cout << endl;
41     }
42
43     return 0;
44 }
45

```
- Output (Terminal Output):**

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 04 ( Restore v
values in Same array after performing certian Operations )/Task V03 ( Use, arr[i][j]-(2^i+j) )$ ls
V03.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 04 ( Restore v
values in Same array after performing certian Operations )/Task V03 ( Use, arr[i][j]-(2^i+j) )$ g+
V03.cpp -o V03
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 04 ( Restore v
values in Same array after performing certian Operations )/Task V03 ( Use, arr[i][j]-(2^i+j) )$ ./V03
Array is :
1 2 3
4 5 6
7 8 9
Modified Array is :
1 1 1
2 2 2
3 3 3

```
- Bottom Status Bar:** C++ ▾ Tab Width: 8 ▾ Ln 44, Col 2 ▾ INS

Program 7

Task 5:
Source Code:

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

int main()
{
    char arr[4][4];

    for(int i = 0; i < 4; i++)
    {
        for(int j = 0; j < 4; j++)
        {
            if(j <= i)
            {
                arr[i][j] = '*';
                cout<<arr[i][j]<<" ";
            }
            else
            {
                arr[i][j] = ' ';
                cout<<arr[i][j]<<" ";
            }
        }
        cout<<endl;
    }

    return 0;
}
```

Output

```

Activities Terminal ▾ Tue Oct 19 23:31
Open ... Task05.cpp
~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 05 (Left Half Pyramid)
Save ...
/*pattern
2
3 *
4 **
5 ***
6 ****
7 *****
8 *****
9

10 #include<iostream>
11 using namespace std;
12
13 int main()
14 {
15     char arr[4][4];
16
17     for(int i = 0; i< 4; i++)
18     {
19         for(int j = 0; j < 4; j++)
20         {
21             if(j <= i)
22             {
23                 arr[i][j] = '*';
24                 cout<<arr[i][j]<< " ";
25             }
26             else
27             {
28                 arr[i][j] = ' ';
29                 cout<<arr[i][j]<< " ";
30             }
31         }
32         cout<<endl;
33     }
34
35
36
37 return 0;
38 }
39

```

The terminal output shows the execution of the program Task05.cpp. The output is a left half pyramid pattern of asterisks:

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 05 (Left Half Pyramid)$ ls
Task05.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 05 (Left Half Pyramid)$ g++ Task05.cpp -o Task05
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 ( Array and 2-D Array )/Task 05 (Left Half Pyramid)$ ./Task05
*
**
***
****
*****

```

Program 7

Task 6:
Source Code:

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

int main()
{
    char arr[4][4];

    for(int i = 0; i < 4; i++)
    {
        for(int j = 0; j < 4; j++)
        {
            if(j >= i)
            {
                arr[i][j] = '*';
                cout<<arr[i][j]<<" ";
            }
            else
            {
                arr[i][j] = ' ';
                cout<<arr[i][j]<<" ";
            }
        }
        cout<<endl;
    }

    return 0;
}
```

Output

```

Activities Terminal ▾ Tue Oct 19 23:31
Open ... Task06.cpp Save ...
~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 06 (Inverted Right Half Pyramid)

2 *   *
3 *   *   *
4 *   *   *   *
5 *   *   *   *
6 *   *   *   *
7 */
8
9 #include<iostream>
10 using namespace std;
11
12 int main()
13 {
14     char arr[4][4];
15
16     for(int i = 0; i < 4; i++)
17     {
18         for(int j = 0; j < 4; j++)
19         {
20             if(j >= i)
21             {
22                 arr[i][j] = '*';
23                 cout<<arr[i][j]<<" ";
24             }
25             else
26             {
27                 arr[i][j] = ' ';
28                 cout<<arr[i][j]<<" ";
29             }
30         }
31         cout<<endl;
32     }
33 }
34
35
36 return 0;
37 }
38

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 06 (Inverted Right Half Pyramid)\$ ls
Task06 Task06.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 06 (Inverted Right Half Pyramid)\$ g++ Task06.cpp -o Task06
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 06 (Inverted Right Half Pyramid)\$./Task06
* *
* * *
* * * *
* * * *
* * * *
* * * *

Program 7

Task 7:

Source Code:

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

int main()
{
    char arr[4][4];

    for(int i = 0; i< 4; i++)
    {
        for(int j = 0; j < 4; j++)
        {
            if((i+j) >= 3)
            {
                arr[i][j] = '*';
                cout<<arr[i][j]<<" ";
            }
            else
            {
                arr[i][j] = ' ';
                cout<<arr[i][j]<<" ";
            }
        }
        cout<<endl;
    }

    return 0;
}
```

Output

```
1 /*Pattern
2 *
3 * *
4 * **
5 * ***
6 * ****
7 */
8 */
9 #include<iostream>
10 using namespace std;
11 int main()
12 {
13     char arr[4][4];
14
15     for(int i = 0; i < 4; i++)
16     {
17         for(int j = 0; j < 4; j++)
18         {
19             if((i+j) >= 3)
20             {
21                 arr[i][j] = '*';
22                 cout<<arr[i][j]<<" ";
23             }
24             else
25             {
26                 arr[i][j] = ' ';
27                 cout<<arr[i][j]<<" ";
28             }
29         }
30     }
31     cout<<endl;
32 }
33
34
35
36 return 0;
37
38
```

Bracket match found on line: 13

Tue Oct 19 23:33

Save ...

*Task07.cpp
~/Documents/OOPS in C++/Practical07 (Array and 2-D Array)/Task 07 (Right Half Pyramid)

knockcat@VICKY: ~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 07 (Right Half Pyramid)\$ ls
Task07 Task07.cpp
knockcat@VICKY: ~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 07 (Right Half Pyramid)\$ g++ Task07.cpp -o Task07
knockcat@VICKY: ~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 07 (Right Half Pyramid)\$./Task07
*
* *
* * *
* * * *

Program 7

Task 8:
Source Code:

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

int main()
{
    char arr[4][4];

    for(int i = 0; i < 4; i++)
    {
        for(int j = 0; j < 4; j++)
        {
            if((i+j) <= 3)
            {
                arr[i][j] = '*';
                cout<<arr[i][j]<<" ";
            }
            else
            {
                arr[i][j] = ' ';
                cout<<arr[i][j]<<" ";
            }
        }
        cout<<endl;
    }

    return 0;
}
```

Output

```

Activities Terminal ▾ Tue Oct 19 23:35
Open ... Task08.cpp
~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 08 (Inverted Right Half)
Save ...
/*Pattern
2   * *
3   * *
4   * *
5   *
6   *
7   */
#include<iostream>
using namespace std;
int main()
{
    char arr[4][4];
    for(int i = 0; i < 4; i++)
    {
        for(int j = 0; j < 4; j++)
        {
            if((i+j) <= 3)
            {
                arr[i][j] = '*';
                cout<<arr[i][j]<<" ";
            }
            else
            {
                arr[i][j] = ' ';
                cout<<arr[i][j]<<" ";
            }
        }
        cout<<endl;
    }
    return 0;
}

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 08 (Inverted Right Half)\$ ls
Task08.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 08 (Inverted Right Half\$ g++ Task08.cpp -o Task08
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 08 (Inverted Right Half\$./Task08
* * *
* *
*
knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 08 (Inverted Right Half\$ |

Program 7

Task 9:
Source Code:

```
#include <iostream>
using namespace std;
int main()
{
    int n, m;
    cout<<"\nInput the number of rows: ";
    cin>>m;
    cout<<"\nInput the number of columns : ";
    cin>>n;
    int arr[m][n];
    int i, j;
    cout<<"\nInput the matrix \n";
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            cin>>arr[i][j];
        }
    }
    cout<<"\nThe Matrix is \n";
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            cout<<arr[i][j]<<" ";
        }
        cout<<"\n";
    }
    cout<<"\nSpiral Matrix      : ";
    int k = 0, l = 0;
    while (k < m && l < n)
    {

```

```
for (i = l; i < n; i++)
{
    cout<< arr[k][i]<< " ";
}
k++;
for (i = k; i < m; i++)
{
    cout<< arr[i][n - 1]<< " ";
}
n-- ;
if (k < m)
{
    for (i = n - 1; i >= 0; i--)
    {
        cout<< arr[m - 1][i] << " ";
    }
    m-- ;
}
if (l < n)
{
    for (i = m-1; i >= k; i--)
    {
        cout<< arr[i][l] << " ";
    }
    l++;
}
return 0;
}
```

Output

Activities Terminal Tue Oct 19 23:36 Task09.cpp -/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 09 (Spiral Matrix) Save ...

```
// Spiral Matrix
2
3 /*
4
5 For a 3 by 3 matrix suppose matrix will be
6      1 2 3
7      4 5 6
8      7 8 9
9
10 Then the Spiral Matrix will be      1 2 3 6 9 8 7 4 5
11
12 */
13
14 #include <iostream>
15 using namespace std;
16 int main()
17 {
18     int n, m;
19     cout<<"\nInput the number of rows: ";
20     cin>>n;
21     cout<<"\nInput the number of columns : ";
22     cin>>m;
23     int arr[n][m];
24     int i, j;
25     cout<<"\nInput the matrix \n";
26     for (i = 0; i < m; i++)
27     {
28         for (j = 0; j < n; j++)
29         {
30             cin>>arr[i][j];
31         }
32     }
33     cout<<"\nThe Matrix is \n";
34     for (i = 0; i < m; i++)
35     {
36         for (j = 0; j < n; j++)
37         {
38             cout<<arr[i][j]<< " ";
39         }
40         cout<<"\n";
41     }
42     cout<<"\nSpiral Matrix      : ";
43     int k = 0, l = 0;
44     while (k < m && l < n)
45     {
46         // Top row
47         for (int i = l; i < n; i++)
48             cout<<arr[k][i]<< " ";
49         k++;
50         // Right column
51         for (int i = k; i < m; i++)
52             cout<<arr[i][n-1]<< " ";
53         n--;
54         // Bottom row
55         if (k < m)
56             for (int i = n-1; i >= l; i--)
57                 cout<<arr[m-1][i]<< " ";
58         m--;
59         // Left column
60         if (l < n)
61             for (int i = m-1; i >= k; i--)
62                 cout<<arr[i][l]<< " ";
63         l++;
64     }
65 }
```

knockcat@VICKY: ~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 09 (Spiral Matrix)\$ ls
Task09 Task09.cpp
knockcat@VICKY: ~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 09 (Spiral Matrix)\$ g++ Task09.cpp -o Task09
knockcat@VICKY: ~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 09 (Spiral Matrix)\$./Task09
Input the number of rows: 4
Input the number of columns : 4
Input the matrix
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
The Matrix is
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
Spiral Matrix : 1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10 9 knockcat@VICKY:~/Documents/OOPS in C++/Practical 07 (Array and 2-D Array)/Task 09 (Spiral Matrix)\$ |

Program 8

Task 1:

Source Code:

```
#include<iostream>
using namespace std;
int main()
{
    int *p = {10,20,30};

    cout << *p;
    p++;
    cout << (*p);

    return 0;
}
```

Output

Program 8

Task 2:

Source Code:

V01 //Issue

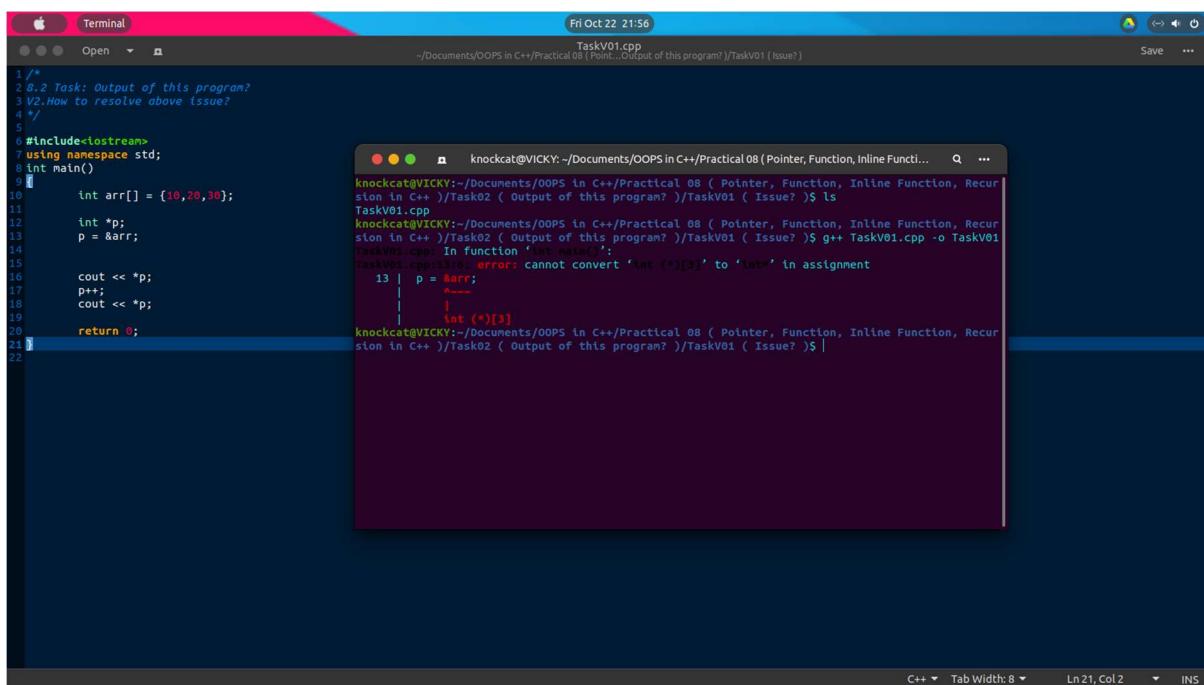
```
#include<iostream>
using namespace std;
int main()
{
    int arr[] = {10,20,30};

    int *p;
    p = arr;      //int *p = &arr;

    cout << *p << endl;
    p++;
    cout << *p << endl;

    return 0;
}
```

Output



The screenshot shows a Mac OS X terminal window titled "Terminal" with the status bar indicating "Fri Oct 22 21:56". The current directory is "/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task02 (Output of this program?)/TaskV01 (Issue?)". The terminal displays the following code:

```

1 /*
2 0.2 Task: Output of this program?
3 V2. How to resolve above issue?
4 */
5
6 #include<iostream>
7 using namespace std;
8 int main()
9 {
10     int arr[] = {10,20,30};
11     int *p;
12     p = &arr;
13     cout << *p;
14     cout << *p;
15     cout << *p;
16     cout << *p;
17     p++;
18     cout << *p;
19
20     return 0;
21 }
22

```

After running the command \$ g++ TaskV01.cpp -o TaskV01, the terminal shows the following error output:

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 ( Pointer, Function, Inline Function, Recursion in C++ )/Task02 ( Output of this program? )/TaskV01 ( Issue? )$ ls
TaskV01.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 ( Pointer, Function, Inline Function, Recursion in C++ )/Task02 ( Output of this program? )/TaskV01 ( Issue? )$ g++ TaskV01.cpp -o TaskV01
TaskV01.cpp: In function 'int main()':
TaskV01.cpp:13:6: error: cannot convert 'int (*)[3]' to 'int' in assignment
    13 |     p = &arr;
      |     ^
      |
      |     int (*)[3]
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 ( Pointer, Function, Inline Function, Recursion in C++ )/Task02 ( Output of this program? )/TaskV01 ( Issue? )$ 

```

The terminal also shows the status bar with "C++" and "Tab Width: 8".

Program 8

Task 2:

Source Code:

V02 //Resolving Issue

```
#include<iostream>
using namespace std;
int main()
{
    int arr[] = {10,20,30};

    int *p;
    p = &arr;

    cout << *p;
    p++;
    cout << *p;

    return 0;
}
```

Output

```

1 /*
2 8.2 Task: Output of this program?
3
4 V1. Issue?
5 */
6
7 #include<iostream>
8 using namespace std;
9 int main()
10 {
11     int arr[] = {10,20,30};
12
13     int *p;
14     p = arr;      //int *p = &arr;
15
16     cout << *p << endl;
17     p++;
18     cout << *p << endl;
19
20     return 0;
21 }

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task02 (Output of this program?)/TaskV02 (How to resolve above issue?)\$ ls
TaskV02.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task02 (Output of this program?)/TaskV02 (How to resolve above issue?)\$ g++ T
skV02.cpp -o TaskV02
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task02 (Output of this program?)/TaskV02 (How to resolve above issue?)\$./Task
V02
10
20
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task02 (Output of this program?)/TaskV02 (How to resolve above issue?)\$ |

Program 8

Task 3:

Source Code:

V01

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

```
int main()
{
    int a = 10;

    int *p;

    int **q;

    p = &a;

    q = &p;

    cout << *p << endl ;

    cout << **q << endl ;

    return 0;
}
```

Output

```

1/*
2 8.3 Task: Output of this program?
3
4 V1. Output?
5 */
6
7#include <iostream>
8using namespace std;
9
10int main()
11{
12    int a = 10;
13    int *p;
14    int **q;
15
16    p = &a;
17
18    q = &p;
19
20    cout << *p << endl ;
21
22    cout << **q << endl ;
23
24
25    return 0;
26}
27

```

Fri Oct 22 22:03

TaskV01.cpp

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task03 (Output of this program?)/TaskV01 (Output?)\$ ls

TaskV01 TaskV01.cpp

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task03 (Output of this program?)/TaskV01 (Output?)\$ g++ TaskV01.cpp -o TaskV01

10

10

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task03 (Output of this program?)/TaskV01 (Output?)\$./TaskV01

Program 8

Task 3:

Source Code:

V02

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

int main()
{
    int a = 10;

    int *p = &a;
    int **q = &p;

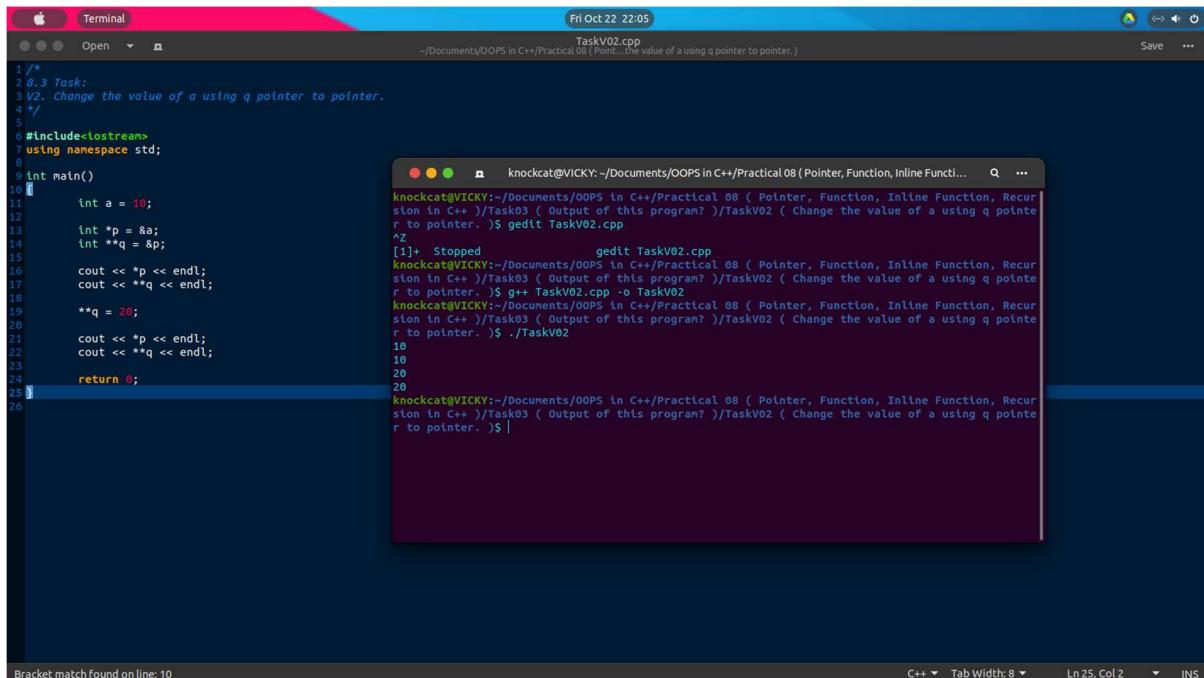
    cout << *p << endl;
    cout << **q << endl;

    **q = 20;

    cout << *p << endl;
    cout << **q << endl;

    return 0;
}
```

Output



```

1 /*
2 0.3 Task:
3 V2. Change the value of a using q pointer to pointer.
4 */
5
6 #include<iostream>
7 using namespace std;
8
9 int main()
10 {
11     int a = 10;
12
13     int *p = &a;
14     int **q = &p;
15
16     cout << *p << endl;
17     cout << **q << endl;
18
19     **q = 20;
20
21     cout << *p << endl;
22     cout << **q << endl;
23
24     return 0;
25 }

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task03 (Output of this program?)/TaskV02 (Change the value of a using q pointer to pointer.)\$ gedit TaskV02.cpp

[1]+ Stopped gedit TaskV02.cpp

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task03 (Output of this program?)/TaskV02 (Change the value of a using q pointer to pointer.)\$ g++ TaskV02.cpp -o TaskV02

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task03 (Output of this program?)/TaskV02 (Change the value of a using q pointer to pointer.)\$./TaskV02

10
20

Program 8

Task 4:
Source Code:

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

int fact(int a)
{
    if(a < 0)
        return 0;
    if(a == 0 || a == 1)
        return 1;

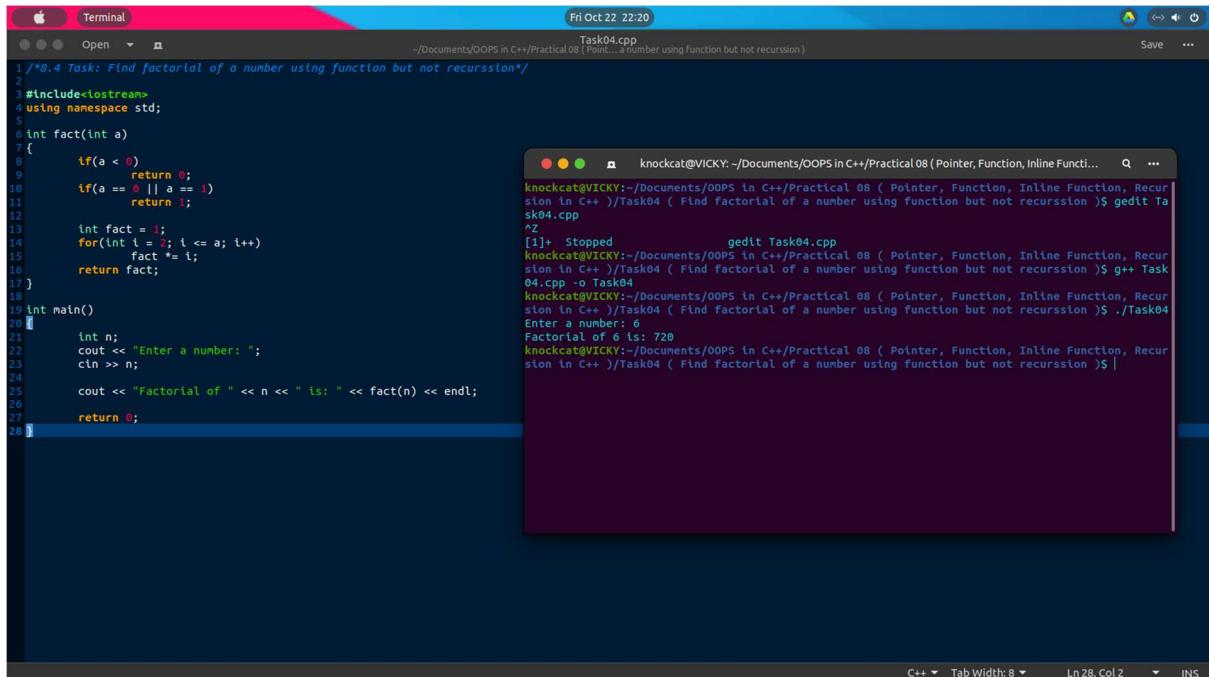
    int fact = 1;
    for(int i = 2; i <= a; i++)
        fact *= i;
    return fact;
}

int main()
{
    int n;
    cout << "Enter a number: ";
    cin >> n;

    cout << "Factorial of " << n << " is: " << fact(n) << endl;

    return 0;
}
```

Output



```

1 /*0.4 Task: Find factorial of a number using function but not recursion*/
2
3 #include<iostream>
4 using namespace std;
5
6 int fact(int a)
7 {
8     if(a < 0)
9         return 0;
10    if(a == 0 || a == 1)
11        return 1;
12
13    int fact = 1;
14    for(int i = 2; i <= a; i++)
15        fact *= i;
16    return fact;
17 }
18
19 int main()
20 {
21     int n;
22     cout << "Enter a number: ";
23     cin >> n;
24
25     cout << "Factorial of " << n << " is: " << fact(n) << endl;
26
27     return 0;
28 }

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task04 (Find factorial of a number using function but not recursion)\$ gedit Task04.cpp
[1]+ Stopped gedit Task04.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task04 (Find factorial of a number using function but not recursion)\$ g++ Task04.cpp -o Task04
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task04 (Find factorial of a number using function but not recursion)\$./Task04
Enter a number: 6
Factorial of 6 is: 720
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task04 (Find factorial of a number using function but not recursion)\$ |

Program 8

Task 5:
Source Code:

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

int fact(int a)
{
    if(a == 0 || a == 1)
        return 1;

    if(a > 1)
        return a*fact(a-1);
    return 1;
}

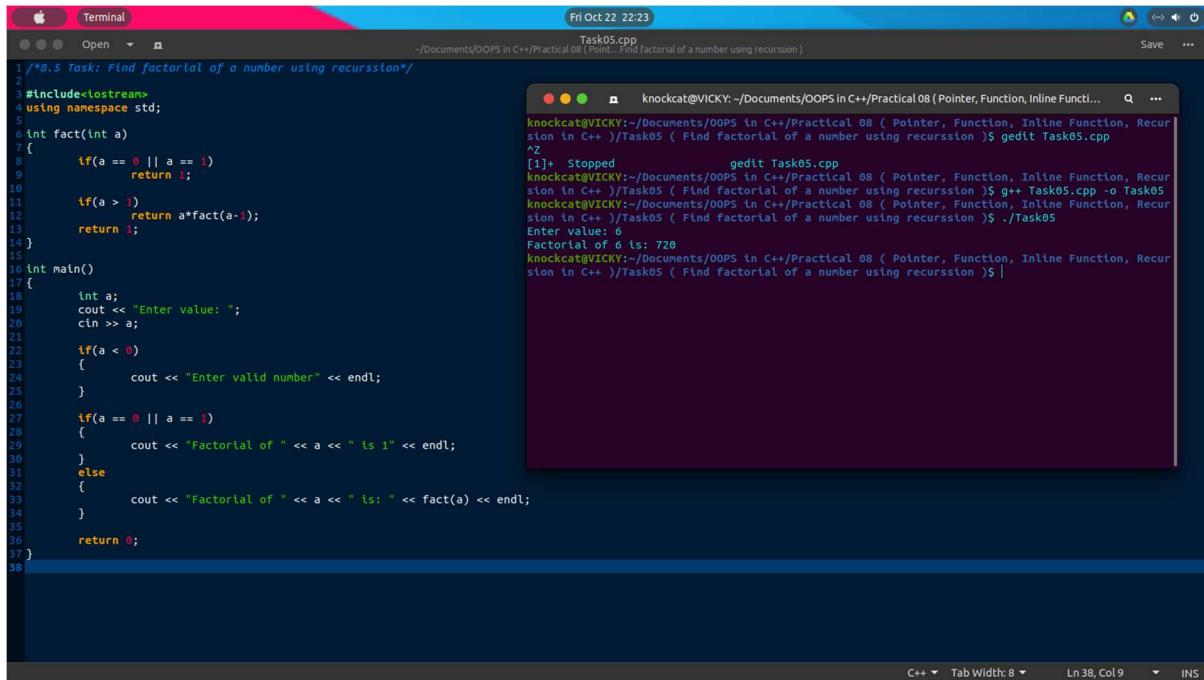
int main()
{
    int a;
    cout << "Enter value: ";
    cin >> a;

    if(a < 0)
    {
        cout << "Enter valid number" << endl;
    }

    if(a == 0 || a == 1)
    {
        cout << "Factorial of " << a << " is 1" << endl;
    }
    else
    {
        cout << "Factorial of " << a << " is: " << fact(a) << endl;
    }

    return 0;
}
```

Output



The screenshot shows a terminal window with two panes. The left pane displays the source code for a C++ program named Task05.cpp. The right pane shows the terminal session where the program is compiled and run.

```

1 /*8.5 Task: Find factorial of a number using recursion*/
2
3 #include<iostream>
4 using namespace std;
5
6 int fact(int a)
7 {
8     if(a == 0 || a == 1)
9         return 1;
10    if(a > 1)
11        return a*fact(a-1);
12    return 1;
13 }
14 }
15
16 int main()
17 {
18     int a;
19     cout << "Enter value: ";
20     cin >> a;
21
22     if(a < 0)
23     {
24         cout << "Enter valid number" << endl;
25     }
26
27     if(a == 0 || a == 1)
28     {
29         cout << "Factorial of " << a << " is 1" << endl;
30     }
31     else
32     {
33         cout << "Factorial of " << a << " is: " << fact(a) << endl;
34     }
35
36     return 0;
37 }
38

```

In the terminal session, the user enters the value 6. The program calculates the factorial of 6 and prints the result 720.

```

knockcat@VICKY:~/Documents/OOPS in C+/Practical 08 ( Pointer, Function, Inline Function, Recursion in C++ )/Task05 ( Find factorial of a number using recursion )$ gedit Task05.cpp
knockcat@VICKY:~/Documents/OOPS in C+/Practical 08 ( Pointer, Function, Inline Function, Recursion in C++ )/Task05 ( Find factorial of a number using recursion )$ g++ Task05.cpp -o Task05
knockcat@VICKY:~/Documents/OOPS in C+/Practical 08 ( Pointer, Function, Inline Function, Recursion in C++ )/Task05 ( Find factorial of a number using recursion )$ ./Task05
Enter value: 6
Factorial of 6 is: 720
knockcat@VICKY:~/Documents/OOPS in C+/Practical 08 ( Pointer, Function, Inline Function, Recursion in C++ )/Task05 ( Find factorial of a number using recursion )$ 

```

Program 8

Task 6:

Source Code:

```
#include<iostream>
#include<cmath>
using namespace std;
void series(int n)
{
    if(n)
    {
        series(n-1);

    }
    else
    {
        return ;
    }

    cout<<pow(n,n)+n<<",";
}

int main()
{
    int NUM;
    cout << "Enter value for which you want to print series: ";
    cin >> NUM;
    series(NUM);
    return 0;
}
```

Output

```

1 /*
2 0.6 Task: Series Problem using recursion for n series
3
4 2, (2^2 + 2), (3^3 + 3), (4^4 + 4), (5^5 + 5), .....
5
6 Hint:
7
8 n * ((n-1)^n(n-1) + (n-1))
9 */
10
11 #include<iostream>
12 #include<cmath>
13 using namespace std;
14 void series(int n)
15 {
16     if(n)
17     {
18         series(n-1);
19
20     }
21     else
22     {
23         return ;
24     }
25
26     cout<<pow(n,n)+n<<" ";
27
28 }
29 int main()
30 {
31     int NUM;
32     cout << "Enter value for which you want to print series: ";
33     cin >> NUM;
34     series(NUM);
35     return 0;
36 }

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task06 (Series Problem using recursion for n series)\$ ls
Task06.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task06 (Series Problem using recursion for n series)\$ g++ Task06.cpp -o Task06
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task06 (Series Problem using recursion for n series)\$./Task06
Enter value for which you want to print series: 7
2,6,30,260,3130,46662,823550,knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task06 (Series Problem using recursion for n series)\$

Program 8

Task 7 v1:

Source Code:

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

void swap(int a, int b)
{
    int temp = b;
    b = a;
    a = temp;
}

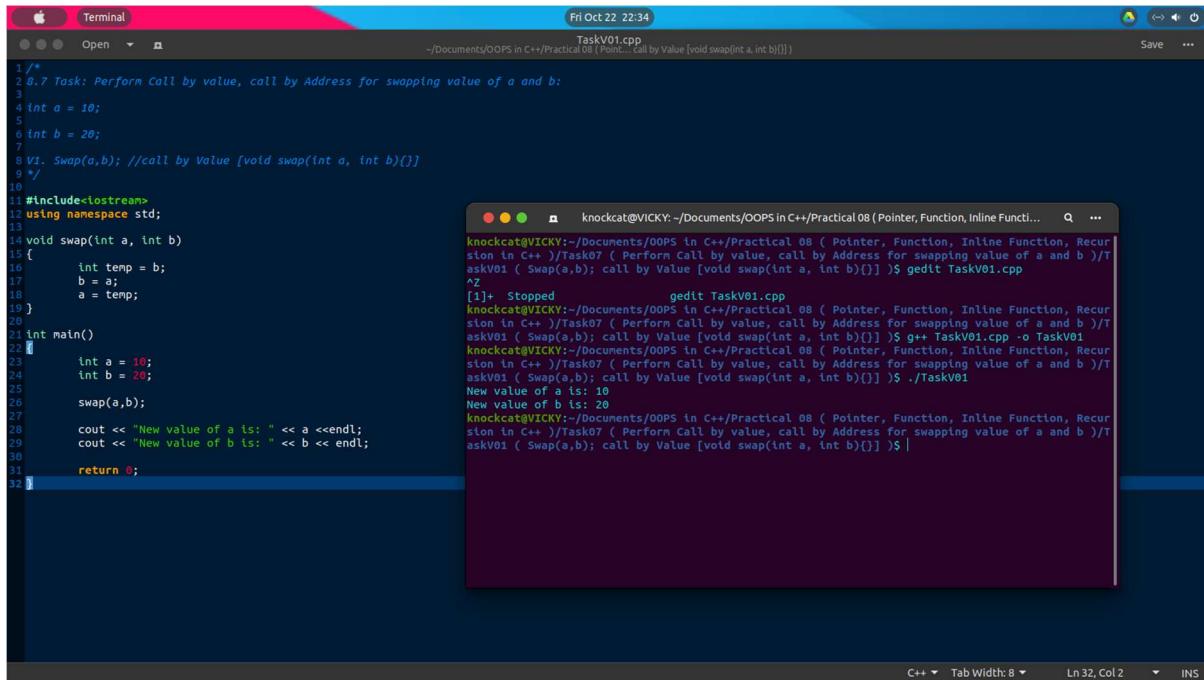
int main()
{
    int a = 10;
    int b = 20;

    swap(a,b);

    cout << "New value of a is: " << a << endl;
    cout << "New value of b is: " << b << endl;

    return 0;
}
```

Output



```

1/*
2 8.7 Task: Perform Call by value, call by Address for swapping value of a and b:
3
4 int a = 10;
5
6 int b = 20;
7
8 Viz. Swap(a,b); //call by Value [void swap(int a, int b){}]
9 */
10
11 #include<iostream>
12 using namespace std;
13
14 void swap(int a, int b)
15 {
16     int temp = b;
17     b = a;
18     a = temp;
19 }
20
21 int main()
22 {
23     int a = 10;
24     int b = 20;
25
26     swap(a,b);
27
28     cout << "New value of a is: " << a << endl;
29     cout << "New value of b is: " << b << endl;
30
31     return 0;
32 }

```

Fri Oct 22 22:34
TaskV01.cpp
~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task07 (Perform Call by value, call by Address for swapping value of a and b)/TaskV01 (Swap(a,b); call by Value [void swap(int a, int b){}]) \$ gedit TaskV01.cpp
^Z
[1]+ Stopped gedit TaskV01.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task07 (Perform Call by value, call by Address for swapping value of a and b)/TaskV01 (Swap(a,b); call by Value [void swap(int a, int b){}]) \$ g++ TaskV01.cpp -o TaskV01
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task07 (Perform Call by value, call by Address for swapping value of a and b)/TaskV01 (Swap(a,b); call by Value [void swap(int a, int b){}]) \$./TaskV01
New value of a is: 10
New value of b is: 20
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)/Task07 (Perform Call by value, call by Address for swapping value of a and b)/TaskV01 (Swap(a,b); call by Value [void swap(int a, int b){}]) \$

Program 8

Task 7 v2:

Source Code:

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

void swap(int &a, int &b)
{
    int temp = b;
    b = a;
    a = temp;
}

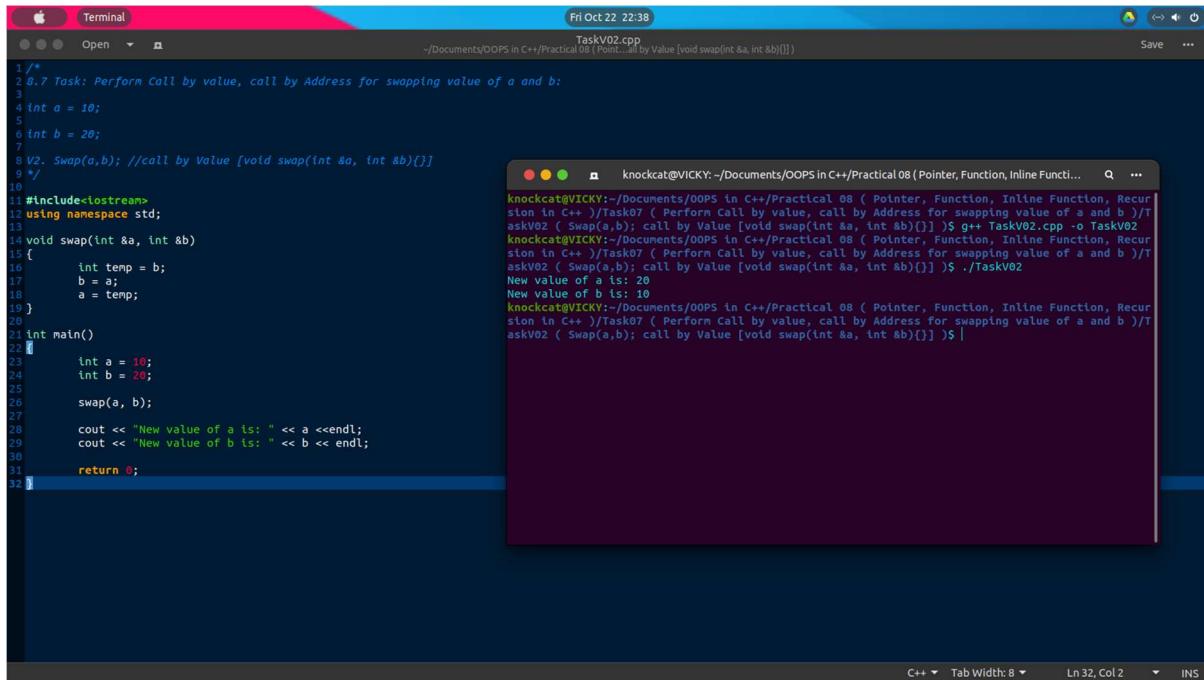
int main()
{
    int a = 10;
    int b = 20;

    swap(a, b);

    cout << "New value of a is: " << a << endl;
    cout << "New value of b is: " << b << endl;

    return 0;
}
```

Output



The screenshot shows a Mac OS X terminal window with two panes. The left pane displays the source code for TaskV02.cpp, which includes a swap function and a main function demonstrating its use. The right pane shows the terminal output where the program is compiled with g++, run, and its output is displayed, showing the values of variables a and b being swapped.

```

1 /*
2 8.7 Task: Perform Call by value, call by Address for swapping value of a and b:
3
4 int a = 10;
5
6 int b = 20;
7
8 V2. Swap(a,b); //call by Value [void swap(int &a, int &b){}]
9 */
10
11 #include<iostream>
12 using namespace std;
13
14 void swap(int &a, int &b)
15 {
16     int temp = b;
17     b = a;
18     a = temp;
19 }
20
21 int main()
22 {
23     int a = 10;
24     int b = 20;
25
26     swap(a, b);
27
28     cout << "New value of a is: " << a << endl;
29     cout << "New value of b is: " << b << endl;
30
31     return 0;
32 }

```

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)$ g++ TaskV02.cpp -o TaskV02
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 (Pointer, Function, Inline Function, Recursion in C++)$ ./TaskV02
New value of a is: 20
New value of b is: 10

```

Program 8

Task 7 v3:

Source Code:

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

void swap(int *a,int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

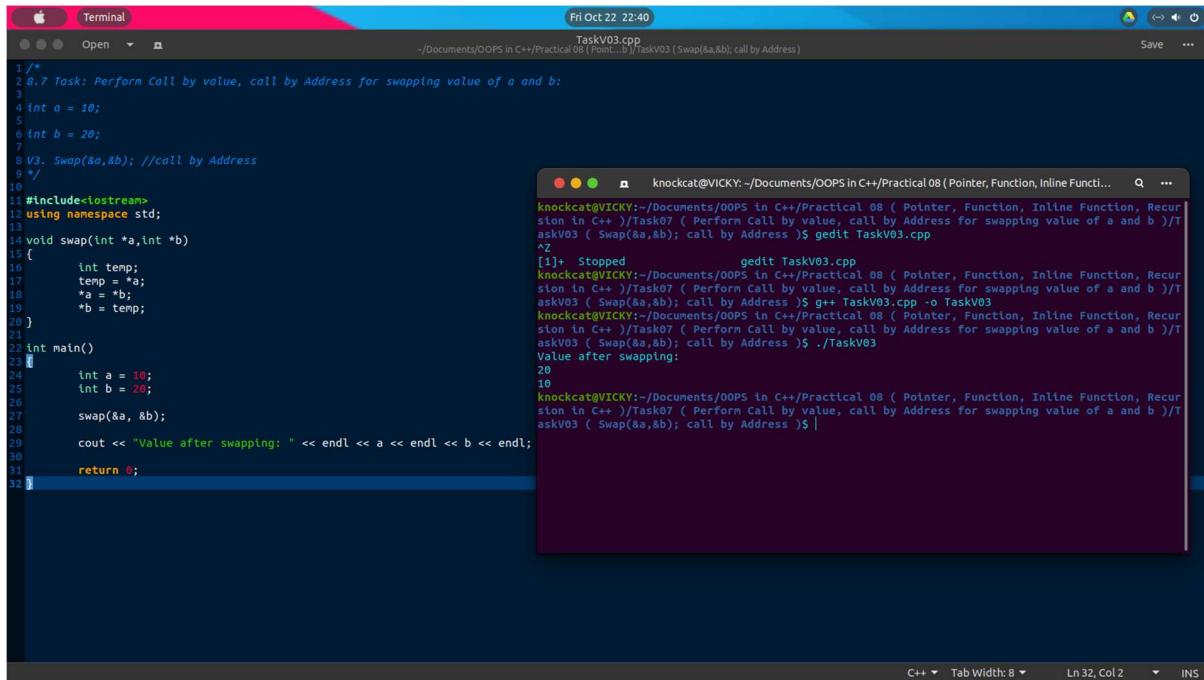
int main()
{
    int a = 10;
    int b = 20;

    swap(&a, &b);

    cout << "Value after swapping: " << endl << a << endl << b <<
endl;

    return 0;
}
```

Output



```

1/*
2 8.7 Task: Perform Call by value, call by Address for swapping value of a and b:
3
4 int a = 10;
5
6 int b = 20;
7
8 V3. Swap(&a,&b); //call by Address
9 */
10
11 #include<iostream>
12 using namespace std;
13
14 void swap(int *a,int *b)
15 {
16     int temp;
17     temp = *a;
18     *a = *b;
19     *b = temp;
20 }
21
22 int main()
23 {
24     int a = 10;
25     int b = 20;
26
27     swap(&a, &b);
28
29     cout << "Value after swapping: " << endl << a << endl << b << endl;
30
31     return 0;
32 }
```

The terminal output shows the program being run and the swap operation. The swap function is called by address, so the values of a and b are swapped.

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 ( Pointer, Function, Inline Function, Recursion in C++ )/Task07 ( Perform Call by value, call by Address for swapping value of a and b )/TaskV03 ( Swap(&a,&b); call by Address )$ gedit TaskV03.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 ( Pointer, Function, Inline Function, Recursion in C++ )/Task07 ( Perform Call by value, call by Address for swapping value of a and b )/TaskV03 ( Swap(&a,&b); call by Address )$ g++ TaskV03.cpp -o TaskV03
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 ( Pointer, Function, Inline Function, Recursion in C++ )/Task07 ( Perform Call by value, call by Address for swapping value of a and b )/TaskV03 ( Swap(&a,&b); call by Address )$ ./TaskV03
Value after swapping:
20
10
knockcat@VICKY:~/Documents/OOPS in C++/Practical 08 ( Pointer, Function, Inline Function, Recursion in C++ )/Task07 ( Perform Call by value, call by Address for swapping value of a and b )/TaskV03 ( Swap(&a,&b); call by Address )$ |
```

Program 9

Task 1 a:

Source Code:

```
#include<iostream>
#include<string>
using namespace std;
int c=1;

class student
{
    private:
        int roll;
        long ph_no;
        string address;
        string name;
    public:
        void set_roll(int roll)
        {
            this->roll=roll;
        }
        void set_ph(long num)
        {
            this->ph_no=num;
        }
        void set_address(string add)
        {
            this->address=add;
        }
        void set_name(string name)
        {

```

```

        this->name=name;
    }

void showdata()
{
    cout<<"NAME :"<<this->name<<endl;
    cout<<"roll no. of student "<<this->roll<<endl;
    cout<<"phone no. of student "<<this->ph_no<<endl;
    cout<<"address of student "<<this->address<<endl;
    c++;
}
};

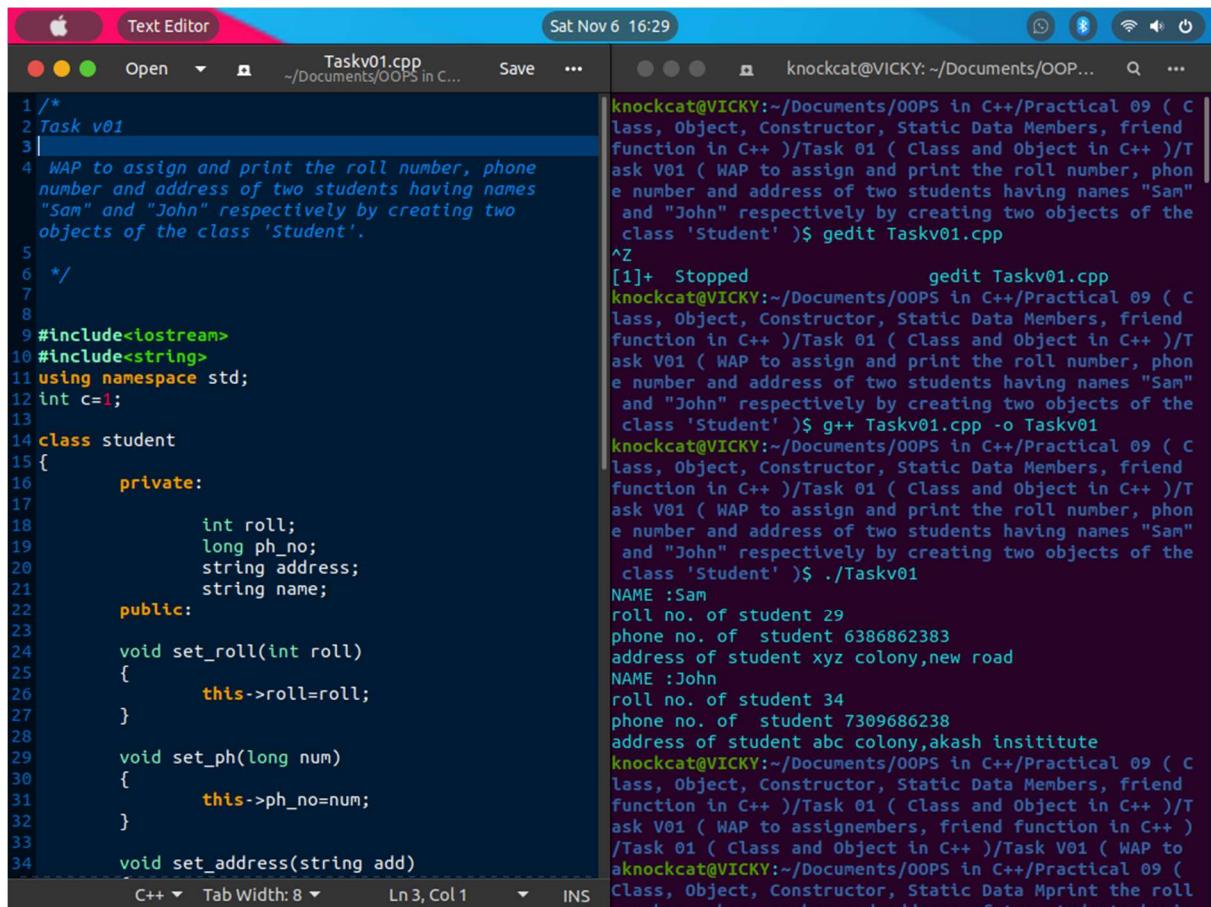
int main()
{
    student obj1,obj2;
    obj1.set_name("Sam");
    obj1.set_roll(29);
    obj1.set_ph(6386862383);
    obj1.set_address("xyz colony,new road");
    obj2.set_name("John");
    obj2.set_roll(34);
    obj2.set_ph(7309686238);
    obj2.set_address("abc colony,akash insititute");

    obj1.showdata();
    obj2.showdata();

    return 0;
}

```

Output



The screenshot shows a Mac OS X desktop environment with a Text Editor window open. The window title is "Text Editor" and the file name is "Taskv01.cpp". The code in the editor is as follows:

```

1 /*
2 Task v01
3
4 WAP to assign and print the roll number, phone
5 number and address of two students having names
6 "Sam" and "John" respectively by creating two
7 objects of the class 'Student'.
8
9 #include<iostream>
10 #include<string>
11 using namespace std;
12 int c=1;
13
14 class student
15 {
16     private:
17         int roll;
18         long ph_no;
19         string address;
20         string name;
21
22     public:
23         void set_roll(int roll)
24         {
25             this->roll=roll;
26         }
27         void set_ph(long num)
28         {
29             this->ph_no=num;
30         }
31         void set_address(string add)
32     };
33
34

```

The terminal window to the right shows the command-line session and the output of the program. The user has typed commands to edit the file, compile it, and run it. The output shows the creation of two student objects with names "Sam" and "John", and their respective roll numbers and addresses.

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 ( Class, Object, Constructor, Static Data Members, friend function in C++ )/Task 01 ( Class and Object in C++ )/Task V01 ( WAP to assign and print the roll number, phon e number and address of two students having names "Sam" and "John" respectively by creating two objects of the class 'Student' )$ gedit Taskv01.cpp
^Z
[1]+  Stopped                  gedit Taskv01.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 ( Class, Object, Constructor, Static Data Members, friend function in C++ )/Task 01 ( Class and Object in C++ )/Task V01 ( WAP to assign and print the roll number, phon e number and address of two students having names "Sam" and "John" respectively by creating two objects of the class 'Student' )$ g++ Taskv01.cpp -o Taskv01
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 ( Class, Object, Constructor, Static Data Members, friend function in C++ )/Task 01 ( Class and Object in C++ )/Task V01 ( WAP to assign and print the roll number, phon e number and address of two students having names "Sam" and "John" respectively by creating two objects of the class 'Student' )$ ./Taskv01
NAME :Sam
roll no. of student 29
phone no. of student 6386862383
address of student xyz colony,new road
NAME :John
roll no. of student 34
phone no. of student 7309686238
address of student abc colony,akash insititute
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 ( Class, Object, Constructor, Static Data Members, friend function in C++ )/Task 01 ( Class and Object in C++ )/Task V01 ( WAP to assignmembers, friend function in C++ )/Task 01 ( Class and Object in C++ )/Task V01 ( WAP to
aknockcat@VICKY:~/Documents/OOPS in C++/Practical 09 ( Class, Object, Constructor, Static Data Mprint the roll
number, phone number and address of two students havi

```

Program 9

Task 1 b:
Source Code:

```
#include<iostream>
#include<string>

using namespace std;
class student
{ //friend function array
    private:
        string name;
        int marks;
        char sec;
    public:
        void set_values(string name,int m,char sec)
        {
            this->name=name;
            this->marks=m;
            this->sec=sec;
        }
        friend int get_marks(student obj);
};

int get_marks(student obj)
{
    return obj.marks;
}

int main()
{
    int n;
    cout<<"enter no. of students"<<endl;
    cin>>n;
    student arr[n];
```

```

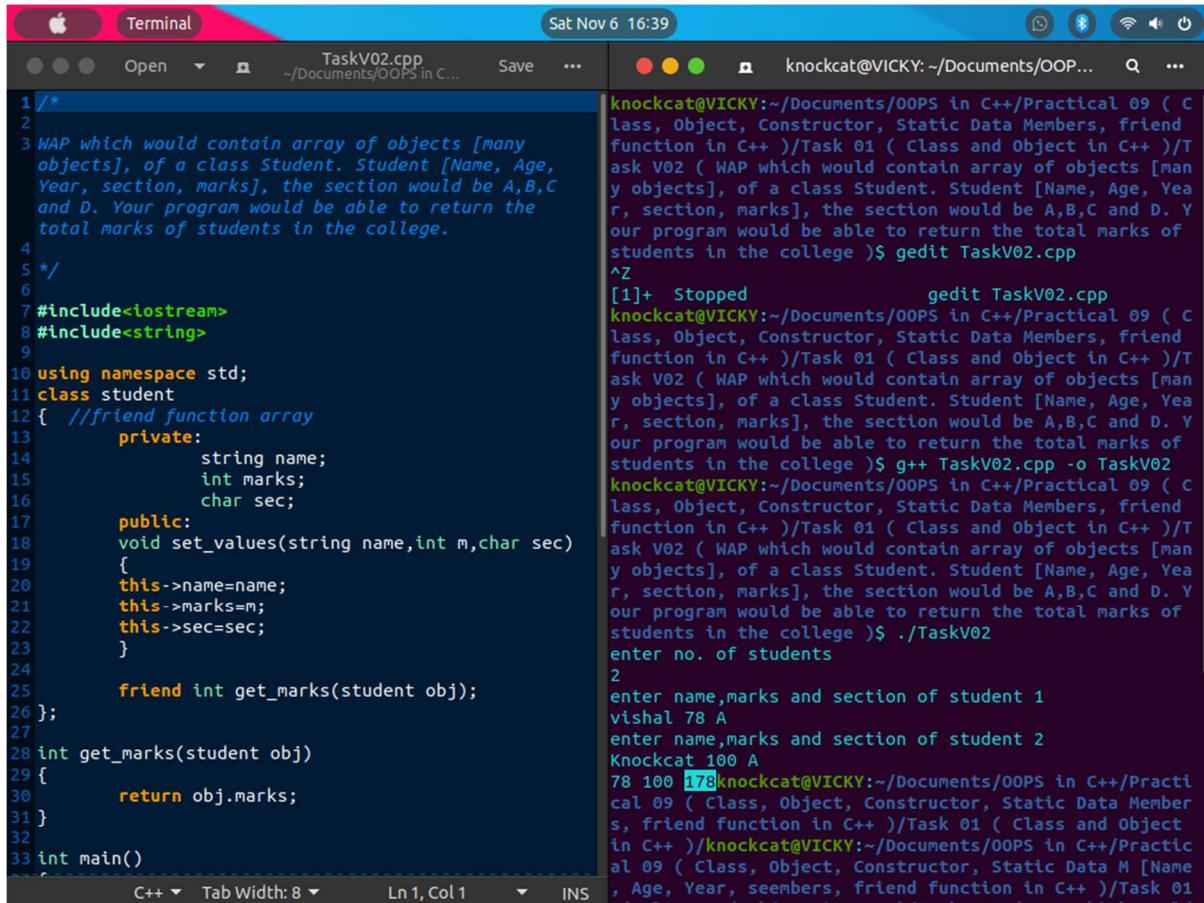
for(int i=0;i<n;i++)
{
    cout<<"enter name,marks and section of student
"<<i+1<<endl;
    int temp;
    string name;
    char sec;
    cin>>name;
    cin>>temp;
    cin>>sec;
    arr[i].set_values(name,temp,sec);
}

int sum=0;
for(int i=0;i<n;i++)
{
    cout<<get_marks(arr[i])<<" ";
    sum+=get_marks(arr[i]);
}

cout<<sum;
}

```

Output



```

1 /*
2
3 WAP which would contain array of objects [many
4 objects], of a class Student. Student [Name, Age,
5 Year, section, marks], the section would be A,B,C
6 and D. Your program would be able to return the
7 total marks of students in the college.
8 */
9
10 #include<iostream>
11 #include<string>
12
13 using namespace std;
14 class student
15 {
16     //friend function array
17     private:
18         string name;
19         int marks;
20         char sec;
21     public:
22         void set_values(string name,int m,char sec)
23         {
24             this->name=name;
25             this->marks=m;
26             this->sec=sec;
27         }
28         friend int get_marks(student obj);
29     };
30     int get_marks(student obj)
31     {
32         return obj.marks;
33     }
34 int main()
35 {
36     cout << "enter no. of students ";
37     cin >> n;
38     cout << "enter name,marks and section of student ";
39     for(i=0;i<n;i++)
40     {
41         cout << "enter name,marks and section of student ";
42         cin >> name >> marks >> sec;
43         student s;
44         s.set_values(name,marks,sec);
45         sum+=s.get_marks();
46     }
47     cout << "Total marks of all students are ";
48     cout << sum;
49 }

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data Members, friend function in C++)/Task 01 (Class and Object in C++)/Task V02 (WAP which would contain array of objects [many objects], of a class Student. Student [Name, Age, Year, section, marks], the section would be A,B,C and D. Your program would be able to return the total marks of students in the college)\$ gedit TaskV02.cpp
^Z [1]+ Stopped gedit TaskV02.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data Members, friend function in C++)/Task 01 (Class and Object in C++)/Task V02 (WAP which would contain array of objects [many objects], of a class Student. Student [Name, Age, Year, section, marks], the section would be A,B,C and D. Your program would be able to return the total marks of students in the college)\$ g++ TaskV02.cpp -o TaskV02
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data Members, friend function in C++)/Task 01 (Class and Object in C++)/Task V02 (WAP which would contain array of objects [many objects], of a class Student. Student [Name, Age, Year, section, marks], the section would be A,B,C and D. Your program would be able to return the total marks of students in the college)\$./TaskV02
enter no. of students
2
enter name,marks and section of student 1
vishal 78 A
enter name,marks and section of student 2
Knockcat 100 A
78 100 178 knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data Members, friend Function in C++)/Task 01 (Class and Object in C++)/knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data M [Name, Age, Year, seembers, friend function in C++]/Task 01 (Class and Object in C++)/Task V02 (WAP which would contain array of objects [many objects], of a class Student. Student [Name, Age, Year, section, marks], the section would be A,B,C and D. Your program would be able to return the total marks of students in the college)\$

Program 9

Task 2:
Source Code:

```
#include<iostream>
#include<string>
using namespace std;
class cal_area
{
public:
    cal_area(int a)
    {
        int area=4*(a)*(a);
        cout<<"area of square"<<" "<<area<<endl;
    }
    cal_area(int l,int b)
    {
        int area=l*b;
        cout<<"area of rectangle"<<" "<<area<<endl;
    }
};

int main()
{
    cal_area A1(4),A2(3,4);
    return 0;
}
```

Output

```

1 /*
2 WAP to create a class to print the area of a square
3 and a rectangle. The class has two functions with
4 the same name but different number of parameters.
5 The function for printing the area of rectangle has
6 two parameters which are its length and breadth
7 respectively while the other function for printing
8 the area of square has one parameter which is the
9 side of the square. Use multiple constructors to for
10 the initialization.*/
11
12 //function overloading question
13 #include<iostream>
14 #include<string>
15 using namespace std;
16 class cal_area
17 {
18     public:
19         cal_area(int a)
20         {
21             int area=4*(a)*(a);
22             cout<<"area of square"<<
23             "<<area<<endl;
24         }
25         cal_area(int l,int b)
26         {
27             int area=l*b;
28             cout<<"area of rectangle"<<
29             "<<area<<endl;
30         }
31 };
32
33 int main()
34 {
35     cal_area A1(4),A2(3,4);
36     return 0;
37 }

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data Members, friend function in C++)/Task 02 (Constructor in C++)\$ gedit Task02.cpp
^Z [1]+ Stopped gedit Task02.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data Members, friend function in C++)/Task 02 (Constructor in C++)\$ g++ Task02.cpp -o Task02
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data Members, friend function in C++)/Task 02 (Constructor in C++)\$./Task02
area of square 64
area of rectangle 12
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data Members, friend function in C++)/Task 02 (Constructor in C++)\$

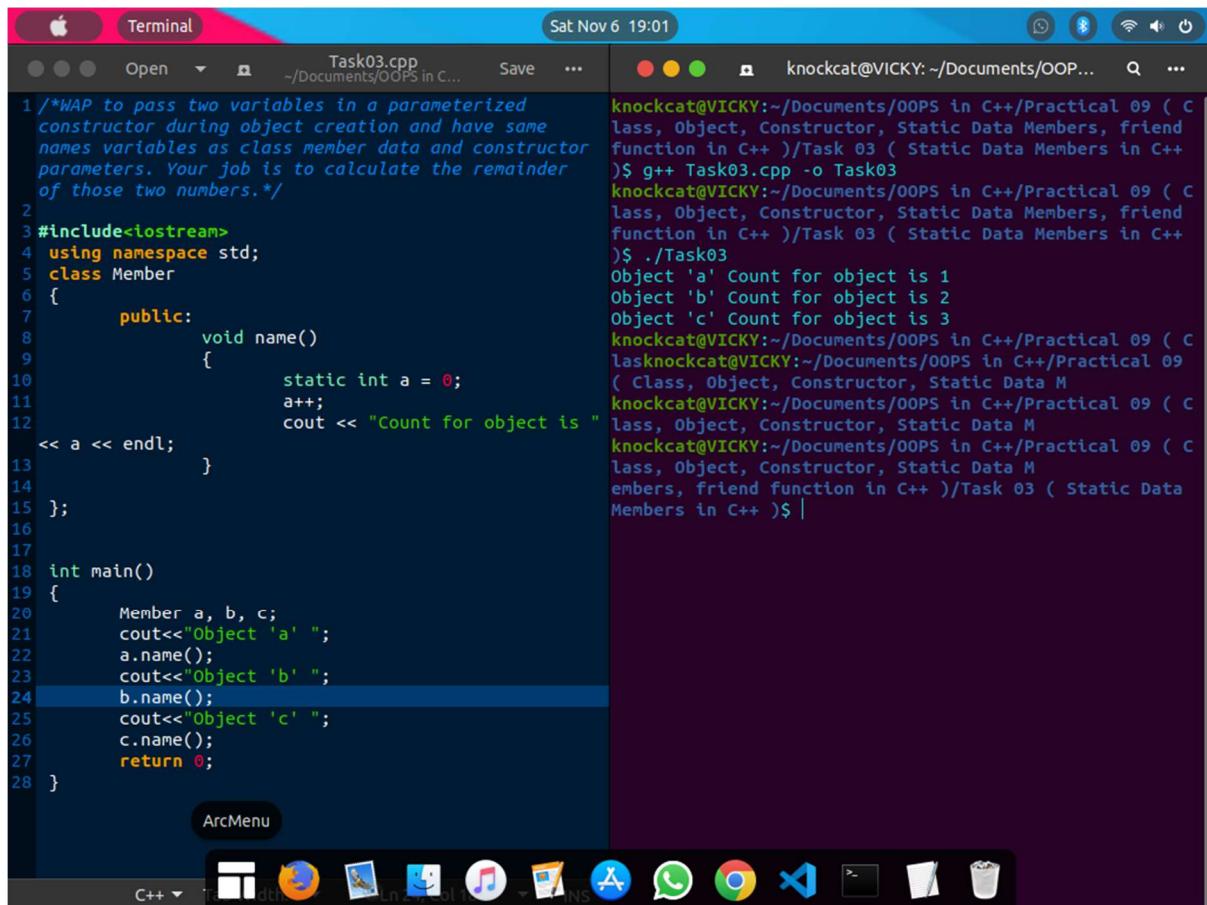
Program 9

Task 3:
Source Code:

```
#include<iostream>
using namespace std;
class Member
{
public:
    void name()
    {
        static int a = 0;
        a++;
        cout << "Count for object is " << a << endl;
    }
};

int main()
{
    Member a, b, c;
    cout << "Object 'a' ";
    a.name();
    cout << "Object 'b' ";
    b.name();
    cout << "Object 'c' ";
    c.name();
    return 0;
}
```

Output



The screenshot shows a macOS desktop environment with a Terminal window open. The window title is "Terminal" and the path is "Task03.cpp ~/Documents/OOPS in C...". The date and time at the top right are "Sat Nov 6 19:01". The terminal content displays the following C++ code and its execution output:

```

1 /*WAP to pass two variables in a parameterized
constructor during object creation and have same
names variables as class member data and constructor
parameters. Your job is to calculate the remainder
of those two numbers.*/
2
3 #include<iostream>
4 using namespace std;
5 class Member
6 {
7     public:
8         void name()
9         {
10             static int a = 0;
11             a++;
12             cout << "Count for object is "
13             << a << endl;
14         }
15 };
16
17
18 int main()
19 {
20     Member a, b, c;
21     cout << "Object 'a' ";
22     a.name();
23     cout << "Object 'b' ";
24     b.name();
25     cout << "Object 'c' ";
26     c.name();
27     return 0;
28 }

```

The output of the code is:

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (~ Class, Object, Constructor, Static Data Members, friend function in C++)/Task 03 (~ Static Data Members in C++)
$ g++ Task03.cpp -o Task03
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (~ Class, Object, Constructor, Static Data Members, friend function in C++)/Task 03 (~ Static Data Members in C++)
$ ./Task03
Object 'a' Count for object is 1
Object 'b' Count for object is 2
Object 'c' Count for object is 3
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (~ Class, Object, Constructor, Static Data Members, friend function in C++)/Task 03 (~ Static Data Members in C++)
$ knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (~ Class, Object, Constructor, Static Data Members, friend function in C++)/Task 03 (~ Static Data Members in C++)
$ knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (~ Class, Object, Constructor, Static Data Members, friend function in C++)/Task 03 (~ Static Data Members in C++)
$ |

```

The desktop dock at the bottom includes icons for Finder, Mail, Safari, Calendar, iBooks, iTunes, iPhoto, iMovie, iDVD, iWork, iLife, iTools, and a trash can.

Program 9

Task 4:
Source Code:

```
#include<iostream>
#include<string.h>
using namespace std;

class student
{
private:
    string name;
    int marks;
    char sec;
public:
    student(string name,int m,char sec)
    {
        this->name=name;
        this->marks=m;
        this->sec=sec;
    }
    friend int get_marks(student);
};

int get_marks(student obj)
{
    return obj.marks;
}

int main()
{
    student obj1("swati",99,'A');
    student obj2("shivi",84,'B');
    student obj3("sneha",78,'C');
    int sum=0;
    sum+=get_marks(obj1);
    sum+=get_marks(obj2);
    sum+=get_marks(obj3);
    cout<<"total sum of all students marks : "<<sum;
    return 0;
}
```

Output

The screenshot shows a Mac OS X desktop environment. On the left, a "Text Editor" window is open, displaying the code for "Task04.cpp". The code defines a "student" class with private attributes (name, marks, sec) and a public constructor. It includes a friend function "get_marks" that returns the marks of a student object. In the main() function, three student objects are created and their marks are printed. On the right, a terminal window shows the command-line session where the code is compiled (using g++) and run (./Task04). The output of the program is displayed, showing the total sum of all students' marks as 261.

```

1 /*
2 WAP in which you create a Student class having
3 basic information for each student, like name, age
4 and marks. By using friend function add marks of
5 all the students [lets say 3 objects] and print it.
6 */
7
8 #include<iostream>
9 #include<string.h>
10 using namespace std;
11
12 class student
13 {
14     private:
15         string name;
16         int marks;
17         char sec;
18     public:
19         student(string name,int m,char sec)
20         {
21             this->name=name;
22             this->marks=m;
23             this->sec=sec;
24         }
25         friend int get_marks(student);
26 };
27
28 int get_marks(student obj)
29 {
30     return obj.marks;
31 }
32
33 int main()
34 {
35     student obj1("swati",99,'A');
36     student obj2("shivi",84,'B');
37     student obj3("sneha",78,'C');
38 }
```

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 ( Class, Object, Constructor, Static Data Members, friend function in C++ )/Task 04 ( Friend Function in C++ )$ gedit Task04.cpp
^Z
[1]+  Stopped                  gedit Task04.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 ( Class, Object, Constructor, Static Data Members, friend function in C++ )/Task 04 ( Friend Function in C++ )$ g++ Task04.cpp -o Task04
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 ( Class, Object, Constructor, Static Data Members, friend function in C++ )/Task 04 ( Friend Function in C++ )$ ./Task04
total sum of all students marks : 261
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 ( Class, Object, Constructor, Static Data Members, friend function in C++ )/Task 04 ( Friend Function in C++ )$ |
```

Program 9

Task 5:
Source Code:

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

class college
{
    string name;
    int roll_no;
};

struct student
{
    string name;
    int roll_no;
};

int main()
{
    college t;
    student s1;
    t.name="shivi";// compiler error because x is
private
    t.roll_no = 20; //error concept of data hiding
    s1.name="shreay";
    s1.roll_no=34;
    getchar();
    return 0;
}
```

Output

The screenshot shows a Mac OS X desktop with two Text Editor windows and a terminal window.

Left Window (Task05.cpp):

```

1 /*
2 WAP to create a College class and Student Structure
3 in C++ in one program. By providing such suitable
4 examples write at least 5 differences between class
5 and struct code you have written above.
6 Hint [Access Specifiers, Heap and Stack, large and
7 small memory, etc.]*/
8
9 class college
10 {
11     string name;
12     int roll_no;
13 };
14
15 struct student
16 {
17     string name;
18     int roll_no;
19 };
20
21 int main()
22 {
23     college t;
24     student s1;
25     t.name="shivi";// compiler error because x
26     //is private
27     t.roll_no = 20; //error concept of data
28     //hiding
29     s1.name="shreay";
30     s1.roll_no=34;
31     getchar();
32     return 0;
33 }

```

Right Window (difference.txt):

- 3 1. access specifier :- a structure is a class defined with struct keyword and access specifier is by default public whereas when class is derived access specifier is by default private.
- 4 2. security :- data members in class are private the concept of data hiding because we cannot access data members directly, on the other hand data members of structure are public by default which shows a lack of security in using structure.
- 5 3. structure is a value type data type and class is a reference type data type instances of stack are stored in stack whereas in class instances are stored in heap as the references of objects are stored in heap.
- 6 4. inheritance :- structure cannot inherit other structures or classes whereas class can inherit other classes and struct.

Terminal Window:

```

object, Constructor, Static Data Members, friend function i
+ )/Task 05 ( Structure in C++ )$ g++ Task05.cpp -o Task05
Task05.cpp: In function 'int main()':
Task05.cpp:25:4: error: 'std::string college::name' is pri
within this context
  25 |     t.name="shivi";// compiler error because x is pri
      |     ^
Task05.cpp:11:12: note: declared private here
  11 |     string name;
      |     ^
Task05.cpp:26:6: error: 'int college::roll_no' is private
in this context
  26 |     t.roll_no = 20; //error concept of data hiding
      |     ^
Task05.cpp:12:9: note: declared private here
  12 |     int roll_no;
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 ( Clas
object, Constructor, Static Data Members, friknockcat@VICKY

```

Program 9

Task 6:
Source Code:

```
#include<iostream>
#include<iomanip>
#include<ios>
using namespace std;

int main()
{
    cout<<"enter our choice"<<endl;
    cout<<"1.come to next line 2.setting field with 3.fill string with * after setw(15)
function" <<endl;
    int n;

    long str=123456789;
    cout<<"before before any operation "<<str;
    while(1)
    {
        cin>>n;
        switch(n)
        {
            case 1 :
                cout<<"before going to next line "<<str;
                cout<<endl;
                cout<<str;
                break;
            case 2:
                cout<<"after setting field width"<<str<<endl;
                cout<<setw(15);
                cout<<str;
                break;
            case 3:
                cout<<"before setfill "<<str<<endl;
                cout<<setfill('*')<<setw(15);
                cout<<str;
                break;
            default : return 0;
        }
    }
}
```

Output

```

1 /*WAP which would perform these tasks of your data:
2 a. Come to next line
3 b. set minimum field width
4 c. fill string with (*) after setw(15) function
5      *****1234
6 by using endl, setw, and setfill [Manipulators in C++
+] */
7
8 #include<iostream>
9 #include<iomanip>
10 #include<ios>
11 using namespace std;
12
13 int main()
14 {
15     cout<<"enter our choice"<<endl;
16     cout<<"1.come to next line 2.setting field with
17 .fill string with * after setw(15) function"
18     <<endl;
19     int n;
20
21     long str=123456789;
22     cout<<"before before any operation "<<str;
23     while(1)
24     {
25         cin>>n;
26         switch(n)
27         {
28             case 1 :
29                 cout<<"before going to next line "<<str;
30                 cout<<endl;
31                 cout<<str;
32                 break;
33             case 2:
34                 cout<<"after setting field width"<<str<<endl;
35                 cout<<setw(15);
36                 cout<<str;
37         }
38     }
39 }
```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data Members, friend function in C++)/Task 06 (Extra Questions)\$ gedit Task06.cpp
^Z
[1]+ Stopped gedit Task06.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data Members, friend function in C++)/Task 06 (Extra Questions)\$ g++ Task06.cpp -o Task06
knockcat@VICKY:~/Documents/OOPS in C++/Practical 09 (Class, Object, Constructor, Static Data Members, friend function in C++)/Task 06 (Extra Questions)\$./Task06
enter our choice
1.come to next line 2.setting field with 3.fill string with * after setw(15) function
before before any operation 123456789
1
before going to next line 123456789
123456789
2
after setting field width123456789
123456789
3
before setfill 123456789
*****123456789
4

Program 10

Task 1:
Source Code:

```

#include<iostream>
#include<string>

using namespace std;
class student
{ //friend function array

private:
    string name;
    string address;
    long int telephone;
    long int Mobile_no;
    string head_of_family;

public:
void set_values(string name,string add,long int t,
long int Mob, string head)
{
    this->name=name;
    this->address=add;
    this->telephone=t;
    this->Mobile_no=Mob;
    this->head_of_family=head;
}

void display()
{
    cout << name << endl;
    cout << address << endl;
    cout << telephone << endl;
    cout << Mobile_no << endl;
    cout << head_of_family << endl;
}

};


```

```

int main()
{
    int n;
    cout<<"enter no. of Family"<<endl;
    cin>>n;
    student arr[n];

    for(int i=0;i<n;i++)
    {
        cout<<"enter name,address, telephone ,
Mobile_no and head_of_family :\n "<<i+1<<endl;
        string name;
        string address;
        long int telephone;
        long int Mobile_no;
        string head_of_family;

        cin>>name;
        cin>>address;
        cin>>telephone;
        cin>>Mobile_no;
        cin>>head_of_family;

        arr[i].set_values(name,address,telephone,Mobile_
no,head_of_family);
    }

    for(int i=0;i<n;i++)
    {
        cout<<"\nDetails of Family "<<i+1<<endl;
        arr[i].display();
    }

    return 0;
}

```

Output

```

1 /*WAP to create a directory that contains the
2 (a) Name of a person
3 (b) Address
4 (c) Telephone Number (if available with STD code)
5 (d) Mobile Number (if available)
6 (e) Head of the family*/
7
8 #include<iostream>
9 #include<string>
10
11 using namespace std;
12 class student
13 { //friend function array
14     private:
15         string name;
16         string address;
17         long int telephone;
18         long int Mobile_no;
19         string head_of_family;
20     public:
21         void set_values(string name,string add,long
22         int t, long int Mob, string head)
23         {
24             this->name=name;
25             this->address=add;
26             this->telephone=t;
27             this->Mobile_no=Mob;
28             this->head_of_family=head;
29         }
30         void display()
31         {
32             cout << name << endl;
33             cout << address << endl;
34             cout << telephone << endl;
35             cout << Mobile_no << endl;

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 (A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++)/Task 01 (Array of Objects in
C++)\$ g++ Task01.cpp -o Task01
knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 (A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++)/Task 01 (Array of Objects in
C++)\$./Task01
enter no. of Family
2
enter name,address, telephone , Mobile_no and head_of_f
amily :
1
Joshi's dehradun 8976548990 9087653489 Father
enter name,address, telephone , Mobile_no and head_of_f
amily :
2
Pandey's nanital 8976899908 8790875698 Father

Details of Family 1
Joshi's
dehradun
8976548990
9087653489
Father

Details of Family 2
Pandey's
nanital
8976899908
8790875698
Father

knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 (A
rray of Objects, Pointer to Object, This
knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 (A
rray of Objects, Pointer to Object, This
knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 (A
rray of Objects, Pointer to Object, This

C++ ▾ Tab Width: 8 ▾ Ln 7, Col 1 ▾ INS

Program 10

Task 2:
Source Code:

```
#include<iostream>
using namespace std;
class student
{
    private:
        string name;
        string add;
        int roll_no;
        char sec;
    public:
        void set_values()
        {
            getline(cin,name);
            getline(cin,add);
            cin>>roll_no;
            cin>>sec;
        }
        void show_data()
        {
            cout<<name<<" "<<add<<" "<<roll_no<<" "<<sec;
        }
};

int main()
{
    student *ptr;
    student obj1;
    ptr=&obj1;
    ptr->set_values();
    ptr->show_data();
    return 0;
}
```

Output

```

1 /*WAP to create print or display Student
   information containing in Student class by using
   pointers to object.*/
2
3 #include<iostream>
4 using namespace std;
5 class student
6 {
7     private:
8         string name;
9         string add;
10        int roll_no;
11        char sec;
12    public:
13    void set_values()
14    {
15        getline(cin,name);
16        getline(cin,add);
17        cin>>roll_no;
18        cin>>sec;
19    }
20    void show_data()
21    {
22        cout<<name<<" "<<add<<" "<<roll_no<<""
23        <<sec;
24    }
25
26 int main()
27 {
28     student *ptr;
29     student obj1;
30     ptr=&obj1;
31     ptr->set_values();
32     ptr->show_data();
33     return 0;
34 }

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 (A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++)/Task 02 (Pointer to Object in
C++)\$ gedit Task02.cpp
^Z
[1]+ Stopped gedit Task02.cpp
knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 (A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++)/Task 02 (Pointer to Object in
C++)\$ g++ Task02.cpp -o Task02
knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 (A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++)/Task 02 (Pointer to Object in
C++)\$./Task02
Knockcat
69
63
A
Knockcat 69 63 Aknockcat@VICKY:~/Documents/OOPS in C++/
Practical 10 (Array of Objects, Pointer to Object, Thi
s pointer, Operator Overloading in C++)/Task 02 (Poin
ter to Object in C++)\$./Task02
Vishu
ddun
1
N
Vishu ddun 1 N knockcat@VICKY:~/Documents/OOPS in C++/Pr
actical 10 (Array of Objects, Pointer to Object, This
pointer knockcat@VICKY:~/Documents/OOPS in C++/Practical
10 (Array of Objects, Pointer to Object, This
pointer knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 (A
rray of Objects, Pointer to Object, This
pointer, Operator Overloading in C++)/Task 02 (Point
er to Object in C++)\$ |

C++ ▾ Tab Width: 8 ▾ Ln 33, Col 18 ▾ INS

Program 10

Task 3:

Source Code:

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

class rem
{
    private:
        int a,b;
    public:
        rem(int a,int b)
        {
            this->a=a;
            this->b=b;
            cout<<"\n" <<this->a / this->b << endl;;
        }
};

int main()
{
    rem obj(78,23);
    return 0;
}
```

Output

```

Text Editor   Sat Nov 6 18:02
Open  Task03.cpp  Save  ...
~/Documents/OOPS in C...
knockcat@VICKY:~/Documents/OOPS...  Q  ...
knockcat@VICKY:~/Documents/OOPS in C+/Practical 10 ( A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++ )/Task 03 ( This pointer in C++
)$ g++ Task03.cpp -o Task03
knockcat@VICKY:~/Documents/OOPS in C+/Practical 10 ( A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++ )/Task 03 ( This pointer in C++
)$ ./Task03
3
knockcat@VICKY:~/Documents/OOPS in C+/Practical 10 ( A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++ )/Task 03 ( This pointer in C++
)$ |

```

C++ ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

Program 10

Task 4 a:

Source Code:

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

class abc
{
    int a;
public:
    abc(int a)
    {
        this->a=a;
    }
    friend void operator < (abc &obj,abc &obj1);
};

void operator < (abc &obj,abc &obj1)
{
    if(obj.a<obj1.a)
        cout<<"true"<<endl;
    else
        cout<<"false"<<endl;
}

int main()
{
    abc obj(56);
    abc obj1(5);
    obj<obj1;
    return 0;
}
```

Output

```

1/*WAP, in which you write a friend function to
2   overload a less than '<' operator in C++.*/
3
4#include<iostream>
5using namespace std;
6
7class abc
8{
9    int a;
10   public:
11       abc(int a)
12       {
13           this->a=a;
14       }
15       friend void operator < (abc &obj,abc &obj1);
16 };
17}
18
19void operator < (abc &obj,abc &obj1)
20{
21     if(obj.a<obj1.a)
22         cout<<"true"<<endl;
23     else
24         cout<<"false"<<endl;
25 }
26
27int main()
28{
29     abc obj(56);
30     abc obj1(5);
31     obj<obj1;
32     return 0;
33 }
34
35
knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 ( A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++ )/Task 04 ( Operator Overloading
in C++ )/Task V02 ( overloading + operator )$ g++ Task
V02.cpp -o TaskV02
knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 ( A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++ )/Task 04 ( Operator Overloading
in C++ )/Task V02 ( overloading + operator )$ ./TaskV02
2
false
knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 ( A
rray of Objects, Pointer to Object, This Overloading
in C++ ) pointer,knockcat@VICKY:~/Documents/OOPS in C+
+Practical 10 ( Array of Objects, Pointer to Object, T
his
knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 ( A
rray of Objects, Pointer to Object, This
pointer, Operator Overloading in C++ )/Task 04 ( Opera
tor Overloading in C++ )/Task V02 ( over
loading + operator )$ |

```

C++ ▾ Tab Width: 8 ▾ Ln 32, Col 18 ▾ INS

Program 10**Task 4 b:****Source Code:**

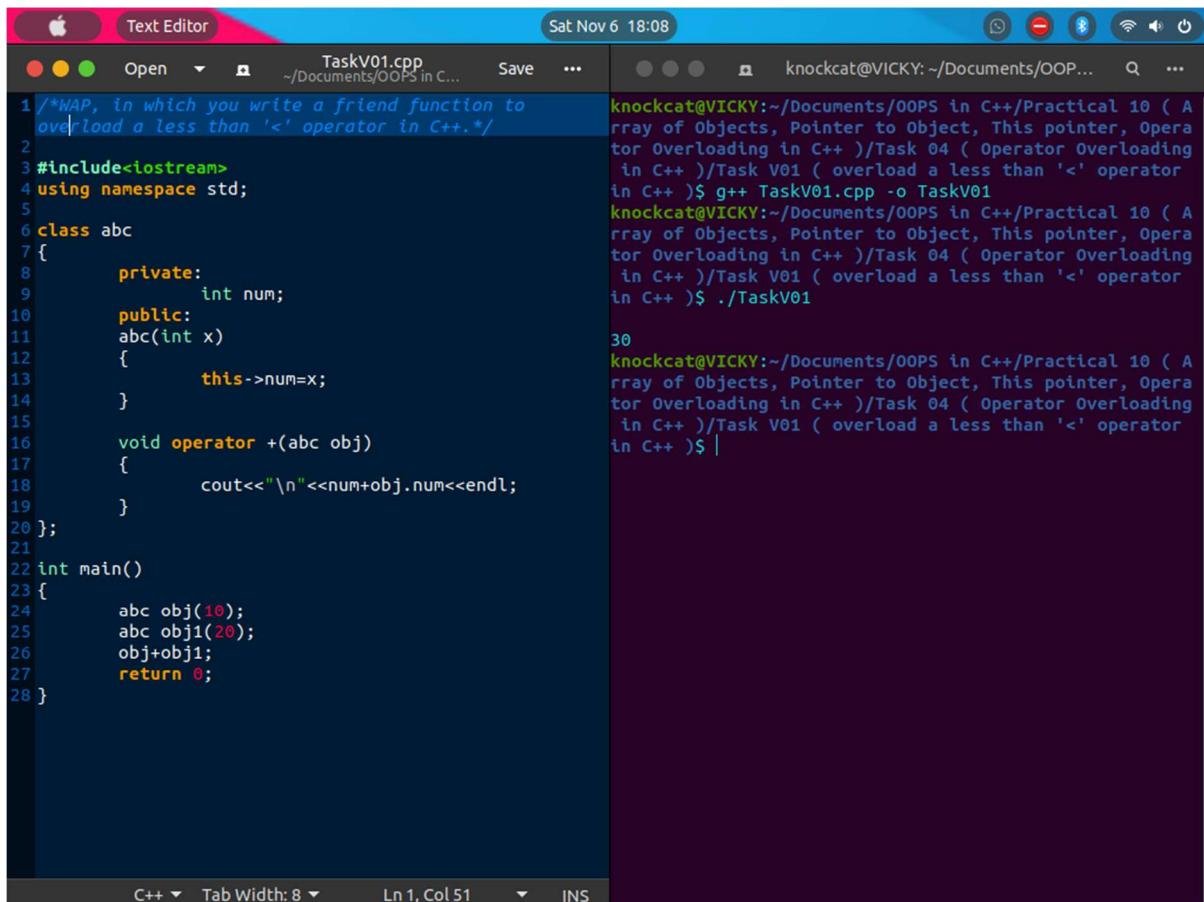
```
#include<iostream>
using namespace std;

class abc
{
    private:
        int num;
    public:
        abc(int x)
        {
            this->num=x;
        }

        void operator +(abc obj)
        {
            cout<<"\n"<<num+obj.num<<endl;
        }
};

int main()
{
    abc obj(10);
    abc obj1(20);
    obj+obj1;
    return 0;
}
```

Output



The screenshot shows a Mac OS X Text Editor window titled "Text Editor". The file being edited is "TaskV01.cpp" located at "/Documents/OOPS in C...". The code implements a class `abc` with a friend function to overload the less than operator (<). The terminal window to the right shows the command `g++ TaskV01.cpp -o TaskV01` being run, followed by the execution of the program `./TaskV01`, which outputs the result of the overloaded operator.

```

1 /*WAP, in which you write a friend function to
2  overload a less than '<' operator in C++.*/
3 #include<iostream>
4 using namespace std;
5
6 class abc
7 {
8     private:
9         int num;
10    public:
11        abc(int x)
12        {
13            this->num=x;
14        }
15        void operator +(abc obj)
16        {
17            cout<<"\n"<<num+obj.num<<endl;
18        }
19    };
20
21
22 int main()
23 {
24     abc obj(10);
25     abc obj1(20);
26     obj+obj1;
27     return 0;
28 }

```

```

knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 ( A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++ )/Task 04 ( Operator Overloading
in C++ )/Task V01 ( overload a less than '<' operator
in C++ )$ g++ TaskV01.cpp -o TaskV01
knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 ( A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++ )/Task 04 ( Operator Overloading
in C++ )/Task V01 ( overload a less than '<' operator
in C++ )$ ./TaskV01
30
knockcat@VICKY:~/Documents/OOPS in C++/Practical 10 ( A
rray of Objects, Pointer to Object, This pointer, Opera
tor Overloading in C++ )/Task 04 ( Operator Overloading
in C++ )/Task V01 ( overload a less than '<' operator
in C++ )$ |

```

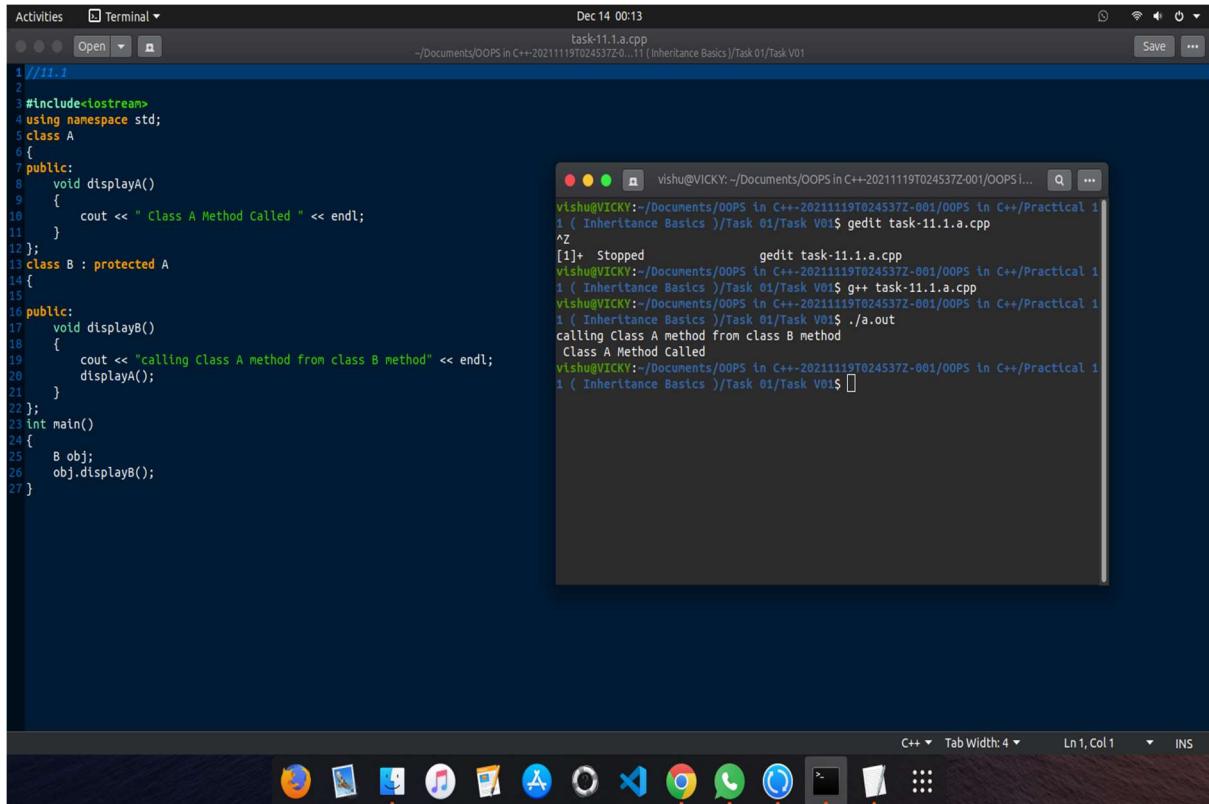
Program 11

Task 11.1.a:

Source Code:

```
#include<iostream>
using namespace std;
class A
{
public:
    void displayA()
    {
        cout << " Class A Method Called " << endl;
    }
};
class B : protected A
{
public:
    void displayB()
    {
        cout << "calling Class A method from class B
method" << endl;
        displayA();
    }
};
int main()
{
    B obj;
    obj.displayB();
}
```

Output



The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor Window:

```
1 //11.1
2
3 #include<iostream>
4 using namespace std;
5 class A
6 {
7 public:
8     void displayA()
9     {
10         cout << " Class A Method Called " << endl;
11     }
12 };
13 class B : protected A
14 {
15 public:
16     void displayB()
17     {
18         cout << "calling Class A method from class B method" << endl;
19         displayA();
20     }
21 };
22 int main()
23 {
24     B obj;
25     obj.displayB();
26 }
```

Terminal Window:

```
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1 ( Inheritance Basics )/Task 01/Task V01$ gedit task-11.1.a.cpp
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1 ( Inheritance Basics )/Task 01/Task V01$ g++ task-11.1.a.cpp
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1 ( Inheritance Basics )/Task 01/Task V01$ ./a.out
calling Class A method from class B method
Class A Method Called
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1 ( Inheritance Basics )/Task 01/Task V01$
```

Task 11.1.b:**Source Code:**

```
#include <iostream>
using namespace std;
class A
{
public:
    void displayA()
    {
        cout << " Class A Method Called " << endl;
    }
};
class B : public A
{
public:
    void displayB()
    {
        cout << "calling Class A method from class B
method" << endl;
        displayA();
    }
};
int main()
{
    B obj;
    obj.displayA();
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. In the top left, there's an 'Activities' button and a 'Terminal' window tab. The terminal window is open to a file named 'task-11.1.b.cpp' located at '/Documents/OOPS in C++-20211119T024537Z-001/OOPS i...'. The terminal output shows the execution of the program, which prints 'Class A Method Called' and then 'calling Class A method from class B method'. Below the terminal is a code editor window for 'gedit task-11.1.b.cpp'. The code defines two classes, A and B, where B inherits from A. The main function creates an object of type B and calls its displayA() method. The code editor has tabs for C++ and Tab Width: 4, and shows line 1, column 9.

```

1 //11.1.b
2
3 #include <iostream>
4 using namespace std;
5 class A
6 {
7 public:
8     void displayA()
9     {
10         cout << " Class A Method Called " << endl;
11     }
12 };
13 class B : public A
14 {
15
16 public:
17     void displayB()
18     {
19         cout << "calling Class A method from class B method" << endl;
20         displayA();
21     }
22 };
23 int main()
24 {
25     B obj;
26     obj.displayA();
27 }

```

Task 11.2:

Source Code:

```
#include <iostream>
using namespace std;
class MarrineAnimals
{
public:
    void displayMA()
    {
        cout << " I am Marrine Animal " << endl;
    }
};

class Mammals
{

public:
    void displayM()
    {
        cout << " I am Mammal " << endl;
    }
};

class BlueWhale : public MarrineAnimals, public Mammals
{

public:
    void displayB()
    {
        cout << " I belong to both categories Mammals as well
as Marine Animals " << endl;
    }
};

int main()
{
    Mammals ob1;
    MarrineAnimals ob2;
    BlueWhale ob3;
    ob1.displayM();
    ob2.displayMA();
    ob3.displayB();
    ob3.displayM();
    ob3.displayMA();
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor Window:

```

1 //task-11.2
2
3 #include <iostream>
4 using namespace std;
5 class MarineAnimals
6 {
7 public:
8     void displayMA()
9     {
10         cout << " I am Marine Animal " << endl;
11     }
12 };
13 class Mammals
14 {
15 public:
16     void displayM()
17     {
18         cout << " I am Mammal " << endl;
19     }
20 };
21 class BlueWhale : public MarineAnimals, public Mammals
22 {
23
24 public:
25     void displayB()
26     {
27         cout << " I belong to both categories Mammals as well as Marine Animals " << endl;
28     }
29 };
30 int main()
31 {
32     Mammals ob1;
33     MarineAnimals ob2;
34     BlueWhale ob3;
35     ob1.displayM();
36     ob2.displayMA();
37     ob3.displayB();
38     ob3.displayM();
39     ob3.displayMA();
40     ob3.displayMA();
41 }

```

Terminal Window:

```

vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
$ gedit task-11.2.cpp
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
$ g++ task-11.2.cpp
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
$ ./a.out
I am Mammal
I am Marine Animal
I belong to both categories Mammals as well as Marine Animals
I am Mammal
I am Marine Animal
I am Mammal
I am Marine Animal

```

Task 11.3.a:**Source Code:**

```
#include <iostream>
using namespace std;
class A{
public:
    int k;
    void displayA()
    {
        cout << "k = " << k << endl;
    }
};
class B : public A
{
public:
    void displayB()
    {
        k = 10;
        cout << "k in B= " << k << endl;
    }
};
class C : public A
{
public:
    void displayc()
    {
        k = 20;
        cout << "k in C = " << k << endl;
    }
};
class D : public B, public C
{
public:
    void displayD()
    {
        k = 30;
        cout << "k in D= " << k << endl;
    }
};
int main(){
    D obj;
    obj.displayD();
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Terminal Window:

```

Activities Terminal ▾
Open ▾
task-11.3.a.cpp
Dec 14 00:22
task-11.3.a.cpp
~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
1 ( Inheritance Basics )/Task 03 ( Diamond Problem in Multiple Inheritance )/Task V01
k V01$ gedit task-11.3.a.cpp
[1]+ Stopped                  gedit task-11.3.a.cpp
vishu@VICKY: ~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
1 ( Inheritance Basics )/Task 03 ( Diamond Problem in Multiple Inheritance )/Task V01
k V01$ g++ task-11.3.a.cpp
task-11.3.a.cpp: In member function 'void D::displayD()':
task-11.3.a.cpp:34:9: error: reference to 'k' is ambiguous
  34 |         k = 30;
      |         ^
task-11.3.a.cpp:5:9: note: candidates are: 'int A::k'
  5 |     int k;
      |
task-11.3.a.cpp:5:9: note:                      'int A::k'
task-11.3.a.cpp:35:31: error: reference to 'k' is ambiguous
  35 |         cout << "k in D= " << k << endl;
      |         ^
task-11.3.a.cpp:5:9: note: candidates are: 'int A::k'
  5 |     int k;
      |
task-11.3.a.cpp:5:9: note:                      'int A::k'
vishu@VICKY: ~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
1 ( Inheritance Basics )/Task 03 ( Diamond Problem in Multiple Inheritance )/Task V01
k V01$ 

```

Code Editor Window:

```

1 #include <iostream>
2 using namespace std;
3 class A{
4 public:
5     int k;
6     void displayA()
7     {
8         cout << "k = " << k << endl;
9     }
10 };
11 class B : public A
12 {
13 public:
14     void displayB()
15     {
16         k = 10;
17         cout << "k in B= " << k << endl;
18     }
19 };
20 class C : public A
21 {
22 public:
23     void displayC()
24     {
25         k = 20;
26         cout << "k in C= " << k << endl;
27     }
28 };
29 class D : public B, public C
30 {
31 public:
32     void displayD()
33     {
34         k = 30;
35         cout << "k in D= " << k << endl;
36     }
37 };
38 int main()
39 {
40     D obj;
41     obj.displayD();
42 }

```

Task 11.3.b:**Source Code:**

```
#include <iostream>
using namespace std;
class A{
public:
    int k;
    void displayA()
    {
        cout << "k = " << k << endl;
    }
};
class B : virtual public A
{
public:
    void displayB()
    {
        k = 10;
        cout << "k in B= " << k << endl;
    }
};
class C : virtual public A
{
public:
    void displayc()
    {
        k = 20;
        cout << "k in C = " << k << endl;
    }
};
class D : public B, public C
{
public:
    void displayD()
    {
        cout << "k in D " << k << endl;
    }
};
int main() {
    D obj;
    obj.displayD();
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 #include <iostream>
2 using namespace std;
3 class A{
4 public:
5     int k;
6     void displayA()
7     {
8         cout << "k = " << k << endl;
9     }
10 };
11 class B : virtual public A
12 {
13 public:
14     void displayB()
15     {
16         k = 10;
17         cout << "k in B= " << k << endl;
18     }
19 };
20 class C : virtual public A
21 {
22 public:
23     void displayC()
24     {
25         k = 20;
26         cout << "k in C = " << k << endl;
27     }
28 };
29 class D : public B, public C
30 {
31 public:
32     void displayD()
33     {
34
35         cout << "k in D " << k << endl;
36     }
37 };
38 int main()
39 {
40     D obj;
41     obj.displayD();
42 }

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
1 ( Inheritance Basics )/Task 03 ( Diamond Problem in Multiple Inheritance )/Task 03
k Vo$ g++ task-11.3.b.cpp
[1]+  Killed                  gedit task-11.3.b.cpp
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
1 ( Inheritance Basics )/Task 03 ( Diamond Problem in Multiple Inheritance )/Task 03
k Vo$ g++ task-11.3.b.cpp
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
1 ( Inheritance Basics )/Task 03 ( Diamond Problem in Multiple Inheritance )/Task 03
k Vo$ ./a.out
k in D -1377963392
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
1 ( Inheritance Basics )/Task 03 ( Diamond Problem in Multiple Inheritance )/Task 03
k Vo$ 

```

Task 11.3.c:

Source Code:

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

class B
{
public:
    int k;
    void displayB()
    {
        cout << "k in B= " << k << endl;
    }
};

class C
{
public:
    int k;
    void displayc()
    {
        cout << "k in C = " << k << endl;
    }
};

class D : public B, public C
{
public:
    void displayD()
    {
        cout << "k of B and C inherited in D => " << B::k << "
" << C::k << endl;
    }
};

int main()
{
    D obj;
    obj.B::k = 12;
    obj.C::k = 11;
    obj.displayD();
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Terminal Window:

```
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
1 ( Inheritance Basics )/Task 03 ( Diamond Problem in Multiple Inheritance )/Task V03
k V03$ g++ task-11.3.c.cpp
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
1 ( Inheritance Basics )/Task 03 ( Diamond Problem in Multiple Inheritance )/Task
k V03$ ./a.out
k of B and C inherited in D => 12 11
vishu@VICKY:~/Documents/OOPS in C++-20211119T024537Z-001/OOPS in C++/Practical 1
1 ( Inheritance Basics )/Task 03 ( Diamond Problem in Multiple Inheritance )/Task
k V03$
```

Code Editor (task-11.3.c.cpp):

```
1 #include <iostream>
2 using namespace std;
3
4 class B
5 {
6
7 public:
8     int k;
9     void displayB()
10    {
11        cout << "k in B= " << k << endl;
12    }
13 };
14
15 class C
16 {
17 public:
18     int k;
19     void displayC()
20    {
21        cout << "k in C = " << k << endl;
22    }
23 };
24 class D : public B, public C
25 {
26 public:
27     void displayD()
28    {
29        cout << "k of B and C inherited in D => " << B::k << " " << C::k << endl;
30    }
31 };
32 int main()
33 {
34     D obj;
35     obj.B::k = 12;
36     obj.C::k = 11;
37     obj.displayD();
38 }
```

The code implements multiple inheritance where class D inherits from classes B and C. The program outputs the value of k from both classes, which is 12 followed by 11, demonstrating the diamond problem.

Program 12

Task 12.1.a: Source Code:

```
#include<iostream>
using namespace std;
class A{
    private:
        int a;
    public:
        int b;
    protected:
        int c;
};
class B:protected A{

public:
    void display(){
        b=10;
        c=15;
        cout<<"b=""<<b<<" c=""<<c;
    }
};
int main(){
    B obj;
    obj.display();
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. At the top, there is a header bar with 'Activities' and 'Terminal'. A terminal window is open, titled 'task-12.1.a.cpp', showing the command 'gedit task-12.1.a.cpp' and its output. The code editor window, also titled 'task-12.1.a.cpp', displays C++ code related to inheritance and access modifiers. The code includes classes A and B, with protected members b and c, and a main function that creates an object of class B and calls its display method. The terminal output shows the execution of the program, which prints 'b=10 c=15'.

```

1 //12.1.a
2
3 #include<iostream>
4 using namespace std;
5 class A{
6     private:
7     int a;
8     public:
9     int b;
10    protected:
11    int c;
12 };
13 class B:protected A{
14
15     public:
16     void display(){
17         b=10;
18         c=15;
19         cout<<"b=<<b<<" c=<<c;
20     }
21 };
22 int main(){
23     B obj;
24     obj.display();
25 }

```

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 01 ( Role of Access Modifiers )/Task V01$ gedit task-12.1.a.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 01 ( Role of Access Modifiers )/Task V01$ g++ task-12.1.a.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 01 ( Role of Access Modifiers )/Task V01$ ./a.out
b=10 c=15vishu@VICKY:~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 01 ( Role of Access Modifiers )/Task V01$ 

```

Task 12.1.b:
Source Code:

```
#include<iostream>
using namespace std;
class A{
    private:
        int a;
    public:
        int b;
    protected:
        int c;
};
class B:public A{

    public:
        void display(){
            cout<<"b="<<b;
        }
};
int main(){
    B obj;
    obj.b=10;
    obj.display();
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there is an 'Activities' button, a 'Terminal' button, and several application icons. The terminal window is open and displays the following text:

```

Activities Terminal ▾ Open ▾ Dec 14 23:03
task-12.1.b.cpp
-/Documents/OOPS in C++/Practical 12 (Inheritance)/Task 01 (Role of Access Modifiers)/Task V02
Save ...
```

The code editor window shows the following C++ code:

```

1 //12.1.b
2
3 #include<iostream>
4 using namespace std;
5 class A{
6     private:
7     int a;
8     public:
9     int b;
10    protected:
11    int c;
12 };
13 class B:public A{
14
15     public:
16     void display(){
17         cout<<"b="<<b;
18     }
19 };
20 int main(){
21     B obj;
22     obj.b=10;
23     obj.display();
24 }
```

The terminal window also shows the command-line session:

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 12 (Inheritance)/Task 01 (Role of Access Modifiers)/Task V02$ gedit task-12.1.b.cpp
^Z
[1]+  Stopped                  gedit task-12.1.b.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 12 (Inheritance)/Task 01 (Role of Access Modifiers)/Task V02$ g++ task-12.1.b.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 12 (Inheritance)/Task 01 (Role of Access Modifiers)/Task V02$ ./a.out
b=10vishu@VICKY:~/Documents/OOPS in C++/Practical 12 (Inheritance)/Task 01 (Role of Access Modifiers)/Task V02$
```

The status bar at the bottom of the terminal window indicates:

Saving file "/home/vishu/Documents/OOPS in C++/Practical 12 (Inheritance)/Task 01 (Role of Access Modifiers)/Task V02/task-12.1.b.cpp"...

C++ Tab Width: 4 Ln 1, Col 3 INS

Task 12.2.a:**Source Code:**

```
#include <iostream>
using namespace std;
class A
{
public:
    A()
    {
        cout << "Execution of Class A Constructor" << endl;
    }
};
class B : public A
{
public:
    B()
    {
        cout << "Execution of Class B Constructor" << endl;
    }
};
class C : public B
{
public:
    C()
    {
        cout << "Execution of Class C Constructor" << endl;
    }
};
int main()
{
    cout << "Called Constructor of C" << endl
        << "....." << endl;
    C obj;
}
```

Output

The screenshot shows a terminal window titled "task_12.2.a.cpp" running on a Linux system. The terminal displays the execution flow of constructors and destructors for classes A, B, and C. Below the terminal, a docked application bar shows icons for various desktop applications like a web browser, file manager, and system tools.

```

Activities Terminal ▾ Dec 14 23:06
task_12.2.a.cpp
~/Documents/OOPS in C++/Practical 12 (Inh...Low of Constructor & Destructot )/Task V01

1 //12.2.b
2
3 #include <iostream>
4 using namespace std;
5 class A
6 {
7 public:
8     A()
9     {
10         cout << "Execution of Class A Constructor" << endl;
11     }
12 };
13 class B : public A
14 {
15
16 public:
17     B()
18     {
19         cout << "Execution of Class B Constructor" << endl;
20     }
21 };
22 class C : public B
23 {
24
25 public:
26     C()
27     {
28         cout << "Execution of Class C Constructor" << endl;
29     }
30 };
31 int main()
32 {
33     cout << "Called Constructor of C" << endl;
34     << "....." << endl;
35     C obj;
36 }

vishu@VICKY: ~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 02 ( Execution Flow of Constructor & Destructot )/Task V01$ ls
task_12.2.a.cpp
vishu@VICKY: ~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 02 ( Execution Flow of Constructor & Destructot )/Task V01$ g++ task_12.2.a.cpp
vishu@VICKY: ~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 02 ( Execution Flow of Constructor & Destructot )/Task V01$ ./a.out
Called Constructor of C
.....
Execution of Class A Constructor
Execution of Class B Constructor
Execution of Class C Constructor
vishu@VICKY: ~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 02 ( Execution Flow of Constructor & Destructot )/Task V01$ 

```

Task 12.2.b:**Source Code:**

```
#include <iostream>
using namespace std;
class A
{
public:
    ~A()
    {
        cout << "Execution of Class A Destructor" << endl;
    }
};
class B : public A
{
public:
    ~B()
    {
        cout << "Execution of Class B Destructor" << endl;
    }
};
class C : public B
{
public:
    ~C()
    {
        cout << "Execution of Class C Destructor" << endl;
    }
};
int main()
{
    C obj;
    cout << "calling of Class A Destructor" << endl;
    obj.~A();
    cout << endl << "calling of Class B Destructor" <<
endl;
    obj.~B();
    cout << endl << "calling of Class C Destructor" <<
endl;
    obj.~C();
    cout << endl << "Default Destructor called as
Program Ends" << endl;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (gedit task_12.2.b.cpp):

```

1 //12.2.b
2
3 #include <iostream>
4 using namespace std;
5 class A
6 {
7 public:
8     ~A()
9     {
10         cout << "Execution of Class A Destructor" << endl;
11     }
12 };
13 class B : public A
14 {
15
16 public:
17     ~B()
18     {
19         cout << "Execution of Class B Destructor" << endl;
20     }
21 };
22 class C : public B
23 {
24 public:
25     ~C()
26     {
27         cout << "Execution of Class C Destructor" << endl;
28     }
29 };
30 int main()
31 {
32     C obj;
33     cout << "calling of Class A Destructor" << endl;
34     obj.~A();
35     cout << endl << "calling of Class B Destructor" << endl;
36     obj.~B();
37     cout << endl << "calling of Class C Destructor" << endl;
38     obj.~C();
39     cout << endl << "Default Destructor called as Program Ends" << endl;
40 }

```

Terminal (gedit task_12.2.b.cpp):

```

^Z
[1]+  Stopped                  gedit task_12.2.b.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 02 ( Execution Flow of Constructor & Destructot )/Task V02$ g++ task_12.2.b.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 02 ( Execution Flow of Constructor & Destructot )/Task V02$ ./a.out
calling of Class A Destructor
Execution of Class A Destructor
calling of Class B Destructor
Execution of Class B Destructor
Execution of Class A Destructor
calling of Class C Destructor
Execution of Class C Destructor
Execution of Class B Destructor
Execution of Class A Destructor
Default Destructor called as Program Ends
Execution of Class C Destructor
Execution of Class B Destructor
Execution of Class A Destructor
vishu@VICKY:~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 02 ( Execution Flow of Constructor & Destructot )/Task V02$ 

```

Task 12.2.c:
Source Code:

```
#include <iostream>
using namespace std;
class A
{
public:
    int a;
    A(int k)
    {
        a = k;
        cout << "a =" << a << endl;
    }
};
class B : public A
{
public:
    int b;
    B(int k):A(k)
    {
        b = k;
        cout << "b =" << b << endl;
    }
};
class C : public B
{
public:
    C(int k):B(k)
    {
        c = k;
        cout << "c =" << c << endl;
    }
};
int main()
{
    C obj(16);
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //12.2.c
2
3 #include <iostream>
4 using namespace std;
5 class A
6 {
7 public:
8     int a;
9     A(int k)
10    {
11        a = k;
12        cout << "a =" << a << endl;
13    }
14 };
15 class B : public A
16 {
17 public:
18     int b;
19     B(int k):A(k)
20    {
21        b = k;
22        cout << "b =" << b << endl;
23    }
24 };
25 class C : public B
26 {
27 public:
28     int c;
29     C(int k):B(k)
30    {
31        c = k;
32        cout << "c =" << c << endl;
33    }
34 };
35 int main()
36 {
37     C obj(16);
38 }
39
40 }
```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 12 (Inheritance)/Task 02 ( Execution Flow of Constructor & Destructot )/Task V03$ gedit task_12.2.c.cpp
^Z
[1]+  Stopped                  gedit task_12.2.c.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 02 ( Execution Flow of Constructor & Destructot )/Task V03$ g++ task_12.2.c.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 02 ( Execution Flow of Constructor & Destructot )/Task V03$ ./a.out
a =16
b =16
c =16
vishu@VICKY:~/Documents/OOPS in C++/Practical 12 ( Inheritance )/Task 02 ( Execution Flow of Constructor & Destructot )/Task V03$ 
```

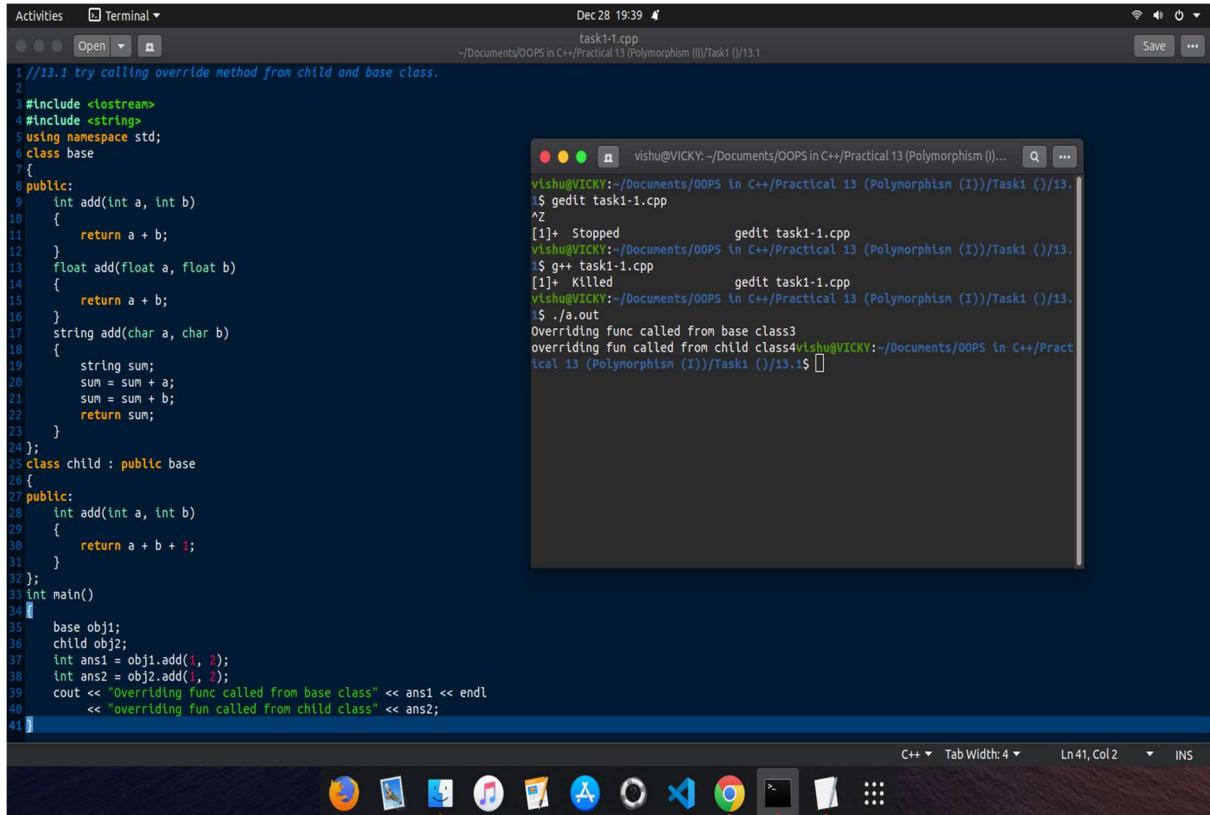
Program 13

Task 13.1.a:

Source Code:

```
#include <iostream>
#include <string>
using namespace std;
class base
{
public:
    int add(int a, int b)
    {
        return a + b;
    }
    float add(float a, float b)
    {
        return a + b;
    }
    string add(char a, char b)
    {
        string sum;
        sum = sum + a;
        sum = sum + b;
        return sum;
    }
};
class child : public base
{
public:
    int add(int a, int b)
    {
        return a + b + 1;
    }
};
int main()
{
    base obj1;
    child obj2;
    int ans1 = obj1.add(1, 2);
    int ans2 = obj2.add(1, 2);
    cout << "Overriding func called from base class" << ans1
<< endl
                << "overriding fun called from child class" << ans2;
}
```

Output



The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (gedit task1-1.cpp):

```

1 //13.I try calling override method from child and base class.
2
3 #include <iostream>
4 #include <string>
5 using namespace std;
6
7 class base
8 {
9     public:
10    int add(int a, int b)
11    {
12        return a + b;
13    }
14    float add(float a, float b)
15    {
16        return a + b;
17    }
18    string add(char a, char b)
19    {
20        string sum;
21        sum = sum + a;
22        sum = sum + b;
23        return sum;
24    }
25 }
26 class child : public base
27 {
28     public:
29     int add(int a, int b)
30     {
31         return a + b + 1;
32     }
33 }
34 int main()
35 {
36     base obj1;
37     child obj2;
38     int ans1 = obj1.add(1, 2);
39     int ans2 = obj2.add(1, 2);
40     cout << "Overriding func called from base class" << ans1 << endl
41     << "overriding fun called from child class" << ans2;
42 }
```

Terminal (vishu@VICKY: ~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task1 ()/13.1):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task1 ()/13.1$ gedit task1-1.cpp
^Z
[1]+  Stopped                  gedit task1-1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task1 ()/13.1$ g++ task1-1.cpp
[1]+  Killed                  gedit task1-1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task1 ()/13.1$ ./a.out
Overriding func called from base class3
overriding fun called from child class4
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task1 ()/13.1$ 
```

Task 13.1.2.a:**Source Code:**

```

#include <iostream>
#include <string>
using namespace std;
class base
{
public:
    int add(int a, int b)
    {
        return a + b;
    }
    float add(float a, float b)
    {
        return a + b;
    }
    string add(char a, char b)
    {
        string sum;
        sum = sum + a;
        sum = sum + b;
        return sum;
    }
};
class child : public base
{
public:
    using base::add;           //with using
    int add(int a, int b)
    {
        return a + b + 1;
    }
};
int main()
{
    child obj2;
    int ans1 = obj2.add(1, 2);
    float ans2 = obj2.add(5.5f, 2.15f);
    string ans3 = obj2.add('a', 'b');
    cout << "With using " << endl
        << ans1 << endl
        << ans2 << endl
        << ans3;
}
/*
not visible functions
with using
base-add(int ,int)
*/

```

Output

```

Activities Terminal ▾
task1-2-a.cpp
Dec 28 19:48
task1-2-a.cpp
-/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 01/Task V02/Task V02 - 1$ gedit task1-2-a.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 01/Task V02/Task V02 - 1$ gedit task1-2-a.cpp
^Z
[1]+  Stopped                  gedit task1-2-a.cpp
Vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 01/Task V02/Task V02 - 1$ g++ task1-2-a.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 01/Task V02/Task V02 - 1$ ./a.out
bash: ./a.out: No such file or directory
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 01/Task V02/Task V02 - 1$ ./a.out
With using
4
7.65
abvishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 01/Task V02/Task V02 - 1$ 

```

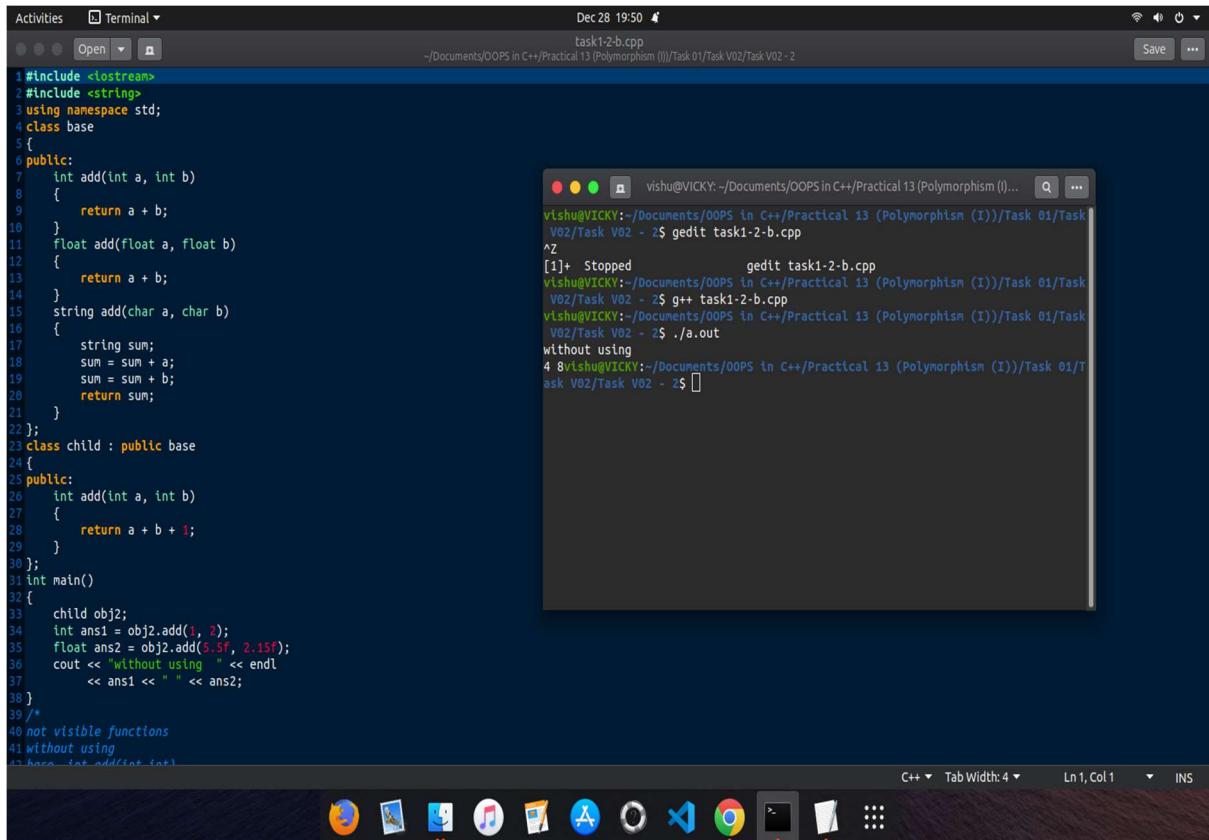
C++ Tab Width: 4 Ln 47, Col 3 INS

Task 13.1.2.b:

Source Code:

```
#include <iostream>
#include <string>
using namespace std;
class base
{
public:
    int add(int a, int b)
    {
        return a + b;
    }
    float add(float a, float b)
    {
        return a + b;
    }
    string add(char a, char b)
    {
        string sum;
        sum = sum + a;
        sum = sum + b;
        return sum;
    }
};
class child : public base
{
public:
    int add(int a, int b)
    {
        return a + b + 1;
    }
};
int main()
{
    child obj2;
    int ans1 = obj2.add(1, 2);
    float ans2 = obj2.add(5.5f, 2.15f);
    cout << "without using " << endl
        << ans1 << " " << ans2;
}
/*not visible functions
without using
base -int add(int,int)
-float add(float ,float)
-string add(char,char)*/
```

Output



The screenshot shows a Linux desktop environment with a dark theme. In the foreground, a terminal window is open with the command line interface. The terminal shows the user's session, including navigating to a directory, opening a file named 'task1-2-b.cpp' with the gedit text editor, and then running the file with g++ to produce an executable. The terminal also shows the user's name and the date and time (Dec 28 19:50). In the background, a code editor window is visible, displaying C++ code for a class hierarchy involving 'base' and 'child' classes with an 'add' method. The code editor has tabs for C++ and Tab Width: 4, and shows line 1, column 1.

```

Activities Terminal ▾ Dec 28 19:50 task1-2-b.cpp
Open ... Save ...
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class base
5 {
6 public:
7     int add(int a, int b)
8     {
9         return a + b;
10    }
11    float add(float a, float b)
12    {
13        return a + b;
14    }
15    string add(char a, char b)
16    {
17        string sum;
18        sum = sum + a;
19        sum = sum + b;
20        return sum;
21    }
22 };
23 class child : public base
24 {
25 public:
26     int add(int a, int b)
27     {
28         return a + b + 1;
29     }
30 };
31 int main()
32 {
33     child obj2;
34     int ans1 = obj2.add(1, 2);
35     float ans2 = obj2.add(5.5f, 2.15f);
36     cout << "without using " << endl
37     << ans1 << " " << ans2;
38 }
39 /*
40 not visible functions
41 without using
42 base::int add(int, int)

```

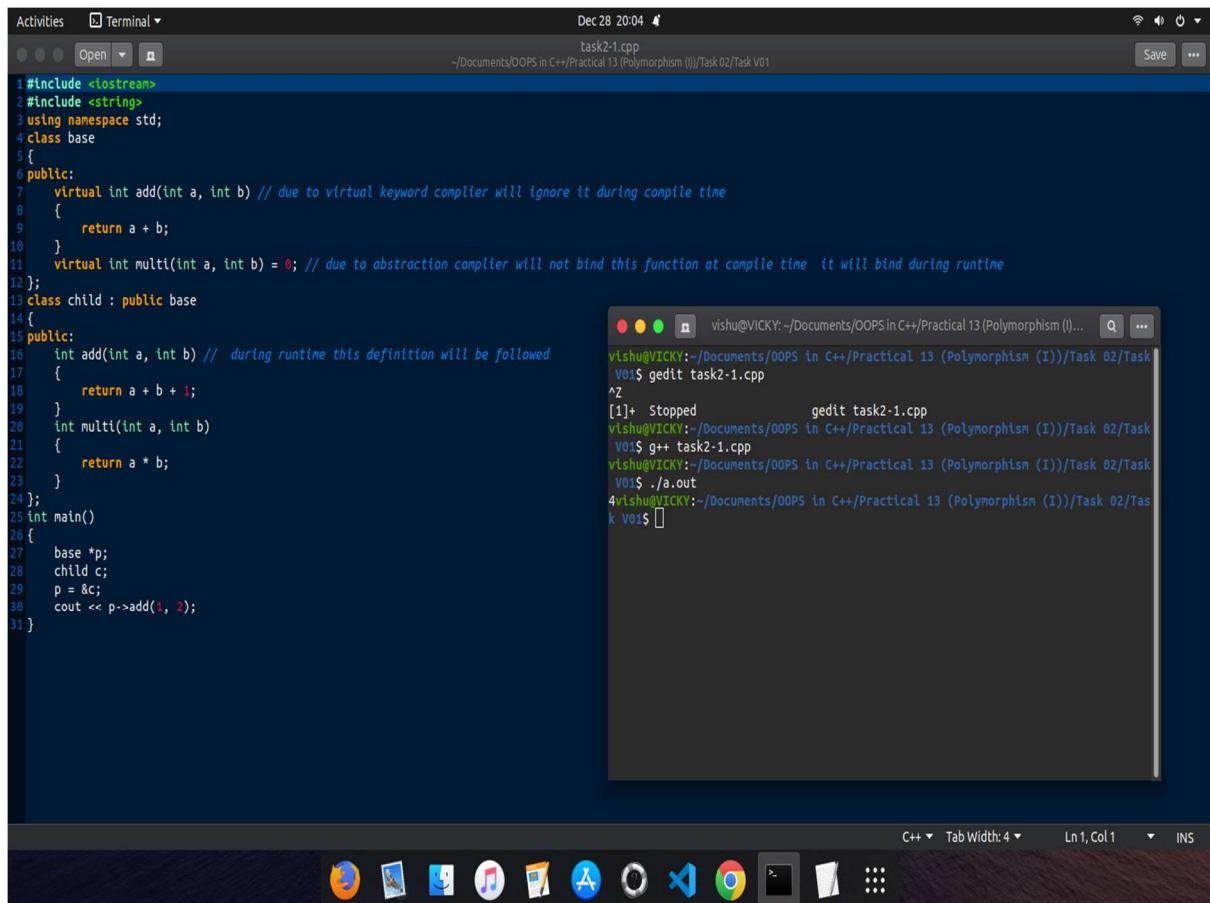
Task 13.2.1:**Source Code:**

```

#include <iostream>
#include <string>
using namespace std;
class base
{
public:
    virtual int add(int a, int b) // due to virtual
keyword complier will ignore it during compile time
    {
        return a + b;
    }
    virtual int multi(int a, int b) = 0; // due to
abstraction complier will not bind this function at
compile time it will bind during runtime
};
class child : public base
{
public:
    int add(int a, int b) // during runtime this
definition will be followed
    {
        return a + b + 1;
    }
    int multi(int a, int b)
    {
        return a * b;
    }
};
int main()
{
    base *p;
    child c;
    p = &c;
    cout << p->add(1, 2);
}

```

Output



The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (gedit) Content:

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class base
5 {
6 public:
7     virtual int add(int a, int b) // due to virtual keyword compiler will ignore it during compile time
8     {
9         return a + b;
10    }
11    virtual int multi(int a, int b) = 0; // due to abstraction compiler will not bind this function at compile time it will bind during runtime
12 };
13 class child : public base
14 {
15 public:
16     int add(int a, int b) // during runtime this definition will be followed
17     {
18         return a + b + 1;
19     }
20     int multi(int a, int b)
21     {
22         return a * b;
23     }
24 };
25 int main()
26 {
27     base *p;
28     child c;
29     p = &c;
30     cout << p->add(1, 2);
31 }

```

Terminal Window:

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 02/Task
V01$ gedit task2-1.cpp
^Z
[1]+  Stopped                  gedit task2-1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 02/Task
V01$ g++ task2-1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 02/Task
V01$ ./a.out
4
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 02/Task
k V01$ 

```

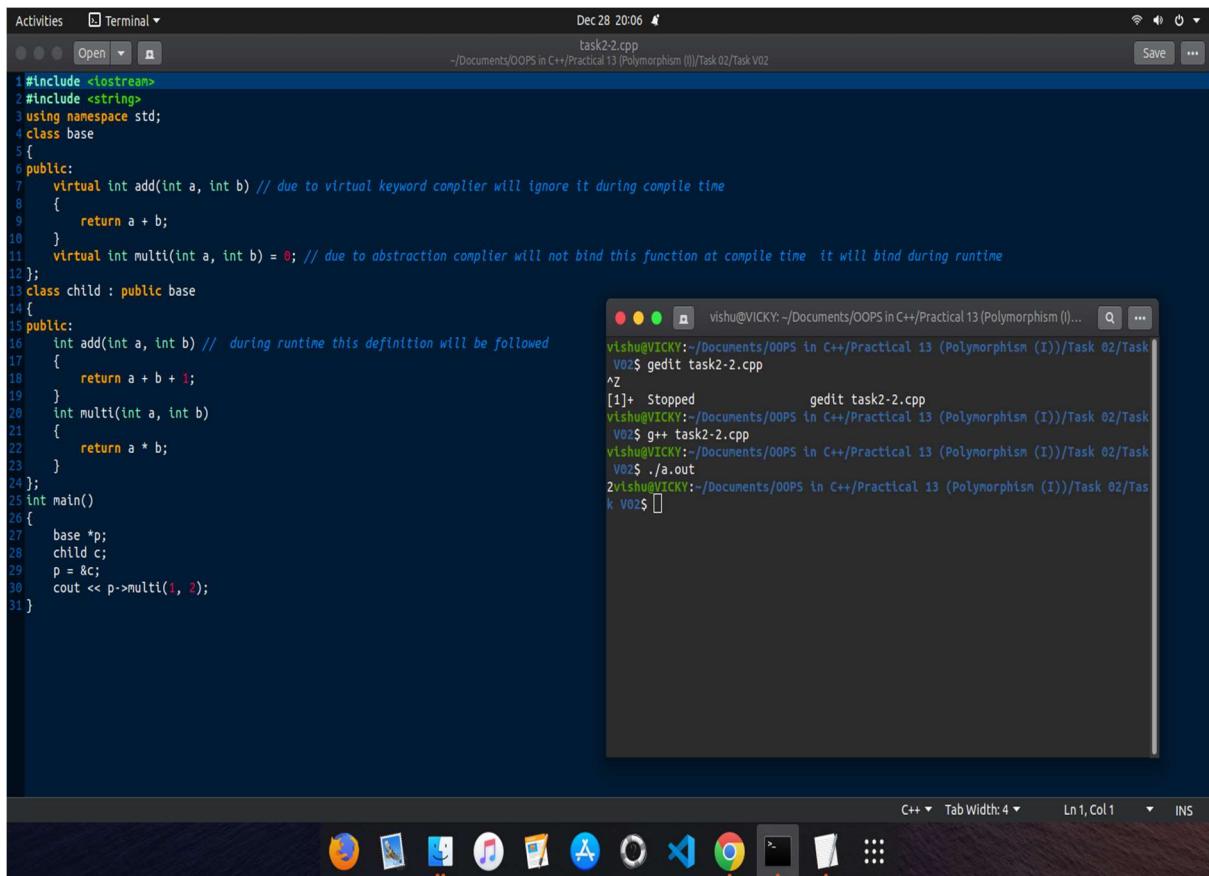
Task 13.2.2:**Source Code:**

```

#include <iostream>
#include <string>
using namespace std;
class base
{
public:
    virtual int add(int a, int b) // due to virtual
keyword complier will ignore it during compile time
    {
        return a + b;
    }
    virtual int multi(int a, int b) = 0; // due to
abstraction complier will not bind this function at
compile time it will bind during runtime
};
class child : public base
{
public:
    int add(int a, int b) // during runtime this
definition will be followed
    {
        return a + b + 1;
    }
    int multi(int a, int b)
    {
        return a * b;
    }
};
int main()
{
    base *p;
    child c;
    p = &c;
    cout << p->multi(1, 2);
}

```

Output



The screenshot shows a Linux desktop environment with a dark theme. At the top, there is a dock with various application icons. Below the dock, a terminal window is open, showing the command line history:

```
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 02/Task V02$ gedit task2-2.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 02/Task V02$ g++ task2-2.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism (I))/Task 02/Task V02$ ./a.out
2147483647
```

To the left of the terminal, a code editor window titled "task2-2.cpp" is visible, displaying the following C++ code:

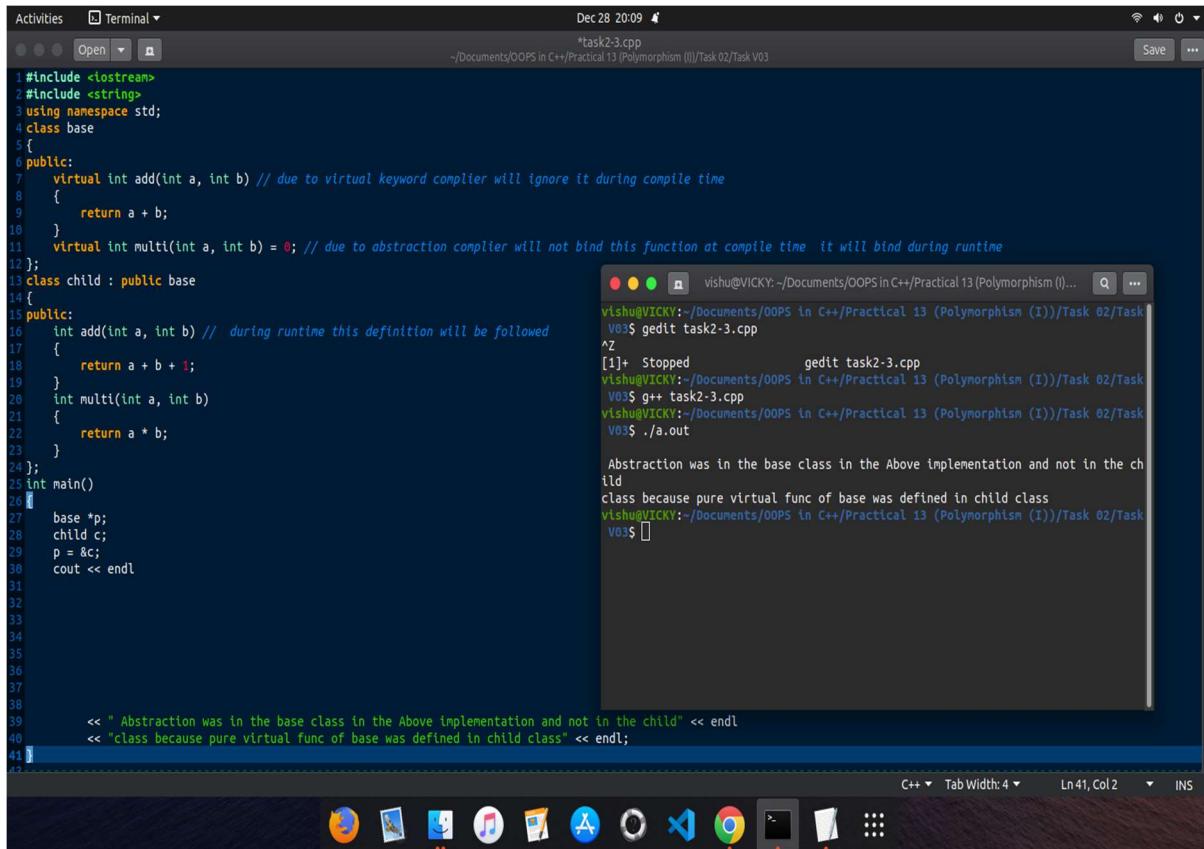
```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class base
5 {
6 public:
7     virtual int add(int a, int b) // due to virtual keyword compiler will ignore it during compile time
8     {
9         return a + b;
10    }
11    virtual int multi(int a, int b) = 0; // due to abstraction compiler will not bind this function at compile time it will bind during runtime
12 };
13 class child : public base
14 {
15 public:
16     int add(int a, int b) // during runtime this definition will be followed
17     {
18         return a + b + 1;
19     }
20     int multi(int a, int b)
21     {
22         return a * b;
23     }
24 };
25 int main()
26 {
27     base *p;
28     child c;
29     p = &c;
30     cout << p->multi(1, 2);
31 }
```

Task 13.2.3:

Source Code:

```
#include <iostream>
#include <string>
using namespace std;
class base
{
public:
    virtual int add(int a, int b) // due to virtual keyword
complier will ignore it during compile time
    {
        return a + b;
    }
    virtual int multi(int a, int b) = 0; // due to abstraction
complier will not bind this function at compile time it will
bind during runtime
};
class child : public base
{
public:
    int add(int a, int b) // during runtime this definition
will be followed
    {
        return a + b + 1;
    }
    int multi(int a, int b)
    {
        return a * b;
    }
};
int main()
{
    base *p;
    child c;
    p = &c;
    cout << endl << " Abstraction was in the base class in the
Above implementation and not in the child" << endl
        << "class because pure virtual func of base was
defined in child class" << endl;
}
```

Output



The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class base
5 {
6 public:
7     virtual int add(int a, int b) // due to virtual keyword compiler will ignore it during compile time
8     {
9         return a + b;
10    }
11    virtual int multi(int a, int b) = 0; // due to abstraction compiler will not bind this function at compile time
12 };
13 class child : public base
14 {
15 public:
16     int add(int a, int b) // during runtime this definition will be followed
17     {
18         return a + b + 1;
19     }
20     int multi(int a, int b)
21     {
22         return a * b;
23     }
24 };
25 int main()
26 {
27     base *p;
28     child c;
29     p = &c;
30     cout << endl;
31
32
33
34
35
36
37
38
39     << " Abstraction was in the base class in the Above implementation and not in the child" << endl;
40     << "class because pure virtual func of base was defined in child class" << endl;
41 }
```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism ())$ gedit task2-3.cpp
^Z
[1]+  Stopped                  gedit task2-3.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism ())$ g++ task2-3.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism ())$ ./a.out
Abstraction was in the base class in the Above implementation and not in the ch
ild
class because pure virtual func of base was defined in child class
vishu@VICKY:~/Documents/OOPS in C++/Practical 13 (Polymorphism ())$ 
```

Program 14

Task 14.1: Source Code:

```
#include <iostream>
#include <string>
using namespace std;
class base
{
public:
    virtual int add(int a, int b)
    {
        return a + b;
    }
    int multi(int a, int b)
    {
        return a * b;
    }
};
class child : public base
{
public:
    int add(int a, int b)
    {
        return a + b + 1;
    }
    int multi(int a, int b)
    {
        return a / b;
    }
};
int main()
{
    base *p;
    child c;
    p = &c;
    cout << p->multi(3, 7) << " "; // early binding (compile
time binding)
    cout << p->add(1, 2);           // late binding (run time
binding)
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor Window:

```

Activities Terminal ▾
task1.cpp
Dec 28 20:16
-/Documents/OOPS in C++/Practical 14 ()/Task 01
Save ...
```

Terminal Window:

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 01$ gedit task1.cpp
^Z
[1]+  Stopped                  gedit task1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 01$ g++ task1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 01$ ./a.out
21 4
vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 01$
```

The terminal shows the execution of the C++ program, which outputs the value 21 4, indicating early binding at compile time.

Task 14.2.1:

Source Code:

```
#include <iostream>
#include <string>
using namespace std;
class base
{
public:
    base()
    {
        cout << "BC" << endl;
    }
    ~base()
    {
        cout << "BD" << endl;
    }
};
class child : public base
{
public:
    child()
    {
        cout << "CC" << endl;
    }
    ~child()
    {
        cout << "CD" << endl;
    }
};
int main()
{
    base *p;
    child c;
    p = &c;
    delete p;           // BC CC BD
    // p->~base();     BC CC BD CD BD
    // p->~child();   error: the type being destroyed is
    'base', but the destructor refers to 'child'
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (gedit task1.cpp):

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class base
5 {
6 public:
7     base()
8     {
9         cout << "BC" << endl;
10    }
11    ~base()
12    {
13        cout << "BD" << endl;
14    }
15 };
16 class child : public base
17 {
18 public:
19     child()
20     {
21         cout << "CC" << endl;
22     }
23     ~child()
24     {
25         cout << "CD" << endl;
26     }
27 };
28 int main()
29 {
30     base *p;
31     child c;
32     p = &c;
33     delete p;      // BC CC BD
34     // p->~base(); BC CC BD CD BD
35     // p->~child(); error: the type being destroyed is 'base', but the destructor refers to 'child'
36 }

```

Terminal (gedit task2-1.cpp):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 02/Task V01$ gedit task2-1.cpp
^Z
[1]+  Stopped                  gedit task2-1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 02/Task V01$ g++ task2-1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 02/Task V01$ ./a.out
BC
CC
BD
free(): invalid pointer
Aborted (core dumped)
vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 02/Task V01$ 

```

The terminal output shows the program's execution and its termination due to a segmentation fault (core dump) at the line where the base destructor is called.

Task 14.2.2:

Source Code:

```
#include <iostream>
#include <string>
using namespace std;
class base
{
public:
    base()
    {
        cout << "BC" << endl;
    }
    virtual ~base()
    {
        cout << "BD" << endl;
    }
};
class child : public base
{
public:
    child()
    {
        cout << "CC" << endl;
    }
    ~child()
    {
        cout << "CD" << endl;
    }
};
int main()
{
    base *p;
    child c;
    p = &c;
    delete p;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (gedit task2-2.cpp):

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4 class base
5 {
6 public:
7     base()
8     {
9         cout << "BC" << endl;
10    }
11    virtual ~base()
12    {
13        cout << "BD" << endl;
14    }
15 };
16 class child : public base
17 {
18 public:
19     child()
20     {
21         cout << "CC" << endl;
22     }
23     ~child()
24     {
25         cout << "CD" << endl;
26     }
27 };
28 int main()
29 {
30     base *p;
31     child c;
32     p = &c;
33     delete p;
34 }
```

Terminal (gedit task2-2.cpp):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 02/Task V02$ gedit task
2-2.cpp
^Z
[1]+  Stopped                  gedit task2-2.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 02/Task V02$ g++ task2-
2.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 02/Task V02$ ./a.out
BC
CC
CD
BD
free(): invalid pointer
Aborted (core dumped)
vishu@VICKY:~/Documents/OOPS in C++/Practical 14 ()/Task 02/Task V02$ 
```

Desktop Bar:

- Firefox icon
- Nautilus icon
- GNOME Terminal icon
- Music icon
- Office icon
- Activities icon
- Search bar
- Google Chrome icon
- Unity Dash icon
- File icon
- Help icon

Program 15

Task 15.1: Source Code:

```

#include<iostream>
#include<fstream>                                //header for
ifstream, ofstream, fstream classes
using namespace std;

int main()
{
    ofstream write;                                //creation
of ofstream class object for writing file

    string name;

    write.open("Sample.txt");                      //opening file for writing

    while(write)
    {
        getline(cin, name);                      //reading line from
standard input

        if(name=="-1")
            break;
        write << name << endl;                  //writing line to
file

    }
    write.close();                                 //closing
file

    ifstream read;                                //creating
ifstream class object for reading the file

    read.open("Sample.txt");                      //opening file for reading

    while(read)
    {

        getline(read, name);                     //reading line from
file
        cout << name << endl;
    }

    read.close();                                //closing file
    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a window manager like Unity. There are two terminal windows and one text editor window.

Terminal Window 1:

```

Activities   Text Editor ▾
Jan 8 13:29  15.1.cpp
Open ...     ~/Documents/OOPS in C++/Practical 15 (File Handling )/Task 01 ( read and write from and to a file using ifstream and ofstream )
Save ...
1//task 15.1
2 //read and write from and to a file using ifstream and ofstream.
3 #include<iostream>
4 #include<fstream>           //header for ifstream, ofstream, fstream classes
5 using namespace std;
6
7
8 int main()
9 {
10    ofstream write;          //creation of ofstream class object for writing file
11    string name;
12
13    write.open("Sample.txt"); //opening file for writing
14
15    while(write)
16    {
17        getline(cin,name);  //reading line from standard input
18
19        if(name== "-1" )
20            break;
21        write << name << endl; //writing line to file
22
23    }
24    write.close();           //closing file
25
26    ifstream read;          //creating ifstream class object for reading the file
27
28    read.open("Sample.txt"); //opening file for reading
29
30    while(read)
31    {
32
33        getline(read,name); //reading line from file
34        cout << name << endl;
35    }
36
37    read.close();           //closing file
38
39    return 0;
40
41 }
42

```

Terminal Window 2:

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task 01 ( read and write from and to a file using ifstream and ofstream )$ ls
15.1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task 01 ( read and write from and to a file using ifstream and ofstream )$ g++ 15.1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task 01 ( read and write from and to a file using ifstream and ofstream )$ ./a.out
hello knockcat this side
github.com/knockcat
-1
hello knockcat this side
github.com/knockcat

vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task 01 ( read and write from and to a file using ifstream and ofstream )$ 

```

Text Editor Window:

Sample.txt

```

Plain Text ▾ Tab Width: 4 ▾ Ln 1, Col 1 ▾ INS
1 hello knockcat this side
2 github.com/knockcat

```

Task 15.2.a:**Source Code:**

```

#include<iostream>
#include<string>
#include<fstream>
#include<stdlib.h>

using namespace std;

int main()
{
    fstream myfile;
    myfile.open("Sample.txt",ios::in);;

    if(!myfile)                                //if
Sample.txt doesn't exists
    {
        cout<<"\nFile Not Exist's\n";
        exit (0);                            //program exit
    }

    char data;
    while(1)
    {
        myfile>>data;                      //reading char
by char
        if(myfile.eof())                   //till we
doesn't encounter end of file
        {
            break;
        }
        else cout<<data<<"_";
    }
    myfile.close();

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with several windows open:

- Code Editor:** An Activities window titled "Text Editor" containing C++ code for reading from a file using fstream. The code includes error handling for file non-existence and character-by-character reading.
- Terminal:** A terminal window titled "15.2.a.cpp" showing the command-line interface. It lists files in the current directory, compiles the C++ code into an executable ("15.2.a"), and runs it ("./15.2.a"). The output shows the contents of "Sample.txt".
- File Browser:** A terminal window titled "Sample.txt" displaying the file's contents: "hello knockcat this side" and "github.com/knockcat".
- System Tray:** Shows icons for various applications like a browser, file manager, and system status indicators.

Task 15.2.b:**Source Code:**

```

#include<iostream>
#include<string>
#include<fstream>

using namespace std;
int main()
{
    fstream myfile;
    //creating object of fstream class
    myfile.open("myfile.txt",ios::out);;           //opening
file in write mode

    if(!myfile)
        //if file not created
    {
        cout<<"file not created"=<<endl;
    }

    else
    {
        cout<<"file is created"=<<endl;           //if file
created
        myfile<<"file is diplaying"=<<endl; //writing
to a file
    }

    myfile.close();
    //closing file
    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with the following components:

- Terminal Window:** Shows the command-line interface with the user's session details and the execution of the C++ program.
- File Browser Window:** Displays the directory structure and files related to the practical task.
- File Editor Window:** Shows the source code of the C++ program (15.2.b.cpp) which demonstrates writing to a file using fstream.

```

Activities   Text Editor ▾
Jan 8 14:15
*15.2.b.cpp
-/Documents/OOPS in C++/Practical 15 (File .../Task V02 (Write into a file using fstream)
Save ...
15.2.a.cpp

1 //15.3
2
3 //b. Write into a file using fstream
4
5 #include<iostream>
6 #include<string>
7 #include<fstream>
8
9 using namespace std;
10 int main()
11 {
12     fstream myfile;
13     myfile.open("myfile.txt",ios::out);    //creating object of fstream class
14                                         //opening file in write mode
15     if(!myfile)                         //if file not created
16     {
17         cout<<"file not created" << endl;
18     }
19     else
20     {
21         cout<<"file is created" << endl;    //if file created
22         myfile<<"file is displaying" << endl; //writing to a file
23     }
24
25     myfile.close();                     //closing file
26
27     return 0;
28 }
29

```

Task 15.2.c:
Source Code:

```
#include<iostream>
#include<string>
#include<fstream>

using namespace std;

int main()
{
    fstream myfile;
    myfile.open("Sample.txt",ios::app);

    if(!myfile)
    {
        cout<<"file not created"<<endl;
    }

    else
    {
        myfile<<"file is displaying"<<endl;
        cout<<"Data Appended Succesfully"<<endl;
    }

    myfile.close();
    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Terminal Window:

```

Activities Terminal Jan 8 19:08
Open ... Save ...
1 //15.3.c
2
3 //c. Append into a file using fstream
4
5 #include<iostream>
6 #include<string>
7 #include<fstream>
8
9 using namespace std;
10
11 int main()
12 {
13     fstream myfile;
14     myfile.open("Sample.txt",ios::app);      //opening file in append mode
15
16     if(!myfile)                         //IF FILE NOT EXIST'S
17     {
18         cout<<"file not created"<<endl;
19     }
20
21     else
22     {
23         myfile<<"file is displaying"<<endl;    //Writing to file
24         cout<<"Data Appended Successfully"<<endl;
25     }
26
27     myfile.close();
28     return 0;
29 }
30

```

Code Editor Window:

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V03 ( Append into a file using fstream )
$ ls
15.2.cpp Sample.txt
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V03 ( Append into a file using fstream
$ cat Sample.txt
hello knockcat this side
github.com/knockcat

vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V03 ( Append into a file using fstream
$ mv 15.2.cpp 15.2.c.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V03 ( Append into a file using fstream
$ g++ 15.2.c.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V03 ( Append into a file using fstream
$ ./a.out
Data Appended Successfully
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V03 ( Append into a file using fstream
$ cat Sample.txt

```

File Content in Code Editor:

```

1 hello knockcat this side
2 github.com/knockcat
3
4 file is displaying

```

Task 15.2.d:

Source Code:

```
//d. Count total number of characters, words and lines in a file

#include<iostream>
#include<fstream>
#include<string>
using namespace std;
int main()
{
    fstream readfile;
    readfile.open("file.txt",ios::in);
    string s;//will help to count no of lines
    char ch;
    int count_l=0;
    int count_w=0;
    int count_c=0;
    while(getline(readfile,s))
    {
        count_l++;
    }
    cout<<"total no of lines in
file.txt==>"<<count_l<<endl;//total no of lines in file.txt

    readfile.close();
    readfile.open("file.txt",ios::in);
    while(getline(readfile,s,' '))
    {
        count_w++;
    }
    cout<<"total no of words in
file.txt==>"<<count_w+1<<endl;//total no of words in file.txt
    readfile.close();
    readfile.open("file.txt",ios::in);
    while(1)
    {
        readfile>>ch;
        if(ch==' ')
            continue;
        if(readfile.eof())
            break;
        else
            count_c++;
    }
    cout<<"total no of words in file.txt==>"<<count_c<<endl;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Terminal Window:

```
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 (File Handling) $ gedit file.txt
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 (File Handling) $ ls
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 (File Handling) $ g++ 15.2.d.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 (File Handling) $ ./a.out
total no of lines in file.txt==>20
total no of words in file.txt==>162
total no of words in file.txt==>808
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 (File Handling) $ ls
```

Code Editor (Left):

```
1 //d. Count total number of characters, words and lines in a file
2
3 #include<iostream>
4 #include<fstream>
5 #include<string>
6 using namespace std;
7 int main()
8 {
9     fstream readfile;
10    readfile.open("file.txt",ios::in);
11    string s;//will help to count no of lines
12    char ch;
13    int count_l=0;
14    int count_w=0;
15    int count_c=0;
16    while(getline(readfile,s))
17    {
18        count_l++;
19    }
20    cout<<"total no of lines in file.txt==>"<<count_l<<endl;//total no of lines in file.txt
21    readfile.close();
22    readfile.open("file.txt",ios::in);
23    while(getline(readfile,s, ' '))
24    {
25        count_w++;
26    }
27    cout<<"total no of words in file.txt==>"<<count_w+1<<endl;//total no of words in file.txt
28    readfile.close();
29    readfile.open("file.txt",ios::in);
30    while(1)
31    {
32        readfile>>ch;
33        if(ch==' ')
34            continue;
35        if(readfile.eof())
36            break;
37        else
38            count_c++;
39    }
40    cout<<"total no of words in file.txt==>"<<count_c<<endl;
```

Code Editor (Right):

```
file.txt
1 Believe you can and you are already halfway there.
2 Never Stop Learning Because Life Never Stops Teaching.
3 A Journey of 1000 miles begins with one step.
4 Be Brave and go on an adventure.
5 The Secret of getting ahead is getting started
6 the 5 second rule the moment you get an instinct to act on a goal you must 5-4-3-2-1 and
   physically move or your brain will stop you.
7 Little things make big days.
8 Imagine with all your mind believe with all your heart achieve with all your might.
9 Never never ever giveup.
10 all things are difficult before they are done.
```

Task 15.3.a:**Source Code:**

```
//a. IOS: hex,dec,skipws,noskipws

#include<iostream>
#include<iomanip>
#include<sstream>
#include<string>
using namespace std ;

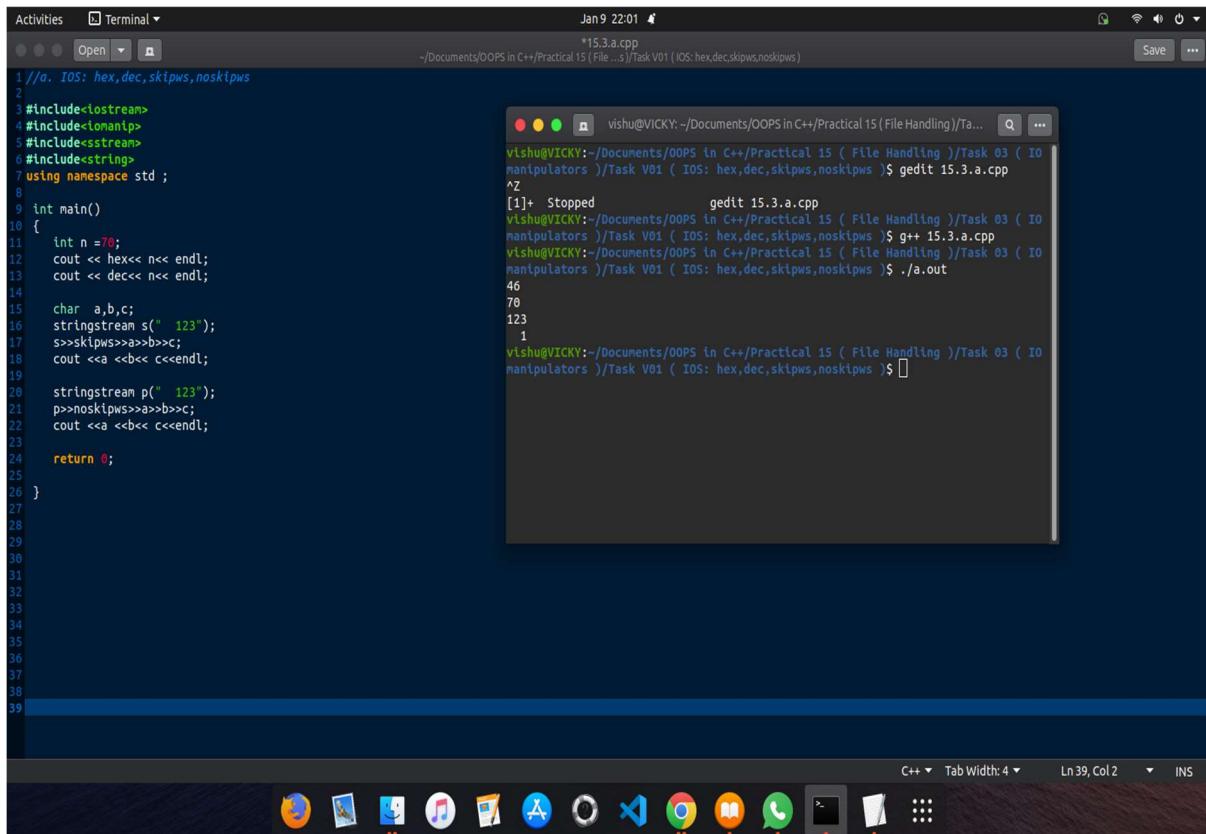
int main()
{
    int n =70;
    cout << hex<< n<< endl;
    cout << dec<< n<< endl;

    char a,b,c;
    stringstream s(" 123");
    s>>skipws>>a>>b>>c;
    cout <<a <<b<< c<<endl;

    stringstream p(" 123");
    p>>noskipws>>a>>b>>c;
    cout <<a <<b<< c<<endl;

    return 0;
}
```

Output



The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor Window:

```

1 //a. IOS: hex,dec,skipws,noskipws
2
3 #include<iostream>
4 #include<iomanip>
5 #include<sstream>
6 #include<string>
7 using namespace std ;
8
9 int main()
10 {
11     int n =70;
12     cout << hex<< n<< endl;
13     cout << dec<< n<< endl;
14
15     char a,b,c;
16     stringstream s(" 123");
17     s>>skipws>>a>>b>>c;
18     cout <<a <<b<< c<<endl;
19
20     stringstream p(" 123");
21     p>>noskipws>>a>>b>>c;
22     cout <<a <<b<< c<<endl;
23
24     return 0;
25 }
26
27
28
29
30
31
32
33
34
35
36
37
38
39

```

Terminal Window:

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V01 ( IOS: hex,dec,skipws,noskipws )$ gedit 15.3.a.cpp
[1]+  Stopped                  gedit 15.3.a.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V01 ( IOS: hex,dec,skipws,noskipws )$ g++ 15.3.a.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V01 ( IOS: hex,dec,skipws,noskipws )$ ./a.out
46
70
123
1

```

Task 15.3.b:**Source Code:**

```
//b. Istream: ws

#include<iostream>
#include<iomanip>
#include<sstream>
#include<string>
using namespace std ;

int main()
{
    stringstream t(" this is a string");
    string line;
    getline(t >> ws, line);
    cout << line << endl;

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon. Below it, a terminal window is open with the title "Activities Terminal". The terminal shows the following session:

```

Jan 9 22:07 *15.3.b.cpp
~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V02 ( Istream: ws)
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V02 ( I...
manipulators )/Task V02 ( Istream: ws)$ ls
15.3.b.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V02 ( IO
manipulators )/Task V02 ( Istream: ws)$ g++ 15.3.b.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V02 ( IO
manipulators )/Task V02 ( Istream: ws)$ ./a.out
this is a string
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task V02 ( IO
manipulators )/Task V02 ( Istream: ws)$

```

The file editor window below the terminal shows the code for "15.3.b.cpp". The code is as follows:

```

1 //b. Istream: ws
2
3 #include<iostream>
4 #include<iomanip>
5 #include<sstream>
6 #include<string>
7 using namespace std ;
8
9 int main()
10 {
11
12     stringstream t(" this is a string");
13     string line;
14     getline(t >> ws,line);
15     cout << line<<endl;
16
17     return 0;
18 }
19
20
21
22
23
24
25
26
27
28
29
30
31
32

```

The status bar at the bottom of the screen indicates "C++ Tab Width: 4" and "Ln 32, Col 2".

Task 15.3.d:
Source Code:

```
//d. Iomanip: setW, setPrecision

#include<iomanip>
#include <iostream>
using namespace std;

int main()
{
    int num = 45;
    cout << "it is: " << fixed << setprecision(2) <<
num << " the end" << endl;           //setprecision

    int i = 18;
    cout << setw(10) << i;             //setw

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Code Editor (Left):

```

1 //d. Iomanip: setW, setPrecision
2
3 #include<iomanip>
4 #include <iostream>
5 using namespace std;
6
7 int main()
8 {
9     int num = 45;
10    cout << "it is: " << fixed << setprecision(2) << num << " the end" << endl;      //setprecision
11
12    int i = 18;
13    cout << setw(10) << i;           //setw
14
15    return 0;
16 }
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task 03 ( IO manipulators )/Task V04 ( Iomanip: setW, setPrecision)$ ls
15.3.d.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task 03 ( IO manipulators )/Task V04 ( Iomanip: setW, setPrecision)$ g++ 15.3.d.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task 03 ( IO manipulators )/Task V04 ( Iomanip: setW, setPrecision)$ ./a.out
it is: 45 the end
18vishu@VICKY:~/Documents/OOPS in C++/Practical 15 ( File Handling )/Task 03 ( IO manipulators )/Task V04 ( Iomanip: setW, setPrecision)$ 

```

Program 16

Task 16.1.a:

Source Code:

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

template <typename A, typename B, typename R>
R add(A num1, B num2)
{
    R ans = num1 + num2;
    return ans;
}

int main()
{
    cout << add<int, int, int>(2, 3) << endl;
    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open in the top right corner, titled 'vishu@VICKY: ~/Documents/OOPS in C++/Practical 16 (Templated & Exception Handling)/Task 01 (create a generic add function for given tasks)/Task V01 (Perform add over two integers and return integer) \$'. The terminal shows the following session:

```

1 //16.1.a
2
3 //a. Perform add over two integers and return integer
4
5 #include<iostream>
6 using namespace std;
7
8 template <typename A, typename B, typename R>
9 R add(A num1, B num2)
10 {
11     R ans = num1 + num2;
12     return ans;
13 }
14
15 int main()
16 {
17     cout << add<int, int, int>(2, 3) << endl;
18     return 0;
19 }
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

```

The code editor window, titled 'gedit 16.1.a.cpp', is visible in the background, showing the same C++ code. The desktop dock at the bottom has icons for various applications like a browser, file manager, and system tools.

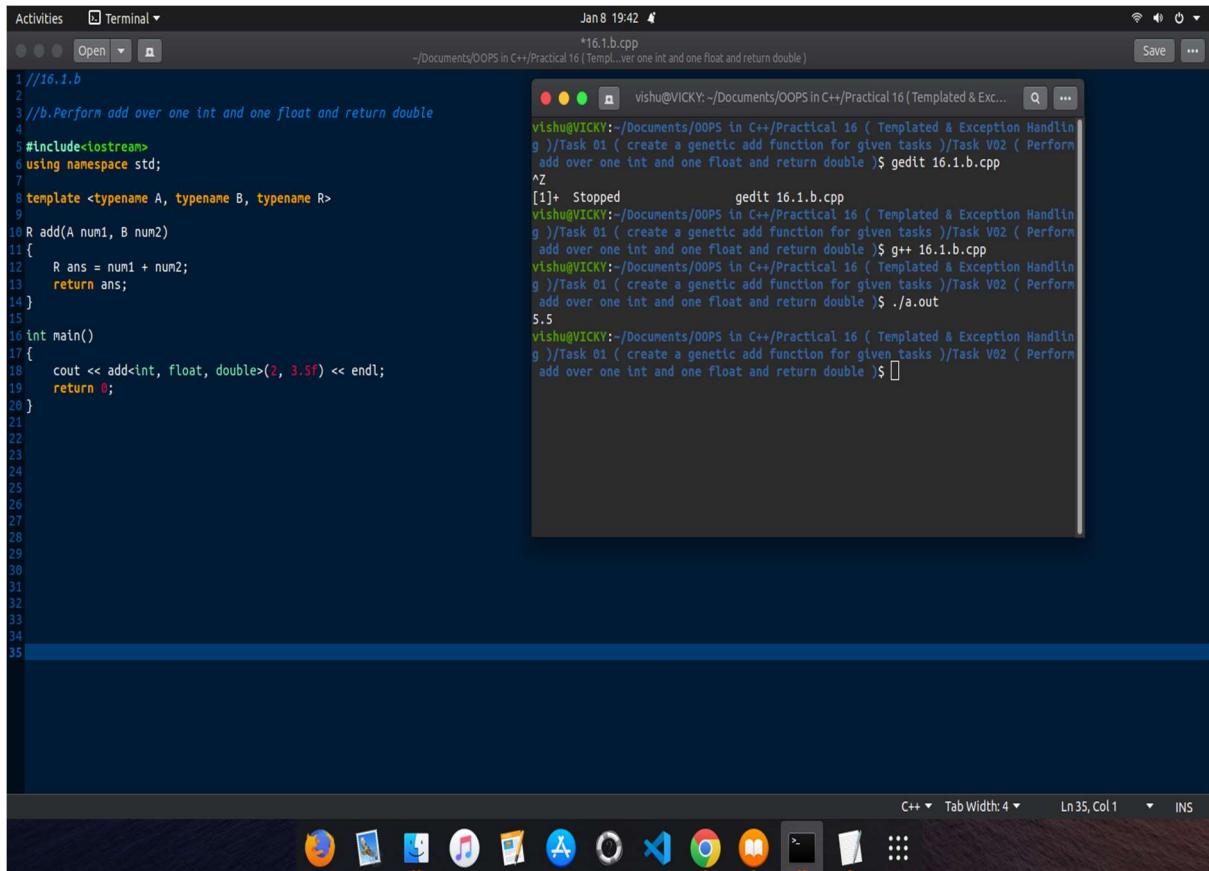
Task 16.1.b:**Source Code:**

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

template <typename A, typename B, typename R>
R add(A num1, B num2)
{
    R ans = num1 + num2;
    return ans;
}

int main()
{
    cout << add<int, float, double>(2, 3.5f) <<
endl;
    return 0;
}
```

Output



The screenshot shows a Linux desktop environment with a dark blue theme. In the top left, there's an 'Activities' button and a 'Terminal' button. The terminal window is open and displays the following text:

```

Activities Terminal ▾ Jan 8 19:42
Open ... vishu@VICKY:~/Documents/OOPS in C++/Practical 16 (Templated & Exception Handlin
1 //16.1.b
2
3 //b.Perform add over one int and one float and return double
4
5 #include<iostream>
6 using namespace std;
7
8 template <typename A, typename B, typename R>
9
10 R add(A num1, B num2)
11 {
12     R ans = num1 + num2;
13     return ans;
14 }
15
16 int main()
17 {
18     cout << add<int, float, double>(2, 3.5f) << endl;
19     return 0;
20 }
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

```

The terminal also shows the command history and the output of the program execution:

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 16 (Templated & Exception Handlin
g )/Task 01 ( create a genetic add function for given tasks )/Task V02 ( Perform
add over one int and one float and return double )$ gedit 16.1.b.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templat...
^Z
[1]+  Stopped                  gedit 16.1.b.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templat...
g )/Task 01 ( create a genetic add function for given tasks )/Task V02 ( Perform
add over one int and one float and return double )$ g++ 16.1.b.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templat...
g )/Task 01 ( create a genetic add function for given tasks )/Task V02 ( Perform
add over one int and one float and return double )$ ./a.out
5.5
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templat...
g )/Task 01 ( create a genetic add function for given tasks )/Task V02 ( Perform
add over one int and one float and return double )$ 

```

The desktop bar at the bottom contains icons for various applications like a browser, file manager, and system tools.

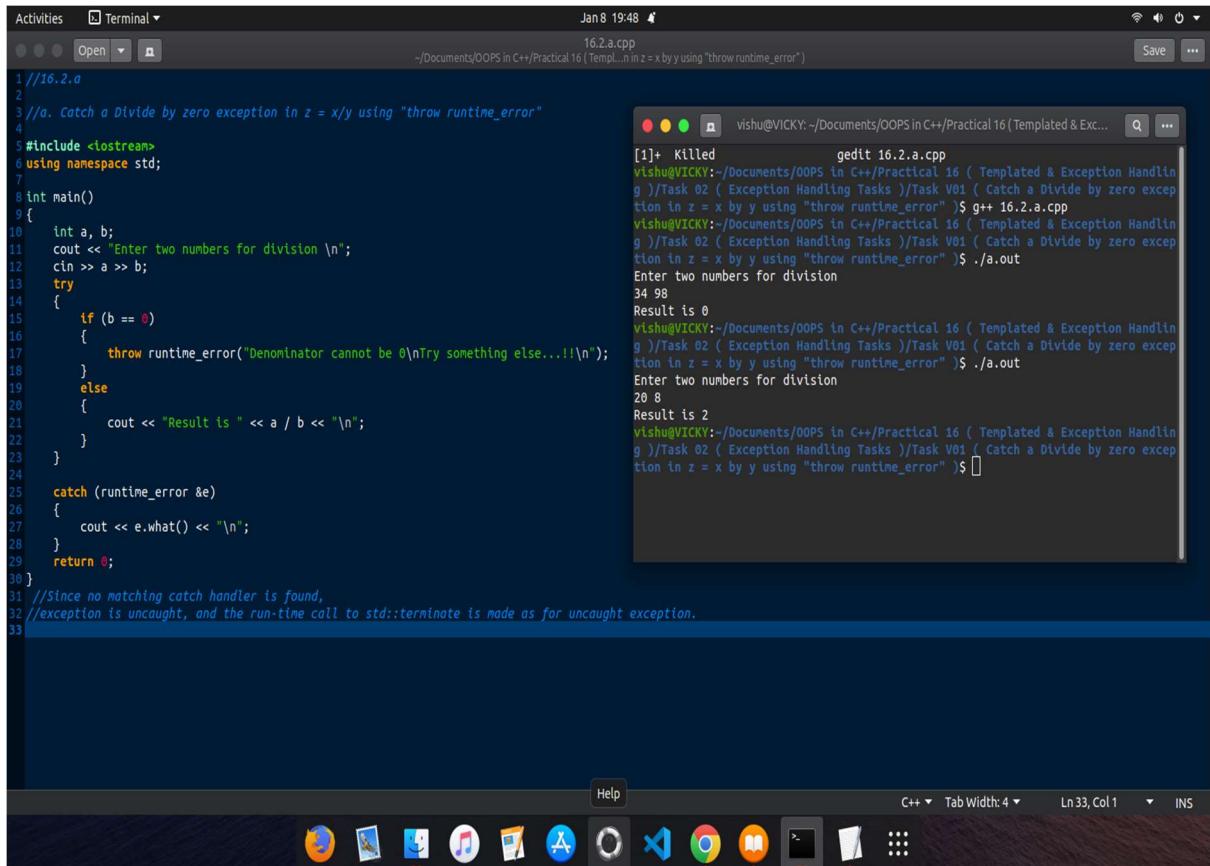
Task 16.2.a:**Source Code:**

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

int main()
{
    int a, b;
    cout << "Enter two numbers for division \n";
    cin >> a >> b;
    try
    {
        if (b == 0)
        {
            throw runtime_error("Denominator cannot
be 0\nTry something else...!!\n");
        }
        else
        {
            cout << "Result is " << a / b << "\n";
        }
    }

    catch (runtime_error &e)
    {
        cout << e.what() << "\n";
    }
    return 0;
}
//Since no matching catch handler is found,
//exception is uncaught, and the run-time call to
std::terminate is made as for uncaught exception.
```

Output



```

Activities Terminal Jan 8 19:48
Open Save ...
1 //16.2.a
2
3 //a. Catch a Divide by zero exception in z = x/y using "throw runtime_error"
4
5 #include <iostream>
6 using namespace std;
7
8 int main()
9 {
10    int a, b;
11    cout << "Enter two numbers for division \n";
12    cin >> a >> b;
13    try
14    {
15        if (b == 0)
16        {
17            throw runtime_error("Denominator cannot be 0\nTry something else...!!\n");
18        }
19        else
20        {
21            cout << "Result is " << a / b << "\n";
22        }
23    }
24
25    catch (runtime_error &e)
26    {
27        cout << e.what() << "\n";
28    }
29    return 0;
30 }
31 //Since no matching catch handler is found,
32 //exception is uncaught, and the run-time call to std::terminate is made as for uncaught exception.
33

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 16 (Templated & Exception Handling)/Task 02 (Exception Handling Tasks)/Task V01 (Catch a Divide by zero exception in z = x by y using "throw runtime_error")\$ g++ 16.2.a.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 (Templated & Exception Handling)/Task 02 (Exception Handling Tasks)/Task V01 (Catch a Divide by zero exception in z = x by y using "throw runtime_error")\$./a.out
[1]+ Killed gedit 16.2.a.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 (Templated & Exception Handling)/Task 02 (Exception Handling Tasks)/Task V01 (Catch a Divide by zero exception in z = x by y using "throw runtime_error")\$ g++ 16.2.a.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 (Templated & Exception Handling)/Task 02 (Exception Handling Tasks)/Task V01 (Catch a Divide by zero exception in z = x by y using "throw runtime_error")\$./a.out
Enter two numbers for division
34 98
Result is 0
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 (Templated & Exception Handling)/Task 02 (Exception Handling Tasks)/Task V01 (Catch a Divide by zero exception in z = x by y using "throw runtime_error")\$./a.out
Enter two numbers for division
20 8
Result is 2
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 (Templated & Exception Handling)/Task 02 (Exception Handling Tasks)/Task V01 (Catch a Divide by zero exception in z = x by y using "throw runtime_error")\$

Task 16.2.b:**Source Code:**

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

int main()
{
    try
    {
        throw 'a';
    }
    catch (int x)
    {
        cout << "caught" << x << "\n";
    }
    // Here this Catch will be executed
    catch (...)
    {
        cout << "Default Exception\n";
    }
    return 0;
}

// The block catch(...) is used for catch all, when
// a data type of a thrown exception doesn't match with
// any other catch block, the code inside catch(...) is
// executed.
// Note that the implicit type conversion doesn't
// happen when exceptions are caught. The character 'a'
// is not automatically converted to int.
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (gedit) Content:

```

1 //16.2.b
2
3 //output of this program
4
5 #include <iostream>
6 using namespace std;
7
8 int main()
9 {
10     try
11     {
12         throw 'a';
13     }
14     catch (int x)
15     {
16         cout << "caught" << x << "\n";
17     }
18     // Here this Catch will be executed
19     catch (...)
20     {
21         cout << "Default Exception\n";
22     }
23     return 0;
24 }
25
26
27
28
29 // The block catch(...) is used for catch all, when a data type of a thrown exception doesn't match with any other catch block, the code inside catch(...) is executed.
30 //Note that the implicit type conversion doesn't happen when exceptions are caught. The character 'a' is not automatically converted to int.
31
32

```

Terminal Output:

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handling )/Task 02 ( Exception Handling Tasks )/Task V02 ( output of the program )$ gedit 16.2.b.cpp
[1]+  Stopped                  gedit 16.2.b.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handling )/Task 02 ( Exception Handling Tasks )/Task V02 ( output of the program )$ gedit 16.2.b.cpp
[1]+  Stopped                  gedit 16.2.b.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handling )/Task 02 ( Exception Handling Tasks )/Task V02 ( output of the program )$ .a.out
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handling )/Task 02 ( Exception Handling Tasks )/Task V02 ( output of the program )$ vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handling )/Task 02 ( Exception Handling Tasks )/Task V02 ( output of the program )$ 

```

The terminal shows the command to open the file, the process being stopped, and then the execution of the program which outputs "caught" followed by the character 'a'.

Task 16.2.c:**Source Code:**

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

int main()
{
    try
    {
        throw 'a';
    }
    catch (int x)
    {
        cout << "caught" << x << "\n";
    }
    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there is an 'Activities' button, a 'Terminal' icon, and several other icons. The terminal window is open and displays the following text:

```
Activities Terminal ▾
Jan 9 17:31 4
16.2.c.cpp
~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handling )/Task 02 ( Exception Handling Tasks )/Task V03 ( output of the program )
Save ...
```

The terminal window shows the command line and its output:

```
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handling )/Task 02 ( Exception Handling Tasks )/Task V03 ( output of the program )$ ls
16.2.c.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handling )/Task 02 ( Exception Handling Tasks )/Task V03 ( output of the program )$ g++
16.2.c.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handling )/Task 02 ( Exception Handling Tasks )/Task V03 ( output of the program )$ ./a.out
terminate called after throwing an instance of 'char'
Aborted (core dumped)
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handling )/Task 02 ( Exception Handling Tasks )/Task V03 ( output of the program )$
```

Below the terminal, the desktop environment's dock shows various application icons, including a browser, file manager, and system tools. The status bar at the bottom indicates 'C++ Tab Width: 4' and 'Ln 1, Col 1'.

Task 16.2.d:**Source Code:**

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

void divide(int a, int b)
{
    if (b == 0)
    {
        throw runtime_error("Denominator cannot be
0\nTry something else...!!\n");
    }
    else
    {
        cout << "Result is " << a / b << "\n";
    }
}

int main()
{
    int a, b;
    cout << "Enter two numbers for division \n";
    cin >> a >> b;
    try
    {
        divide(a, b);
    }
    catch (runtime_error &e)
    {
        cout << e.what() << "\n";
    }
    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //16.2.d
2
3
4 /*d. Rethrow and catch an exception by creating a separate
5 user defined divide function for condition divide by
6 zero.*/
7
8 #include <iostream>
9 using namespace std;
10
11 void divide(int a, int b)
12 {
13     if (b == 0)
14     {
15         throw runtime_error("Denominator cannot be 0\nTry something else...!!\n");
16     }
17     else
18     {
19         cout << "Result is " << a / b << "\n";
20     }
21 }
22
23 int main()
24 {
25     int a, b;
26     cout << "Enter two numbers for division \n";
27     cin >> a >> b;
28     try
29     {
30         divide(a, b);
31     }
32     catch (runtime_error &e)
33     {
34         cout << e.what() << "\n";
35     }
36     return 0;
37 }
38
39

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handlin
g )/Task 02 ( Exception Handling Tasks )/Task V04 ( Rethrow and catch an excepti
on by creating a separate user defined divide function for condition divide by z
ero )$ ls
16.2.d.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handlin
g )/Task 02 ( Exception Handling Tasks )/Task V04 ( Rethrow and catch an excepti
on by creating a separate user defined divide function for condition divide by z
ero )$ g++ 16.2.d.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handlin
g )/Task 02 ( Exception Handling Tasks )/Task V04 ( Rethrow and catch an excepti
on by creating a separate user defined divide function for condition divide by z
ero )$ ./a.out
Enter two numbers for division
7 3
Result is 2
vishu@VICKY:~/Documents/OOPS in C++/Practical 16 ( Templated & Exception Handlin
g )/Task 02 ( Exception Handling Tasks )/Task V04 ( Rethrow and catch an excepti
on by creating a separate user defined divide function for condition divide by z
ero )$ 

```

The terminal shows the execution of the program, which prompts for two integers (7 and 3), performs the division, and outputs the result (2). It also shows the command-line interface with file navigation and compilation information.

Program 17

Task 17.1.a.1:

Source Code:

```
#include <iostream>
#include <list>
#include <iterator>
using namespace std;

// function for printing the elements in a list
void printList(list<int> g)
{
    list<int>::iterator it;
    for (it = g.begin(); it != g.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}
int main()
{
    list<int> l;
    int n;
    cout << "Enter the size of the List \n";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        int ele;
        cin >> ele;
        l.push_back(ele);
    }
    printList(l);

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Terminal Window:

```

Activities Terminal Jan 9 17:43
Open A
Jan 9 17:43
vishu@VICKY: ~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V01 ( Iterate a int list using iterator and print it )$ ls
17.1.a.1.cpp  a.out
[1]+  Killed                  gedit 17.1.a.1.cpp
vishu@VICKY: ~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V01 ( Iterate a int list using iterator and print it )$ g++ 17.1.a.1.cpp
vishu@VICKY: ~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V01 ( Iterate a int list using iterator and print it )$ ./a.out
Enter the size of the List
5
9 4 2 7 1
9 4 2 7 1
vishu@VICKY: ~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V01 ( Iterate a int list using iterator and print it )$ 

```

Code Editor:

```

1 //1. Iterate a int list using iterator and print it
2
3 #include <iostream>
4 #include <list>
5 #include <iterator>
6 using namespace std;
7
8 // function for printing the elements in a list
9 void printList(list<int> g)
10 {
11     list<int>::iterator it;
12     for (it = g.begin(); it != g.end(); ++it)
13         cout << *it << " ";
14     cout << '\n';
15 }
16 int main()
17 {
18     list<int> l;
19     int n;
20     cout << "Enter the size of the List \n";
21     cin >> n;
22     for (int i = 0; i < n; i++)
23     {
24         int ele;
25         cin >> ele;
26         l.push_back(ele);
27     }
28     printList(l);
29
30     return 0;
31 }
32
33

```

Task 17.1.a.2:**Source Code:**

```

#include <iostream>
#include <list>
#include <iterator>
using namespace std;

// function for printing the elements in a list
void printList(list<int> g)
{
    list<int>::iterator it;
    for (it = g.begin(); it != g.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}
int main()
{
    list<int> l;
    int n;
    cout << "Enter the size of the List \n";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        int ele;
        cin >> ele;
        l.push_back(ele);
    }
    printList(l);

    cout << "Size of the list is : " << l.size() <<
"\n";

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //2. Find size of a list
2
3 #include <iostream>
4 #include <list>
5 #include <iterator>
6 using namespace std;
7
8 // function for printing the elements in a list
9 void printList(list<int> g)
10 {
11     list<int>::iterator it;
12     for (it = g.begin(); it != g.end(); ++it)
13         cout << *it << " ";
14     cout << '\n';
15 }
16 int main()
17 {
18     list<int> l;
19     int n;
20     cout << "Enter the size of the List \n";
21     cin >> n;
22     for (int i = 0; i < n; i++)
23     {
24         int ele;
25         cin >> ele;
26         l.push_back(ele);
27     }
28     printList(l);
29
30     cout << "Size of the list is : " << l.size() << "\n";
31
32     return 0;
33 }
34
35
36

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V02 ( Find size of a list )$ ls
17.1.a.2.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V02 ( Find size of a list )$ g++ 17.1.a.2.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V02 ( Find size of a list )$ ./a.out
Enter the size of the List
9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
Size of the list is : 9
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V02 ( Find size of a list )$ 

```

The terminal output shows the execution of the program. It prompts for the size of the list, takes input, prints the elements, and then prints the size of the list.

Task 17.1.a.3:**Source Code:**

```

#include <iostream>
#include <list>
#include <iterator>
using namespace std;

// function for printing the elements in a list
void printList(list<int> g)
{
    list<int>::iterator it;
    for (it = g.begin(); it != g.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}
int main()
{
    list<int> l;
    int n;
    cout << "Enter the size of the List \n";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        int ele;
        cin >> ele;
        l.push_back(ele);
    }
    printList(l);

    l.sort();

    cout << "Printing list after sorting\n";

    printList(l);

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Terminal Window:

```

Activities Terminal ▾ Jan 9 17:47 *17.1.a.3.cpp
Open Save ... ~Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V03 ( Sort a list )
1 //3. Sort a list
2
3 #include <iostream>
4 #include <list>
5 #include <iterator>
6 using namespace std;
7
8 // function for printing the elements in a list
9 void printList(list<int> g)
10 {
11     list<int>::iterator it;
12     for (it = g.begin(); it != g.end(); ++it)
13         cout << *it << " ";
14     cout << '\n';
15 }
16 int main()
17 {
18     list<int> l;
19     int n;
20     cout << "Enter the size of the List \n";
21     cin >> n;
22     for (int i = 0; i < n; i++)
23     {
24         int ele;
25         cin >> ele;
26         l.push_back(ele);
27     }
28     printList(l);
29
30     l.sort();
31
32     cout << "Printing list after sorting\n";
33
34     printList(l);
35
36     return 0;
37 }
38
39

```

Code Editor:

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V03 ( Sort a list )$ ls
17.1.a.3.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V03 ( Sort a list )$ g++ 17.1.a.3.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V03 ( Sort a list )$ ./a.out
Enter the size of the List
7
9 7 2 4 1 5 0
9 7 2 4 1 5 0
Printing list after sorting
0 1 2 4 5 7 9
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V03 ( Sort a list )$ 

```

The terminal shows the execution of a C++ program named 17.1.a.3.cpp. It prompts the user for the size of a list (7) and then prints the list before and after sorting. The sorted list is 0 1 2 4 5 7 9.

Task 17.1.a.4:**Source Code:**

```

#include <iostream>
#include <list>
#include <iterator>
using namespace std;

// function for printing the elements in a list
void printList(list<int> g)
{
    list<int>::iterator it;
    for (it = g.begin(); it != g.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}
int main()
{
    list<int> l;
    int n;
    cout << "Enter the size of the List \n";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        int ele;
        cin >> ele;
        l.push_back(ele);
    }
    printList(l);

    l.reverse();

    cout << "Printing list after Reversing\n";

    printList(l);

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Code Editor (Left):

```

Activities Terminal ▾
Open ... Save ...
1 //4. Reverse a list
2
3 #include <iostream>
4 #include <list>
5 #include <iterator>
6 using namespace std;
7
8 // function for printing the elements in a list
9 void printList(list<int> g)
10 {
11     list<int>::iterator it;
12     for (it = g.begin(); it != g.end(); ++it)
13         cout << *it << " ";
14     cout << '\n';
15 }
16 int main()
17 {
18     list<int> l;
19     int n;
20     cout << "Enter the size of the List \n";
21     cin >> n;
22     for (int i = 0; i < n; i++)
23     {
24         int ele;
25         cin >> ele;
26         l.push_back(ele);
27     }
28     printList(l);
29
30     l.reverse();
31
32     cout << "Printing list after Reversing\n";
33
34     printList(l);
35
36     return 0;
37 }
38
39

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V04 ( Reverse a List )$ ls
17.1.a.4.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V04 ( Reverse a List )$ g++ 17.1.a.4.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V04 ( Reverse a List )$ ./a.out
Enter the size of the List
5
1 2 3 4 5
1 2 3 4 5
Printing list after Reversing
5 4 3 2 1
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 01 ( List )/Task V04 ( Reverse a List )$ 

```

The terminal output shows the execution of the program. It asks for the size of the list (5), prints the original list (1 2 3 4 5), and then prints the reversed list (5 4 3 2 1).

Task 17.1.b.1:**Source Code:**

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <iterator>
using namespace std;

int main()
{
    vector<int> v;
    int n;
    cout << "Enter the size of the vector \n";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        int ele;
        cin >> ele;
        v.push_back(ele);           //inserting
elements in vector
    }

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Terminal Window:

- Terminal title: *17.1.b.1.cpp
- Terminal path: ~/Documents/OOPS in C++/Practical 17 (Standard Template Libraries (STL))/Task 02 (Vector)/Task V01 (Insert elements into a int vector)
- Terminal content:


```
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL ) )/Task 02 ( Vector )/Task V01 ( Insert elements into a int vector )$ ls
17.1.b.1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL ) )/Task 02 ( Vector )/Task V01 ( Insert elements into a int vector )$ g++ 17.1.b.1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL ) )/Task 02 ( Vector )/Task V01 ( Insert elements into a int vector )$ ./a.out
Enter the size of the vector
5
1 2 3 4 5
```

Code Editor:

- Code editor title: vishu@VICKY:~/Documents/OOPS in C++/Practical 17 (Standard Template Libraries (STL))/Task 02 (Vector)/Task V01 (Insert elements into a int vector)
- Code editor content (C++ code):


```
1 //1. Insert elements into a int vector
2
3 #include <iostream>
4 #include <vector>
5 #include <algorithm>
6 #include <iterator>
7 using namespace std;
8
9
10 int main()
11 {
12     vector<int> v;
13     int n;
14     cout << "Enter the size of the vector \n";
15     cin >> n;
16     for (int i = 0; i < n; i++)
17     {
18         int ele;
19         cin >> ele;
20         v.push_back(ele);           //inserting elements in vector
21     }
22
23
24
25     return 0;
26 }
```
- Code editor status bar:
 - C++ Tab Width: 4
 - Ln 34, Col 1
 - INS

Task 17.1.b.2:**Source Code:**

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <iterator>
using namespace std;

// function for printing the elements in a vector
void printvector(vector<int> v)
{
    vector<int>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}
int main()
{
    vector<int> v;
    int n;
    cout << "Enter the size of the vector \n";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        int ele;
        cin >> ele;
        v.push_back(ele);
    }

    printvector(v);

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a terminal window titled "Terminal". The terminal window has two tabs: "17.1.b.1.cpp" and "17.1.b.2.cpp". The "17.1.b.1.cpp" tab contains C++ code for printing elements of a vector using an iterator. The "17.1.b.2.cpp" tab shows the output of the program, which is a sequence of numbers from 1 to 5. Below the terminal is a file browser window titled "vishu@VICKY: ~/Documents/OOPS in C++/Practical 17 (Standard Template Libraries (STL))/Task 02 (Vector)/Task V02 (Iterate the vector using iterator and print it)\$ ls". The file browser lists several files including "17.1.b.1.cpp", "17.1.b.2.cpp", and "17.1.b.2.out". At the bottom of the screen, there is a dock with various application icons.

```

1 //2. Iterate this vector using iterator and print it
2
3 #include <iostream>
4 #include <vector>
5 #include <algorithm>
6 #include <iterator>
7 using namespace std;
8
9 // function for printing the elements in a vector
10 void printvector(vector<int> v)
11 {
12     vector<int>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << *it << " ";
15     cout << '\n';
16 }
17 int main()
18 {
19     vector<int> v;
20     int n;
21     cout << "Enter the size of the vector \n";
22     cin >> n;
23     for (int i = 0; i < n; i++)
24     {
25         int ele;
26         cin >> ele;
27         v.push_back(ele);
28     }
29     printvector(v);
30
31     return 0;
32 }
33
34
35
36
37

```

Task 17.1.b.3:**Source Code:**

```

#include <iostream>
#include <vector>
#include <algorithm>
#include <iterator>
using namespace std;

// function for printing the elements in a vector
void printvector(vector<int> v)
{
    vector<int>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}
int main()
{
    vector<int> v;
    int n;
    cout << "Enter the size of the vector \n";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        int ele;
        cin >> ele;
        v.push_back(ele);
    }

    printvector(v);

    cout << "Size of the vector is : " << v.size() <<
"\t Capacity of vector is : " << v.capacity() <<
"\n";

    return 0;
}

```

Output

```

Activities Terminal ▾ Jan 9 18:04
Open ... 17.1.b.3.cpp ~/Documents/OOPS in C++/Practical 17 (Standard Template Libraries (STL) )/Task 02 ( Vector )/Task V03 ( Find size of a capacity and max size of a vector )
Save ...
1//3. Find size of a capacity and max size of a vector
2
3
4 #include <iostream>
5 #include <vector>
6 #include <algorithm>
7 #include <iterator>
8 using namespace std;
9
10 // function for printing the elements in a vector
11 void printvector(vector<int> v)
12 {
13     vector<int>::iterator it;
14     for (it = v.begin(); it != v.end(); ++it)
15         cout << *it << " ";
16     cout << '\n';
17 }
18 int main()
19 {
20     vector<int> v;
21     int n;
22     cout << "Enter the size of the vector \n";
23     cin >> n;
24     for (int i = 0; i < n; i++)
25     {
26         int ele;
27         cin >> ele;
28         v.push_back(ele);
29     }
30
31     printvector(v);
32
33     cout << "Size of the vector is : " << v.size() << "\t Capacity of vector is : " << v.capacity() << "\n";
34
35     return 0;
36 }
37
38
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL) )/Task 02 ( Vector )/Task V03 ( Find size of a capacity and max size of a vector )$ ls
17.1.b.3.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL) )/Task 02 ( Vector )/Task V03 ( Find size of a capacity and max size of a vector )$ g++ 17.1.b.3.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL) )/Task 02 ( Vector )/Task V03 ( Find size of a capacity and max size of a vector )$ ./a.out
Enter the size of the vector
6
1 2 3 0 8 3
1 2 3 0 8 3
Size of the vector is : 6      Capacity of vector is : 8
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL) )/Task 02 ( Vector )/Task V03 ( Find size of a capacity and max size of a vector )$ 

```

C++ ▾ Tab Width: 4 ▾ Ln 38, Col 1 ▾ INS

Firefox, LibreOffice, VLC, iTunes, WinRAR, Aria2, Transmission, Google Chrome, Bookmarks, Dash, Terminal, File Manager, Home, System Settings, Help.

Task 17.1.b.4:**Source Code:**

```

#include <iostream>
#include <vector>
#include <algorithm>
#include <iterator>
using namespace std;

// function for printing the elements in a vector
void printvector(vector<int> v)
{
    vector<int>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}
int main()
{
    vector<int> v;
    int n;
    cout << "Enter the size of the vector \n";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        int ele;
        cin >> ele;
        v.push_back(ele);
    }

    printvector(v);

    v.resize(2 * n, 0);
    cout
        << "Printing vector after resizing and
initialising after 0\n";
    printvector(v);

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a terminal window titled 'Terminal' with the command 'ls' running, showing the directory structure of a C++ project. In the bottom left corner, there is a code editor window titled 'Activities' with the file '17.1.b.4.cpp' open. The code is a C++ program that demonstrates how to resize a vector. It includes comments explaining the purpose of each section and uses standard library headers like <iostream> and <vector>. The terminal output shows the program being compiled with g++ and run, displaying the initial elements of the vector and its state after resizing.

```

1 //4. Resize a vector
2
3 #include <iostream>
4 #include <vector>
5 #include <algorithm>
6 #include <iterator>
7 using namespace std;
8
9 // function for printing the elements in a vector
10 void printvector(vector<int> v)
11 {
12     vector<int>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << *it << " ";
15     cout << '\n';
16 }
17 int main()
18 {
19     vector<int> v;
20     int n;
21     cout << "Enter the size of the vector \n";
22     cin >> n;
23     for (int i = 0; i < n; i++)
24     {
25         int ele;
26         cin >> ele;
27         v.push_back(ele);
28     }
29     printvector(v);
30
31     v.resize(2 * n, 0);
32     cout
33         << "Printing vector after resizing and initialising after 0\n";
34     printvector(v);
35
36     return 0;
37 }
38
39
40
41
42

```

Task 17.1.b.5:**Source Code:**

```

#include <iostream>
#include <vector>
#include <algorithm>
#include <iterator>
using namespace std;

// function for printing the elements in a vector
void printvector(vector<int> v)
{
    vector<int>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}
int main()
{
    vector<int> v;
    int n;
    cout << "Enter the size of the vector \n";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        int ele;
        cin >> ele;
        v.push_back(ele);
    }

    printvector(v);

    cout << "Checking vector is empty or not after
:\n";
    if (v.empty())
        cout << "Vector is empty\t";
    else
        cout << "Vector is not empty";

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Terminal Window:

- Title bar: Activities Terminal ▾
- Time: Jan 9 18:09
- File: *17.1.b.5.cpp
- Path: ~/Documents/OOPS in C++/Practical 17 (Stan.../V05 (checks if the vector is empty or not)
- Content (Terminal Output):


```
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 02 ( Vector )/Task V05 ( checks if the vector is empty or not )$ ls
17.1.b.5.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 02 ( Vector )/Task V05 ( checks if the vector is empty or not )$ g+
+ 17.1.b.5.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task 02 ( Vector )/Task V05 ( checks if the vector is empty or not )$ ./a.out
Enter the size of the vector
0
Checking vector is empty or not after :
Vector is empty vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Temp
late Libraries (STL ) )/Task 02 ( Vector )/Task V05 ( checks if the vector is em
pty or not )$ 
```

Code Editor Window:

- Title bar: Open ▾
- Code content (17.1.b.5.cpp):


```
1 //5. checks if the vector is empty or not
2
3 #include <iostream>
4 #include <vector>
5 #include <algorithm>
6 #include <iterator>
7 using namespace std;
8
9 // function for printing the elements in a vector
10 void printvector(vector<int> v)
11 {
12     vector<int>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << *it << " ";
15     cout << '\n';
16 }
17 int main()
18 {
19     vector<int> v;
20     int n;
21     cout << "Enter the size of the vector \n";
22     cin >> n;
23     for (int i = 0; i < n; i++)
24     {
25         int ele;
26         cin >> ele;
27         v.push_back(ele);
28     }
29     printvector(v);
30
31     cout << "Checking vector is empty or not after :\n";
32     if (v.empty())
33         cout << "Vector is empty\n";
34     else
35         cout << "Vector is not empty\n";
36
37     return 0;
38 }
39
40
41
42
```
- Status bar: C++ ▾ Tab Width: 4 ▾ Ln 41, Col 1 ▾ INS

Task 17.1.c.1:**Source Code:**

```

#include <bits/stdc++.h>
#include <map>
#include <algorithm>
#include <iterator>
using namespace std;

// function for printing the elements in a map
void printmap(map<int, string> v)
{
    map<int, string>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << it->first << " " << it->second <<
"\n";
    cout << '\n';
}

int main()
{
    map<int, string> m;
    // Inserting Elements in map

    m.insert(pair<int, string>(1, "coders"));
    m.insert(pair<int, string>(2, "begin"));

    // printing map
    printmap(m);

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //1. Insert elements into a <int, string> map
2
3 #include <bits/stdc++.h>
4 #include <map>
5 #include <algorithm>
6 #include <iterator>
7 using namespace std;
8
9 // function for printing the elements in a map
10 void printmap(map<int, string> v)
11 {
12     map<int, string>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << it->first << " " << it->second << "\n";
15     cout << '\n';
16 }
17
18 int main()
19 {
20     map<int, string> m;
21     // Inserting Elements in map
22
23     m.insert(pair<int, string>(1, "coders"));
24     m.insert(pair<int, string>(2, "begin"));
25
26     // printing map
27     printmap(m);
28
29     return 0;
30 }
31
32
33
34
35
36
37
38
39

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL) )/Task V03 ( Map )/Task V01 ( Insert elements into a <int, string> map )
$ g++ 17.1.c.1.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL) )/Task V03 ( Map )/Task V01 ( Insert elements into a <int, string> map )
$ ./a.out
1 coders
2 begin
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL) )/Task V03 ( Map )/Task V01 ( Insert elements into a <int, string> map )
$ 

```

The terminal output shows the execution of the program, which prints the contents of the map to the console.

Task 17.1.c.2:**Source Code:**

```

#include <bits/stdc++.h>
#include <map>
#include <algorithm>
#include <iterator>
using namespace std;

// function for printing the elements in a map
void printmap(map<int, string> v)
{
    map<int, string>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << it->first << " " << it->second <<
"\n";
    cout << '\n';
}

int main()
{
    map<int, string> m;
    // Inserting Elements in map

    m.insert(pair<int, string>(6, "Map"));
    m.insert(pair<int, string>(3, "Iterator"));
    m.insert(pair<int, string>(8, "pair"));
    m.insert(pair<int, string>(1, "coders"));
    m.insert(pair<int, string>(9, "algorithm"));
    m.insert(pair<int, string>(2, "begin"));
    m.insert(pair<int, string>(5, "end"));

    // printing map
    printmap(m);

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //2. insert elements in random order
2
3 #include <bits/stdc++.h>
4 #include <map>
5 #include <algorithm>
6 #include <iterator>
7 using namespace std;
8
9 // function for printing the elements in a map
10 void printmap(map<int, string> v)
11 {
12     map<int, string>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << it->first << " " << it->second << "\n";
15     cout << '\n';
16 }
17
18 int main()
19 {
20     map<int, string> m;
21     // Inserting Elements in random Order map
22
23     m.insert(pair<int, string>(6, "Map"));
24     m.insert(pair<int, string>(3, "Iterator"));
25     m.insert(pair<int, string>(8, "pair"));
26     m.insert(pair<int, string>(1, "coders"));
27     m.insert(pair<int, string>(9, "algorithm"));
28     m.insert(pair<int, string>(2, "begin"));
29     m.insert(pair<int, string>(5, "end"));
30
31     // printing map
32     printmap(m);
33
34     return 0;
35 }
36
37
38
39
40
41
42

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V02 ( insert elements in random order)$ ls
17.1.c.2.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V02 ( insert elements in random order $ g++ 17.1.c.2.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V02 ( insert elements in random order $ ./a.out
1 coders
2 begin
3 Iterator
5 end
6 Map
8 pair
9 algorithm

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V02 ( insert elements in random order $ 

```

Task 17.1.c.3:**Source Code:**

```

#include <bits/stdc++.h>
#include <map>
#include <algorithm>
#include <iterator>
using namespace std;

// function for printing the elements in a map
void printmap(map<int, string> v)
{
    map<int, string>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << it->first << " " << it->second <<
"\n";
    cout << '\n';
}

int main()
{
    map<int, string> m;
    // Inserting Elements in map

    m.insert(pair<int, string>(6, "Map"));
    m.insert(pair<int, string>(3, "Iterator"));
    m.insert(pair<int, string>(8, "pair"));
    m.insert(pair<int, string>(1, "coders"));
    m.insert(pair<int, string>(9, "algorithm"));
    m.insert(pair<int, string>(2, "begin"));
    m.insert(pair<int, string>(5, "end"));

    // printing map
    printmap(m);

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //3. Iterate this map using iterator and print its keys and values
2
3 #include <bits/stdc++.h>
4 #include <map>
5 #include <algorithm>
6 #include <iterator>
7 using namespace std;
8
9 // function for printing the elements in a map
10 void printmap(map<int, string> v)
11 {
12     map<int, string>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << it->first << " " << it->second << "\n";
15     cout << '\n';
16 }
17
18 int main()
19 {
20     map<int, string> m;
21     // Inserting Elements in map
22
23     m.insert(pair<int, string>(6, "Map"));
24     m.insert(pair<int, string>(3, "Iterator"));
25     m.insert(pair<int, string>(8, "pair"));
26     m.insert(pair<int, string>(1, "coders"));
27     m.insert(pair<int, string>(9, "algorithm"));
28     m.insert(pair<int, string>(2, "begin"));
29     m.insert(pair<int, string>(5, "end"));
30
31     // printing map
32     printmap(m);
33
34     return 0;
35 }
36
37
38
39
40
41
42

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V03 ( Iterate this map using iterator and print its keys and values )$ ls
17.1.c.3.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V03 ( Iterate this map using iterator and print its keys and values )$ g++ 17.1.c.3.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V03 ( Iterate this map using iterator and print its keys and values )$ ./a.out
1 coders
2 begin
3 Iterator
5 end
6 Map
8 pair
9 algorithm

```

Task 17.1.c.4:**Source Code:**

```

#include <bits/stdc++.h>
#include <map>
#include <algorithm>
#include <iterator>
using namespace std;

// function for printing the elements in a map
void printmap(map<int, string> v)
{
    map<int, string>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << it->first << " " << it->second << "\n";
    cout << '\n';
}

int main()
{
    map<int, string> m;
    // Inserting Elements in map

    m.insert(pair<int, string>(6, "Map"));
    m.insert(pair<int, string>(3, "Iterator"));
    m.insert(pair<int, string>(8, "pair"));
    m.insert(pair<int, string>(1, "coders"));
    m.insert(pair<int, string>(9, "algorithm"));
    m.insert(pair<int, string>(2, "begin"));
    m.insert(pair<int, string>(5, "end"));

    // printing map
    printmap(m);

    // Find an element as key from this map
    auto itr = m.find(9);
    cout << "itr is pointing to \n"
        << itr->first << " " << itr->second << " \n";

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Terminal Window:

- Terminal title: 17.1.c.4.GPP
- Terminal path: ~/Documents/OOPS in C++/Practical 17 (Standard Template Libraries (STL))/Task V03 (Map)/Task V04 (Find an element as key from this map)
- Terminal output:


```
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V04 ( Find an element as key from this map )$ ls
17.1.c.4.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V04 ( Find an element as key from this map )$ g++
17.1.c.4.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V04 ( Find an element as key from this map )$ ./a.out
1 coders
2 begin
3 Iterator
5 end
6 Map
8 pair
9 algorithm

itr is pointing to
9 algorithm
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V04 ( Find an element as key from this map )$ 
```

Code Editor:

```

1 //4. Find an element as key from this map
2
3 #include <bits/stdc++.h>
4 #include <map>
5 #include <algorithm>
6 #include <iterator>
7 using namespace std;
8
9 // function for printing the elements in a map
10 void printmap(map<int, string> v)
11 {
12     map<int, string>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << it->first << " " << it->second << "\n";
15     cout << '\n';
16 }
17
18 int main()
19 {
20     map<int, string> m;
21     // Inserting Elements in map
22
23     m.insert(pair<int, string>(6, "Map"));
24     m.insert(pair<int, string>(3, "Iterator"));
25     m.insert(pair<int, string>(8, "pair"));
26     m.insert(pair<int, string>(1, "coders"));
27     m.insert(pair<int, string>(9, "algorithm"));
28     m.insert(pair<int, string>(2, "begin"));
29     m.insert(pair<int, string>(5, "end"));
30
31     // printing map
32     printmap(m);
33
34     // Find an element as key from this map
35     auto itr = m.find(3);
36     cout << "itr is pointing to \n"
37         << itr->first << " " << itr->second << "\n";
38
39     return 0;
40 }
41
42

```

Task 17.1.c.5:**Source Code:**

```

#include <bits/stdc++.h>
#include <map>
#include <algorithm>
#include <iostream>
using namespace std;

// function for printing the elements in a map
void printmap(map<int, string> v)
{
    map<int, string>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << it->first << " " << it->second << "\n";
    cout << '\n';
}

int main()
{
    map<int, string> m;
    // Inserting Elements in map

    m.insert(pair<int, string>(6, "Map"));
    m.insert(pair<int, string>(3, "Iterator"));
    m.insert(pair<int, string>(8, "pair"));
    m.insert(pair<int, string>(1, "coders"));
    m.insert(pair<int, string>(9, "algorithm"));
    m.insert(pair<int, string>(2, "begin"));
    m.insert(pair<int, string>(5, "end"));

    // printing map
    printmap(m);

    //Assigning one map to another
    map<int, string> copyMap = m;
    cout << "Printing copyMap :\n";
    printmap(copyMap);

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //5. assigning the elements from map1 to map2
2
3 #include <bits/stdc++.h>
4 #include <map>
5 #include <algorithm>
6 #include <iostream>
7 using namespace std;
8
9 // function for printing the elements in a map
10 void printmap(map<int, string> v)
11 {
12     map<int, string>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << it->first << " " << it->second << "\n";
15     cout << '\n';
16 }
17
18 int main()
19 {
20     map<int, string> m;
21     // Inserting Elements in map
22
23     m.insert(pair<int, string>(6, "Map"));
24     m.insert(pair<int, string>(3, "Iterator"));
25     m.insert(pair<int, string>(8, "pair"));
26     m.insert(pair<int, string>(1, "coders"));
27     m.insert(pair<int, string>(9, "algorithm"));
28     m.insert(pair<int, string>(2, "begin"));
29     m.insert(pair<int, string>(5, "end"));
30
31     // printing map
32     printmap(m);
33
34     //Assigning one map to another
35     map<int, string> copyMap = m;
36     cout << "Printing copyMap :\n";
37     printmap(copyMap);
38
39     return 0;
40 }
41
42

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL ) )/Task V03 ( Map )/Task V05 ( assigning the elements from map1 to map2 )
$ ./a.out
1 coders
2 begin
3 Iterator
5 end
6 Map
8 pair
9 algorithm

Printing copyMap :
1 coders
2 begin
3 Iterator
5 end
6 Map
8 pair
9 algorithm

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries (STL ) )/Task V03 ( Map )/Task V05 ( assigning the elements from map1 to map2 )
$ 

```

Task 17.1.c.6:**Source Code:**

```

#include <bits/stdc++.h>
#include <map>
#include <algorithm>
#include <iostream>
using namespace std;

// function for printing the elements in a map
void printmap(map<int, string> v)
{
    map<int, string>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << it->first << " " << it->second << "\n";
    cout << '\n';
}

int main()
{
    map<int, string> m;
    // Inserting Elements in map

    m.insert(pair<int, string>(6, "Map"));
    m.insert(pair<int, string>(3, "Iterator"));
    m.insert(pair<int, string>(8, "pair"));
    m.insert(pair<int, string>(1, "coders"));
    m.insert(pair<int, string>(9, "algorithm"));
    m.insert(pair<int, string>(2, "begin"));
    m.insert(pair<int, string>(5, "end"));

    // printing map
    printmap(m);

    // Deleting a key from map
    cout << "Deleting a key-value from copyMap : 9
algorithm\n";
    m.erase(9);
    cout << "Printing map After deleting key = 9 from it\n";
    printmap(m);

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //6. remove all elements with key = x (any key present in map)
2
3 #include <bits/stdc++.h>
4 #include <map>
5 #include <algorithm>
6 #include <iterator>
7 using namespace std;
8
9 // function for printing the elements in a map
10 void printmap(map<int, string> v)
11 {
12     map<int, string>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << it->first << " " << it->second << "\n";
15     cout << '\n';
16 }
17
18 int main()
19 {
20     map<int, string> m;
21     // Inserting Elements in map
22
23     m.insert(pair<int, string>(6, "Map"));
24     m.insert(pair<int, string>(3, "Iterator"));
25     m.insert(pair<int, string>(8, "pair"));
26     m.insert(pair<int, string>(1, "coders"));
27     m.insert(pair<int, string>(9, "algorithm"));
28     m.insert(pair<int, string>(2, "begin"));
29     m.insert(pair<int, string>(5, "end"));
30
31     // printing map
32     printmap(m);
33
34     // Deleting a key from map
35     cout << "Deleting a key-value from copyMap : 9 algorithm\n";
36     m.erase(9);
37     cout << "Printing map After deleting key = 9 from it\n";
38     printmap(m);
39
40     return 0;
41 }
42

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V06 ( remove all elements with key = x (any key present in map) )$ g++ 17.1.c.6.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V06 ( remove all elements with key = x (any key present in map) )$ ./a.out
1 coders
2 begin
3 Iterator
5 end
6 Map
8 pair
9 algorithm

Deleting a key-value from copyMap : 9 algorithm
Printing map After deleting key = 9 from it
1 coders
2 begin
3 Iterator
5 end
6 Map
8 pair

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V06 ( remove all elements with key = x (any key present in map) )$ 

```

Task 17.1.c.7:**Source Code:**

```

#include <bits/stdc++.h>
#include <map>
#include <algorithm>
#include <iterator>
using namespace std;

// function for printing the elements in a map
void printmap(map<int, string> v)
{
    map<int, string>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << it->first << " " << it->second <<
"\n";
    cout << '\n';
}

int main()
{
    map<int, string> m;
    // Inserting Elements in map

    m.insert(pair<int, string>(6, "Map"));
    m.insert(pair<int, string>(3, "Iterator"));
    m.insert(pair<int, string>(8, "pair"));
    m.insert(pair<int, string>(1, "coders"));
    m.insert(pair<int, string>(9, "algorithm"));
    m.insert(pair<int, string>(2, "begin"));
    m.insert(pair<int, string>(5, "end"));

    // printing map
    printmap(m);

    // Finding size and max size of map
    cout << "Size of the map is : " << m.size() <<
"\t maxSize of map is : " << m.max_size() << "\n";

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a file browser window.

Terminal Window:

```

Activities Terminal ▾ Jan 9 19:23
17.1.c.7.cpp
~/Documents/OOPS in C++/Practical 17 (Standard Template Libraries (STL) )/Task V03 ( Map )/Task V07 ( Find size, max size of a map )
Save ...
```

```

1 //7. Find size, max size of a map
2
3 #include <bits/stdc++.h>
4 #include <map>
5 #include <algorithm>
6 #include <iterator>
7 using namespace std;
8
9 // function for printing the elements in a map
10 void printmap(map<int, string> v)
11 {
12     map<int, string>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << it->first << " " << it->second << "\n";
15     cout << '\n';
16 }
17
18 int main()
19 {
20     map<int, string> m;
21     // Inserting Elements in map
22
23     m.insert(pair<int, string>(6, "Map"));
24     m.insert(pair<int, string>(3, "Iterator"));
25     m.insert(pair<int, string>(8, "pair"));
26     m.insert(pair<int, string>(1, "coders"));
27     m.insert(pair<int, string>(9, "algorithm"));
28     m.insert(pair<int, string>(2, "begin"));
29     m.insert(pair<int, string>(5, "end"));
30
31     // printing map
32     printmap(m);
33
34     // Finding size and max size of map
35     cout << "Size of the map is : " << m.size() << "\t maxSize of map is : " << m.max_size() << "\n";
36
37     return 0;
38 }
```

File Browser Window:

```

vishu@VICKY: ~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V07 ( Find size, max size of a map )$ g++ 17.1.c.7.cpp
vishu@VICKY: ~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V07 ( Find size, max size of a map )$ ./a.out
1 coders
2 begin
3 Iterator
4 end
5 Map
6 pair
7 algorithm
```

Output from the terminal window:

```

Size of the map is : 7  maxSize of map is : 128102389400760775
```

Task 17.1.c.8:**Source Code:**

```

#include <bits/stdc++.h>
#include <map>
#include <algorithm>
#include <iterator>
using namespace std;

// function for printing the elements in a map
void printmap(map<int, string> v)
{
    map<int, string>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << it->first << " " << it->second << "\n";
    cout << '\n';
}

int main()
{
    map<int, string> m;
    // Inserting Elements in map

    m.insert(pair<int, string>(6, "Map"));
    m.insert(pair<int, string>(3, "Iterator"));
    m.insert(pair<int, string>(8, "pair"));
    m.insert(pair<int, string>(1, "coders"));
    m.insert(pair<int, string>(9, "algorithm"));
    m.insert(pair<int, string>(2, "begin"));
    m.insert(pair<int, string>(5, "end"));

    // printing map
    printmap(m);

    // Checking a map is empty or not
    cout << "Checking map is empty or not after :\n";
    if (m.empty())
        cout << "Map is empty\n";
    else
        cout << "Map is not empty\n";

    return 0;
}

```

Output

The screenshot shows a terminal window titled "Terminal" with the file path "/Documents/OOPS in C++/Practical 17 (Standard Template Libraries (STL))/Task V03 (Map)/Task V08 (checks if this map is empty or not)". The terminal window displays the following output:

```
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V08 ( checks if this map is empty or not )$ ls
17.1.c.8.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V08 ( checks if this map is empty or not )$ g++ 17
.1.c.8.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V08 ( checks if this map is empty or not )$ ./a.out
1 coders
2 begin
3 Iterator
5 end
6 Map
8 pair
9 algorithm

Checking map is empty or not after :
Map is not empty
```

The terminal window is part of a desktop environment with a dark theme. The desktop bar at the bottom shows various application icons.

```
1 //0. checks if this map is empty or not
2
3 #include <bits/stdc++.h>
4 #include <map>
5 #include <algorithm>
6 #include <iterator>
7 using namespace std;
8
9 // function for printing the elements in a map
10 void printmap(map<int, string> v)
11 {
12     map<int, string>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << it->first << " " << it->second << "\n";
15     cout << '\n';
16 }
17
18 int main()
19 {
20     map<int, string> m;
21     // Inserting Elements in map
22
23     m.insert(pair<int, string>(6, "Map"));
24     m.insert(pair<int, string>(3, "Iterator"));
25     m.insert(pair<int, string>(8, "pair"));
26     m.insert(pair<int, string>(1, "coders"));
27     m.insert(pair<int, string>(9, "algorithm"));
28     m.insert(pair<int, string>(2, "begin"));
29     m.insert(pair<int, string>(5, "end"));
30
31     // printing map
32     printmap(m);
33
34     // Checking a map is empty or not
35     cout << "Checking map is empty or not after :\n";
36     if (m.empty())
37         cout << "Map is empty\n";
38     else
39         cout << "Map is not empty\n";
40
41     return 0;
42 }
```

Task 17.1.c.9:**Source Code:**

```

#include <bits/stdc++.h>
#include <map>
#include <algorithm>
#include <iostream>
using namespace std;

// function for printing the elements in a map
void printmap(map<int, string> v)
{
    map<int, string>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << it->first << " " << it->second << "\n";
    cout << '\n';
}

int main()
{
    map<int, string> m;
    // Inserting Elements in map

    m.insert(pair<int, string>(6, "Map"));
    m.insert(pair<int, string>(3, "Iterator"));
    m.insert(pair<int, string>(8, "pair"));
    m.insert(pair<int, string>(1, "coders"));
    m.insert(pair<int, string>(9, "algorithm"));
    m.insert(pair<int, string>(2, "begin"));
    m.insert(pair<int, string>(5, "end"));

    // printing map
    printmap(m);

    // Clearing a map
    m.clear();
    cout << "Printing a Map after Clearing it :\n";
    printmap(m);

    return 0;
}

```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //1. Insert elements into a <int, string> map
2
3 #include <bits/stdc++.h>
4 #include <map>
5 #include <algorithm>
6 #include <iostream>
7 using namespace std;
8
9 // function for printing the elements in a map
10 void printmap(map<int, string> v)
11 {
12     map<int, string>::iterator it;
13     for (it = v.begin(); it != v.end(); ++it)
14         cout << it->first << " " << it->second << "\n";
15     cout << '\n';
16 }
17
18 int main()
19 {
20     map<int, string> m;
21     // Inserting Elements in map
22
23     m.insert(pair<int, string>(6, "Map"));
24     m.insert(pair<int, string>(3, "Iterator"));
25     m.insert(pair<int, string>(8, "pair"));
26     m.insert(pair<int, string>(1, "coders"));
27     m.insert(pair<int, string>(9, "algorithm"));
28     m.insert(pair<int, string>(2, "begin"));
29     m.insert(pair<int, string>(5, "end"));
30
31     // printing map
32     printmap(m);
33
34     // Clearing a map
35     m.clear();
36     cout << "Printing a Map after Clearing it :\n";
37     printmap(m);
38
39     return 0;
40 }
41
42

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V09 ( Clear a map )$ ls
17.1.c.9.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V09 ( Clear a map )$ g++ 17.1.c.9.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V09 ( Clear a map )$ ./a.out
1 coders
2 begin
3 Iterator
5 end
6 Map
8 pair
9 algorithm

Printing a Map after Clearing it :

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V03 ( Map )/Task V09 ( Clear a map )$ 

```

Task 17.1.d.1:**Source Code:**

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

void printvector(vector<int> v)
{
    vector<int>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}
void printArray(int a[], int n)
{
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";
    cout << '\n';
}
int main()
{
    // creating a Array
    int arr[] = {2, 7, 5, 1, 8, 3, 3, 4};

    // Converting a array into vector
    int n = sizeof(arr) / sizeof(arr[0]);
    vector<int> vec(arr, arr + n);

    cout<<"\n\nElements in Array\n\n";
    printArray(arr,n);

    cout<<"\n\nElements in Vector\n\n";
    printvector(vec);

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //1. Covert Array into Vector
2
3 #include <bits/stdc++.h>
4 using namespace std;
5
6 void printvector(vector<int> v)
7 {
8     vector<int>::iterator it;
9     for (it = v.begin(); it != v.end(); ++it)
10        cout << *it << " ";
11    cout << '\n';
12 }
13 void printArray(int a[], int n)
14 {
15     for (int i = 0; i < n; i++)
16         cout << a[i] << " ";
17     cout << '\n';
18 }
19 int main()
20 {
21     // creating a Array
22     int arr[] = {2, 7, 5, 1, 8, 3, 3, 4};
23
24     // Converting a array into vector
25     int n = sizeof(arr) / sizeof(arr[0]);
26     vector<int> vec(arr, arr + n);
27
28     cout << "\nElements in Array\n";
29     printArray(arr, n);
30
31     cout << "\nElements in Vector\n";
32     printvector(vec);
33
34     return 0;
35 }
36
37
38

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V01 ( Covert Array into Vector )$ g++ 17.1.d
1.cpp
./vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V01 ( Covert Array into Vector )$ ./a.out

Elements in Array
2 7 5 1 8 3 3 4

Elements in Vector
2 7 5 1 8 3 3 4

```

Task 17.1.d.2:**Source Code:**

```
void printvector(vector<int> v)
{
    vector<int>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}

int main()
{
    // creating a vector

    vector<int> vec{2, 7, 5, 1, 8, 3, 3, 4};

    // sorting an vector

    cout << "Printing vector before sorting : \n";
    printvector(vec);

    sort(vec.begin(), vec.end());

    cout << "Printing vector after sorting : \n";
    printvector(vec);

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //2. Sort an Vector
2
3 #include <bits/stdc++.h>
4 using namespace std;
5
6 void printvector(vector<int> v)
7 {
8     vector<int>::iterator it;
9     for (it = v.begin(); it != v.end(); ++it)
10        cout << *it << " ";
11    cout << '\n';
12 }
13 void printArray(int a[], int n)
14 {
15     for (int i = 0; i < n; i++)
16         cout << a[i] << " ";
17     cout << '\n';
18 }
19 int main()
20 {
21     // creating a Array
22
23     vector<int> vec{7, 5, 1, 8, 3, 3, 4};
24
25     // sorting an vector
26
27     cout << "Printing vector before sorting : \n";
28     printvector(vec);
29
30     sort(vec.begin(), vec.end());
31
32     cout << "Printing vector after sorting : \n";
33     printvector(vec);
34
35
36
37     return 0;
38 }
39
40
41

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V02 ( Sort an Vector )$ ls
17.1.d.2.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V02 ( Sort an Vector )$ g++ 17.1.d.2.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V02 ( Sort an Vector )$ ./a.out
Printing vector before sorting :
2 7 5 1 8 3 3 4
Printing vector after sorting :
1 2 3 3 4 5 7 8
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V02 ( Sort an Vector )$ 

```

Task 17.1.d.3:**Source Code:**

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

void printvector(vector<int> v)
{
    vector<int>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}

int main()
{
    // creating a vector

    vector<int> vec{2, 7, 5, 1, 8, 3, 3, 4};

    cout << "Printing vector after Reversing : \n";

    printvector(vec);

    // Reversing a vector

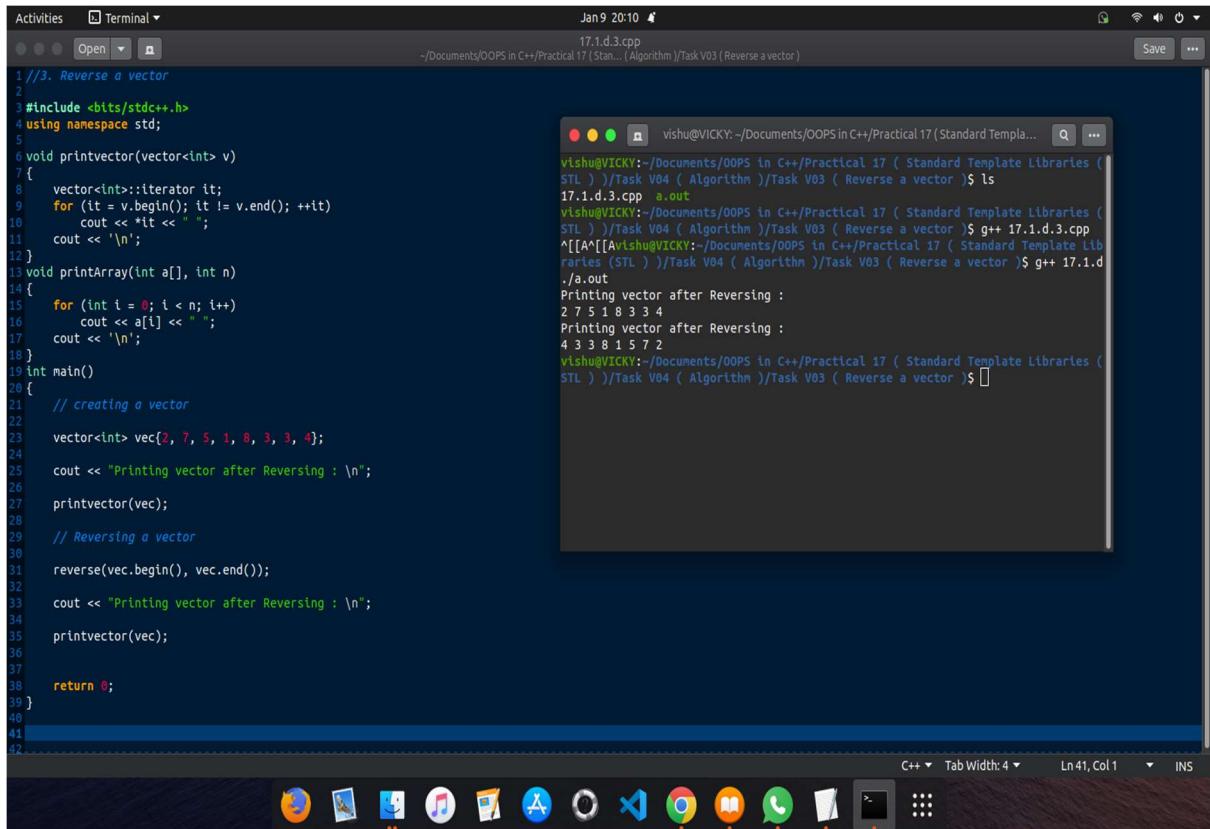
    reverse(vec.begin(), vec.end());

    cout << "Printing vector after Reversing : \n";

    printvector(vec);

    return 0;
}
```

Output



The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Code Editor (Left):

```

1 //3. Reverse a vector
2
3 #include <bits/stdc++.h>
4 using namespace std;
5
6 void printvector(vector<int> v)
7 {
8     vector<int>::iterator it;
9     for (it = v.begin(); it != v.end(); ++it)
10        cout << *it << " ";
11    cout << '\n';
12 }
13 void printArray(int a[], int n)
14 {
15     for (int i = 0; i < n; i++)
16         cout << a[i] << " ";
17     cout << '\n';
18 }
19 int main()
20 {
21     // creating a vector
22
23     vector<int> vec{2, 7, 5, 1, 8, 3, 3, 4};
24
25     cout << "Printing vector after Reversing : \n";
26
27     printvector(vec);
28
29     // Reversing a vector
30
31     reverse(vec.begin(), vec.end());
32
33     cout << "Printing vector after Reversing : \n";
34
35     printvector(vec);
36
37
38     return 0;
39 }
40
41
42

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V03 ( Reverse a vector )$ ls
17.1.d.3.cpp  a.out
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V03 ( Reverse a vector )$ g++ 17.1.d.3.cpp
^[[A^[[AVishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V03 ( Reverse a vector )$ g++ 17.1.d
./a.out
Printing vector after Reversing :
2 7 5 1 8 3 3 4
Printing vector after Reversing :
4 3 3 8 1 5 7 2
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V03 ( Reverse a vector )$ 

```

Task 17.1.d.4:**Source Code:**

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

void printvector(vector<int> v)
{
    vector<int>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}

int main()
{
    // creating a vector

    vector<int> vec{2, 7, 5, 1, 8, 3, 3, 4};

    cout<<"\n\nElements in Vector\n\n";
    printvector(vec);

    cout << "Printing max and min element of vector : \n";
    cout << "Max element is : " <<
    *(max_element(vec.begin(), vec.end()));

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Code Editor (Left):

```

1 //4. Max element in a Vector
2
3 #include <bits/stdc++.h>
4 using namespace std;
5
6 void printvector(vector<int> v)
7 {
8     vector<int>::iterator it;
9     for (it = v.begin(); it != v.end(); ++it)
10        cout << *it << " ";
11    cout << '\n';
12 }
13 void printArray(int a[], int n)
14 {
15     for (int i = 0; i < n; i++)
16         cout << a[i] << " ";
17     cout << '\n';
18 }
19 int main()
20 {
21     // creating a vector
22
23     vector<int> vec{2, 7, 5, 1, 8, 3, 3, 4};
24
25     cout << "\nElements in Vector\n\n";
26     printvector(vec);
27
28     cout << "Printing max and min element of vector : \n";
29     cout << "Max element is : " << *(max_element(vec.begin(), vec.end()));
30
31     return 0;
32 }
33
34
35

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V04 ( Max element in a Vector )$ ls
17.1.d.4.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V04 ( Max element in a Vector )$ g++ 17.1.d.4.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V04 ( Max element in a Vector )$ ./a.out

Elements in Vector
2 7 5 1 8 3 3 4
Printing max and min element of vector :
Max element is : 8
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V04 ( Max element in a Vector )$ 

```

Task 17.1.d.5:**Source Code:**

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

void printvector(vector<int> v)
{
    vector<int>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}

int main()
{
    // creating a vector

    vector<int> vec{2, 7, 5, 1, 8, 3, 3, 4};

    cout<<"\n\nElements in Vector\n\n";
    printvector(vec);

    cout << "Printing min element of vector : \n";
    cout << "Min element is : " <<
    *(min_element(vec.begin(), vec.end()));

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left):

```

1 //4. Min element in a Vector
2
3 #include <bits/stdc++.h>
4 using namespace std;
5
6 void printvector(vector<int> v)
7 {
8     vector<int>::iterator it;
9     for (it = v.begin(); it != v.end(); ++it)
10        cout << *it << " ";
11    cout << '\n';
12 }
13 void printArray(int a[], int n)
14 {
15     for (int i = 0; i < n; i++)
16         cout << a[i] << " ";
17     cout << '\n';
18 }
19 int main()
20 {
21     // creating a vector
22
23     vector<int> vec{2, 7, 5, 1, 8, 3, 3, 4};
24
25     cout<<"\n\nElements in Vector\n\n";
26     printvector(vec);
27
28     cout << "Printing min element of vector : \n";
29     cout << "Min element is : " << *(min_element(vec.begin(), vec.end()));
30
31     return 0;
32 }
33
34
35

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V05 ( Min element in a Vector)$ ls
17.1.d.5.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V05 ( Min element in a Vector)$ g++ 17.1.d.5.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V05 ( Min element in a Vector)$ ./a.out

Elements in Vector
2 7 5 1 8 3 3 4
Printing min element of vector :
Min element is : 1
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V05 ( Min element in a Vector)$ 

```

Task 17.1.d.6:**Source Code:**

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

void printvector(vector<int> v)
{
    vector<int>::iterator it;
    for (it = v.begin(); it != v.end(); ++it)
        cout << *it << " ";
    cout << '\n';
}

int main()
{
    // creating an vecotr
    vector<int> vec {2, 7, 5, 1, 8, 3, 3, 4};

    cout<<"\n\nElements in Vector\n\n";
    printvector(vec);

    // counting occurrence of an element
    cout << "occurrence of an 3 in vector :" <<
    count(vec.begin(), vec.end(), 3) << endl;

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Code Editor (Left):

```

1 //6. Occurrences of x in vector
2
3
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 void printvector(vector<int> v)
8 {
9     vector<int>::iterator it;
10    for (it = v.begin(); it != v.end(); ++it)
11        cout << *it << " ";
12    cout << '\n';
13 }
14 void printArray(int a[], int n)
15 {
16     for (int i = 0; i < n; i++)
17         cout << a[i] << " ";
18     cout << '\n';
19 }
20 int main()
21 {
22     // creating an vector
23     vector<int> vec {2, 7, 5, 1, 8, 3, 3, 4};
24
25     cout << "\n\nElements in Vector\n\n";
26     printvector(vec);
27
28     // counting occurrence of an element
29     cout << "occurrence of an 3 in vector :" << count(vec.begin(), vec.end(), 3) << endl;
30
31     return 0;
32 }
33
34
35
36
37
38
39
40
41
42

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V06 ( Occurrences of x in vector )$ ls
17.1.d.6.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V06 ( Occurrences of x in vector )$ g++ 17.1
.d.6.cpp
.vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V06 ( Occurrences of x in vector )$ ./a.out
Elements in Vector
2 7 5 1 8 3 3 4
occurrence of an 3 in vector :2
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V06 ( Occurrences of x in vector )$ 

```

Task 17.1.d.7:**Source Code:**

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

void printArray(int a[], int n)
{
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";
    cout << '\n';
}

int main()
{
    // creating a Array
    int arr[] = {2, 7, 5, 1, 8, 3, 3, 4};

    int n = sizeof(arr) / sizeof(arr[0]);

    cout<<"\n\nElements in Array before Sorting\n\n";
    printArray(arr,n);

    sort(arr, arr + n);

    cout<<"\n\nElements in Array after Sorting\n\n";
    printArray(arr,n);

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a dark theme. In the top-left corner, there's an 'Activities' button and a 'Terminal' window icon. The terminal window is open and displays the following content:

```

Activities Terminal ▾
Jan 9 20:34 17.1.d.7.cpp
~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V07 ( Sort an Array )$ ls
17.1.d.7.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V07 ( Sort an Array )$ g++ 17.1.d.7.cpp
.vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V07 ( Sort an Array )$ ./a.out

Elements in Array before Sorting
2 7 5 1 8 3 3 4

Elements in Array after Sorting
1 2 3 3 4 5 7 8
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V07 ( Sort an Array )$ []

```

The code editor window is visible on the left, showing the source code for '17.1.d.7.cpp'. The code implements a bubble sort algorithm to sort an array of integers.

```

1 //7. Sort an Array
2
3 #include <bits/stdc++.h>
4 using namespace std;
5
6
7 void printArray(int a[], int n)
8 {
9     for (int i = 0; i < n; i++)
10    cout << a[i] << " ";
11    cout << '\n';
12 }
13
14 int main()
15 {
16     // creating a Array
17     int arr[] = {2, 7, 5, 1, 8, 3, 3, 4};
18
19     int n = sizeof(arr) / sizeof(arr[0]);
20
21     cout<<"\n\nElements in Array before Sorting\n\n";
22     printArray(arr,n);
23
24     sort(arr, arr + n);
25
26     cout<<"\n\nElements in Array after Sorting\n\n";
27     printArray(arr,n);
28
29     return 0;
30 }
31
32
33
34

```

Task 17.1.d.8:**Source Code:**

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

void printArray(int a[], int n)
{
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";
    cout << '\n';
}

int main()
{
    // creating a Array
    int arr[] = {2, 7, 5, 1, 8, 3, 3, 4};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout<<"\n\nElements in Array\n\n";
    printArray(arr,n);

    cout << "Using Binary Search for finding 50 in an
Array : \n ";

    if (binary_search(arr, arr + n, 50))
    {
        cout << "50 Exist in the Array \n";
    }
    else
    {
        cout << "50 Doesn't Exist in the Array\n";
    }

    return 0;
}
```

Output

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Code Editor (Left):

```

1 //8. Binary Search in an Array
2
3 #include <bits/stdc++.h>
4 using namespace std;
5
6 void printArray(int a[], int n)
7 {
8     for (int i = 0; i < n; i++)
9         cout << a[i] << " ";
10    cout << '\n';
11 }
12
13 int main()
14 {
15     // creating a Array
16     int arr[] = {2, 7, 5, 1, 8, 3, 3, 4};
17     int n = sizeof(arr) / sizeof(arr[0]);
18
19     cout << "\n\nElements in Array\n\n";
20     printArray(arr, n);
21
22     cout << "Using Binary Search for finding 50 in an Array : \n";
23
24     if (binary_search(arr, arr + n, 50))
25     {
26         cout << "50 Exist in the Array \n";
27     }
28     else
29     {
30         cout << "50 Doesn't Exist in the Array\n";
31     }
32 }
33
34
35 return 0;
36 }
37
38
39

```

Terminal (Right):

```

vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V08 ( Binary Search in an Array )$ ls
17.1.d.8.cpp
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V08 ( Binary Search in an Array )$ g++ 17.1.
d.8.cpp
.vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V08 ( Binary Search in an Array )$ ./a.out

Elements in Array
2 7 5 1 8 3 3 4
Using Binary Search for finding 50 in an Array :
50 Doesn't Exist in the Array
vishu@VICKY:~/Documents/OOPS in C++/Practical 17 ( Standard Template Libraries ( STL ) )/Task V04 ( Algorithm )/Task V08 ( Binary Search in an Array )$ 

```

The terminal output shows the execution of the program, which prints the elements of the array and then performs a binary search for the value 50, correctly identifying that it does not exist in the array.