



C & C++ PROGRAMMING

CODE: TECH 101 to 115
STUDENT HANDOUT

TABLE OF CONTENTS

Sl. No.	Code	Topic	Pg. No
1	Tech 101	Introduction to C Programming	3 - 5
2	Tech 102	Conditional Statements	6 - 11
3	Tech 103	Control Statements	12 - 15
4	Tech 104	Functions	16 - 18
5	Tech 105	Arrays	19 - 22
6	Tech 106	Pointers	23 - 25
7	Tech 107	Strings	26 - 28
8	Tech 108	Introduction to C++	29 - 32
9	Tech 109	Basics of OOP	33 - 34
10	Tech 110	Inheritance	35 - 37
11	Tech 111	Polymorphism	38 - 41
12	Tech 112	Abstraction, Encapsulation, Exception	42 - 44
13	Tech 113	Structures and Unions	45 - 48
14	Tech 114	Dynamic Memory Allocation	49 - 51
15	Tech 115	STL (Standard Template Library)	52 - 53

MODULE 1 – TECH 101 - INTRODUCTION TO C PROGRAMING

THEORY CONCEPTS:

1. Introduction, Variables and Data types, Console IO Operations
2. Preprocessor Directives and Macros, Operators and Expressions

PROGRAM 1:

Ramu and Somu are going on a picnic. Ramu packs m apples, n oranges. Somu packs $m1$ more apples than Ramu and $n1$ more oranges than Ramu. If Somu eats x of his apples and Ramu eats y of Somu's oranges, how many apples and oranges are left in total?

Input Format

Input consists of 6 integers $m, n, m1, n1, x, y$.

m - corresponds to the apples brought by Ramu.

n - corresponds to the oranges brought by Ramu.

$m1$ – Apples brought by Somu.

$n1$ – Oranges brought by Somu.

x - apples ate by Somu.

y – oranges ate by Ramu.

Output Format

Print two integers representing remaining apples and oranges.

Sample Input

4 3 2 8 3 2

Sample Output

7 12



PROGRAM 2:

Pranav and Change

Pranav, an enthusiastic kid, visited the "Fun Fair 2017" along with his family. His father wanted him to purchase entry tickets from the counter for his family members. Being a little kid, he is just learning to understand units of money. Pranav has paid some amount of money for the tickets but he wants your help to give him back the change of Rs. N using the minimum number of rupee notes.

Consider a currency system in which there are notes of seven denominations, namely, Rs. 1, Rs. 2, Rs. 5, Rs. 10, Rs. 50, Rs. 100. If the change is given to Pranav Rs. N is input, write a program to computer smallest number of notes that will combine to give Rs. N.

Input Format

The first line of the input is an integer N, the change to be given to Pranav.

Output Format

The output should display the smallest number of notes that will combine to give N.

Refer sample input and output for formatting specifications.

Constraints

Do not use any looping structures and conditional statements .

Sample Input

1200

Sample Input

242

Sample Output

12

Sample Output

7



PROGRAM 3:

Wisconsin State Fair

Wisconsin State Fair is one of the largest midsummer celebrations in the Midwest Allis, showcasing the agriculture skills and prowess of the state. The Event organisers hired few part-time employees to work at the fair and the agreed salary paid to them are as given below:

Weekdays --- 80 / hour

Weekends --- 50 / hour

Justin is a part-time employee working at the fair. Number of hours Justin has worked in the weekdays is 10 more than the number of hours he had worked during weekends. If the total salary paid to him in this month is known, write a program to estimate the number of hours he had worked during weekdays and the number of hours he had worked during weekends.

Input Format

First line of the input is a double value that corresponds to the total salary paid to Justin.

Output Format

First line of the output should display the number of hours Justin has worked during the weekdays.

Second line of the output should display the number of hours Justin has worked during the weekends.

Refer sample input and output for formatting specifications.

Sample Input

2750

Sample Output

Number of weekday hours is 25

Number of weekend hours is 15



MODULE 2 – TECH 102 - CONDITIONAL STATEMENTS

THEORY CONCEPTS:

1. if, if-else, if-else-if, nested if , switch-case statement

PROGRAM 1:

An automobile company manufactures both a two wheeler (TW) and a four wheeler (FW). A company manager wants to make the production of both types of vehicle according to the given data below:

- 1st data, Total number of vehicle (two-wheeler + four-wheeler)=v
- 2nd data, Total number of wheels = W

The task is to find how many two-wheelers as well as four-wheelers need to manufacture as per the given data.

Input format:

The candidate has to write the code to accept two positive numbers separated by a new line.

- First Input line – Accept value of V.
- Second Input line- Accept value for W.

Output format:

- Written program code should generate two outputs, each separated by a single space character

Example :

Sample Input

200-> Value of V

540 -> Value of W

Sample Output

TW =130 FW=70



PROGRAM 2:

Lucky Winner

It was the inaugural ceremony of "Fantasy Kingdom" Amusement park and the park Management has announced some lucky prizes for the visitors on the first day. Based on this, the visitors whose ticket number has the last digit as 3 or 8, are declared as lucky winners and attracting prizes are awaiting to be presented for them.

Write a program to find if the last digit of the ticket number of visitors is 3 or 8.

Input Format

First line of the input is an integer that corresponds to the ticket number.

Output Format

Output should display as "Lucky Winner" if the last digit of the ticket number is 3 or 8. Otherwise print "Not a Lucky Winner".

Sample Input

43

Sample Input

41

Sample Output

Lucky Winner

Sample Output

Not a Lucky Winner



PROGRAM 3:

Triangle Game

The Westland Game Fair is the premier event of its kind for kids interested in some intellectual and cognitive brain games. Exciting games were organized for kids between age group of 8 and 10. One such game was called the "Triangle game", where different number boards in the range 1 to 180 are available. Each kid needs to select three number boards, where the numbers on the boards correspond to the angles of a triangle.

If the angles selected by a kid forms a triangle, he/she would receive Prize 1. If the angles selected by a kid forms a right triangle, he/she would receive Prize 2. If the angles selected by the kids form an equilateral triangle, he/she would receive Prize 3. If the angles selected by a kid do not form even a triangle, then he/she will not receive any prizes. Write a program for the organizers to fetch the result based on the number boards selected by the kids.

Input Format

There are 3 lines in the input, each of which corresponds to the numbers on the boards that the kids select.

Output Format

Output should display "Prize 1" or "Prize 2" or "Prize 3" or "No Prize" based on the conditions given.

Sample Input

60

50

70

Sample Output

Prize 1

Sample Input

60

60

70

Sample Output

No prize



PROGRAM 4:

The Parking Lot

In a busy city, a parking lot charges different fees based on the type of vehicle. The manager wants an automated system where customers enter the vehicle type, and the system calculates the fee.

For Car: \$20

For Bike: \$10

For Bus: \$30

The system asks the customer to choose their vehicle type from a list and then displays the parking fee.

Sample Input : Enter vehicle type (1 for Car, 2 for Bike, 3 for Bus): 1

Sample Output: Parking Fee: \$20

Write a program that uses a switch statement to determine the parking fee based on the vehicle type.



ADDITIONAL QUESTION:

PROGRAM 5:

Card Game

The Westland Game Fair is the premier event of its kind for kids interested in some intellectual and cognitive brain games. Alan, a middle school boy is visiting the fair where he is very much drawn by the Card game.

The game's rules are:

A player needs to pick 3 cards from a big lot of cards. There are 4 types of Cards namely Spade(S), Heart(H), Club(C) and Diamond (D). If all the 3 cards that the player picks are of the same type and same number, they get a Double Bonanza. If all the 3 cards are of the same type or if they all have the same number, they get a Bonanza. Otherwise they do not get a Bonanza. Alan has now picked 3 cards and is awaiting to know if he has got a bonanza. Please help him to know if he has won the Bonanza or not.

Input Format

There are 3 lines of input.

Each of the line consists of character and integer input, which corresponds to the type of the card and the number in it that Alan picked. The type of card and the number are separated by a single spa

Output Format

Output should display "Double Bonanza" or "Bonanza" or "No Bonanza" based on the conditions given.

Input

H 5

C 3

D 3

Output

No Bonanza



Input

H 5

H 4

H 7

Output

Bonanza

Input

D 8

D 8

D 8

Output

Double Bonanza

MODULE 3 – TECH 103 - CONTROL STATEMENTS

THEORY CONCEPTS:

1. for loop, while loop, do-while loop, break and continue statements
2. Nested loops

PROGRAM 1:

Candy Game

Mona set off with great zeal to the "Fun Fair 2017". There were numerous activities in the fair, though Mona liked the Candy game. Delicious candies were wrapped in colourful foiled sheets with some random numbers on each of the candies. The game coordinators then formed many groups of few candies together, such that each candy group makes an integer and hid them all around the room. The objective of the game is that the players should look for the occurrences of number four anywhere in the integers (candy groups) placed in the room.

Mona started off with the game where there are many such integers, for each of them she should calculate the number of occurrences of the digit 4 in the decimal representation. Can you please help her in succeeding in the game?

Input Format

The only line of input contains a single integer from the candy group.

Output Format

Output should contain the number of occurrences of the digit 4 in the respective integer from the candy groups that Mona gets.



Sample Input

447474

Sample Input

12

Sample Output

4

Sample Output

0

PROGRAM 2:

The online math course provided 'MathAtTip' has designed a course for children called Learning Number Recognition and Counting. The assessment part of the course has a question where the student is given a number and a digit. The student needs to find out the total count of the digits present in the number excluding the given digit.

Write an algorithm to help the student find out the count of the total number of digits present in the number excluding the given digit.

Example**Input**

5644456 5

Output

5

Explanation

Excluding 5: the digits in the numbers are 4 and 6 and their total count is 5. Hence the output is 5.



PROGRAM 3:**Exploring Special Numbers**

In a maths club, students are exploring interesting properties of numbers. One student comes across a unique class of numbers known as special numbers. A two-digit number is classified as a special number if the sum of its digits and the product of its digits equals the number itself.

To make this exploration interactive, the teacher gives the students a challenge:

Task:

Write a program to identify all special numbers between two specified limits, m and n (both inclusive), where m and n are two-digit numbers.

Sample Input: 34 59

Sample output : 59



MODULE 4 – TECH 104 - FUNCTIONS

THEORY CONCEPTS:

1. Defining and calling functions, Function prototypes and declarations, Passing arguments to functions: pass by value, pass by reference
2. Recursion

PROGRAM 1:

In a computer science class, a teacher named Sachin is conducting a workshop focused on the principles of modular programming. To reinforce these concepts, she assigns her students to create a simple calculator program using a single user-defined function to perform arithmetic operations.

During the workshop, Sachin instructs the students to write a program that allows users to perform basic arithmetic operations (addition, subtraction, multiplication, and division) by calling a single user-defined function that takes two numbers and an operator as input.

Input: num1 = 10, num2 = 5, op = +

Output: res = 15.00

PROGRAM 2:

In a high school computer science class, the students are tasked with exploring the mathematical concept of the Fibonacci sequence and its connection to the Golden Ratio. Their teacher, Mrs. Carter, encourages them to think critically about recursion and its applications in problem-solving.

Mrs. Carter assigns the students to write a program that calculates the Fibonacci sequence using recursion.

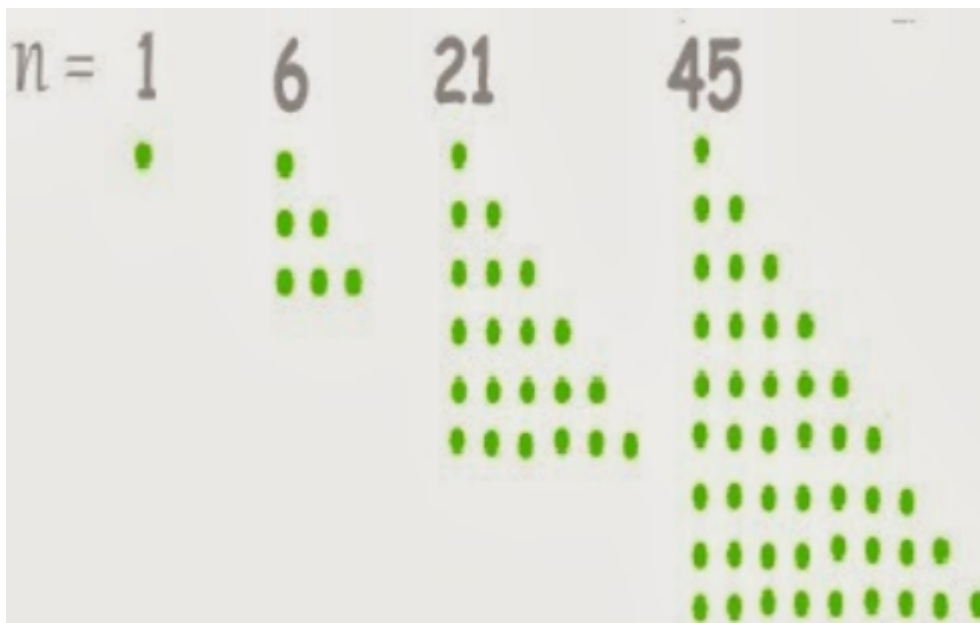
Input: Enter a positive integer (n): 7

Output: The 7th Fibonacci number is 13.



PROGRAM 3:

Right Triangle of Dots The much-awaited event in the entertainment industry every year is the "Screen Awards". This year the event is going to be organized on December 25 to honour the Artists for their professional excellence in Cinema. The Organizers of the event, J&R Events, decided to design the logo of the Screen Awards as a digitized image and display it on the LED panel boards for the show promotions across the venue. The Event team wanted to border the logo with right triangles which will describe it better. For this purpose, the Event development team is in the task to find if N dots can make a right triangle or not (all N dots must be used). Given N dots, we can make it look like a Right Triangle (45-45-90 triangle) exactly with N dots. Rearrange the given N dots, like this:



Your task is to help the team write a program using functions to find if N dots can make a right triangle or not.

Function Specifications: Use the function name, return type, and the argument type as: **int find(int)** The function must return 1 if you can make a right triangle using N dots, else return 0;

Input Format : The first line of the input consists of an integer N.



Output Format: Output "Yes" (without quotes) if you can make a right triangle using N dots, otherwise "No"(without quotes). Refer sample input and output for formatting specifications.

Sample Input

6

Sample Output

We can create Right Triangle of dots with 6 dots

Sample Input

4

Sample Output

We can't create Right Triangle of dots with 4 dots



MODULE 5 – TECH 105 - ARRAYS

THEORY CONCEPTS:

1. Arrays: declaration, initialization, accessing elements
2. Multi-dimensional arrays, Arrays as function arguments

PROGRAM 1:

A party has been organised on cruise. The party is organised for a limited time(T). The number of guests entering (E[i]) and leaving (L[i]) the party at every hour is represented as elements of the array. The task is to find the maximum number of guests present on the cruise at any given instance within T hours.

Example

Input :

5 -> Value of T

[7,0,5,1,3] -> E[], Element of E[0] to E[N-1], where input each element is separated by new line

[1,2,1,3,4] -> L[], Element of L[0] to L[N-1], while input each element is separated by a new line.

Output :

8 -> Maximum number of guests on cruise at an instance.



PROGRAM 2:

Airport security officials have confiscated several items of the passenger at the security checkpoint. All the items have been dumped into a huge box(array). Each item possessed a certain amount of risk(0,1,2). Here is the risk severity of the item representing an array[] of N number of integer values. The risk here is to sort the item based on their level of risk values range from 0 to 2.

Example 1:**Input:**

7 ----- Value of N

[1,0,2,0,1,0,2] -> Element of arr[0] to arr[N-1], while input each element is separated by new line

Output:

0 0 0 1 1 2 2



PROGRAM 3:

Mid Aged

The Pan Am 73 flight from Bombay to New York en route Karachi and Frankfurt was hijacked by a few Palestinian terrorists at the Karachi International Airport. The senior flight purser Neerja Banhot withered her fear and helped evacuate the passengers on board.

Neerja very well knew that she would not be able to evacuate all passengers dodging the hijackers. So she wanted to hand over the responsibility of evacuating the senior citizens(above 60 years of age) and children(below 18 years of age) in the flight to the mid-aged passengers seated in the diagonals. Given n the number of rows of seats and the number of seats in a row and the ages of passengers in each seat can you find the number of mid-aged passengers seated in the main diagonals?

Input Format

The first line of input consists of an integer n , corresponding to the number of rows of seats and the number of seats in the aircraft. The next n lines of input consist of n integers that correspond to the ages of passengers.

Output Format

The output consists of an integer corresponding to the number of mid-aged passengers seated in the diagonals.

Sample Input

```
3
21 3 44
78 25 19
50 23 6
```

Sample Output

```
2
```



PROGRAM 4:

A common problem in statistics is that of generating frequency distribution of the given data. Assuming that the data consists of n positive integers in the range 1 to 25, write a program that prints the number of times each integer occurs in the data.

Input Format

The first line of the input consists of the value of n.

The next n inputs are the array elements.

Output Format

The output prints the frequency of each data.

Sample Input

```
8
10 20 20 10 10 20 5 20
```

Sample Output

```
10 3
20 4
5 1
```



MODULE 6 – TECH 106 - POINTERS

THEORY CONCEPTS:

1. Introduction to Pointers, Pointer Arithmetic, Pointers and Arrays, Array of Pointers, Pointers as Function Arguments, Function Returning pointers

PROGRAM 1:

Sarah is a young and ambitious investor who has been closely tracking the stock prices of a particular company. She knows that if she can buy the stock when the price is low and sell it when the price goes high, she can make a significant profit. However, she must ensure that the stock price she buys at comes **before** the price she sells at.

Sarah has recorded the stock prices for the past few days. Now, she needs to figure out the **maximum profit** she could have made during this period. She has a list of the prices for each day and wants to calculate the **largest difference** between any two prices, where the higher price comes after the lower one.

Input Format :

The first line of the input consists of the value of n.

The next n inputs are the array elements.

Output Format

The output prints the maximum difference in the array.

Sample Input

5

1 10 12 15 30

Sample Output

29



PROGRAM 2:

In a faraway futuristic city, **Technoville**, there are two enormous supercomputers, **Alpha** and **Omega**, that control the city's defence systems. To ensure that the city remains safe, the two supercomputers need to run the same program without any differences in their code execution. Each line of code is represented by a sequence of unique codes, and these codes need to match between both supercomputers.

You, the lead engineer, have been tasked with comparing the program sequences from **Alpha** and **Omega** to make sure they are identical. If all codes match, the city's defence systems will remain fully operational. However, if there is even a single mismatch in the codes, the defence systems will shut down, leaving the city vulnerable to attacks from cybercriminals.

Problem Statement:

Write a program that compares two arrays using **pointers** to check if they contain the same elements at the same index positions.

Input Format:

- The first integer corresponds to **n**, which is the number of program lines in each supercomputer (maximum value of **n** is 15).
- The next **n** integers correspond to the codes of the program running in **Alpha** (the first array).
- The next **n** integers correspond to the codes of the program running in **Omega** (the second array).

Output Format:

- Print **"yes"** if the two arrays (programs) are the same.
- Print **"no"** if the arrays (programs) are different.



Sample Input 1:

4

45 67 89 23

45 67 89 23

Sample Output 1:

Yes

Sample Input 2:

3

100 200 300

100 250 300

Sample Output 2:

No

MODULE 7 – TECH 107 - STRINGS

THEORY CONCEPTS:

1. Introduction to strings in C, String input and output, String manipulation functions, Handling strings using pointers

PROGRAM 1:

The MNC 'Softcomp' had a security breach recently and company officials have decided to change the system password. The system password is in string format tagged a-z or A-Z. To change the password the officials will simply convert the lowercase characters of the old password to uppercase, and uppercase characters of the old password to lowercase.

Write an algorithm to display the new password. If no such password is possible display null.

Example

Input: bowANDarrow

Output : BOWandARROW

Explanation

The lowercase characters are converted into uppercase characters and vice-versa.

Sample Input

bowANDarrow

Sample Output

BOWandARROW



PROGRAM 2:

An event management company has come up with a unique idea of printing their event tickets. Based on the ticket number combination (str1), the visitor is directed towards a particular class of audience. The task is to create a program/application to fetch the ticket number based on the following conditions:

Any occurrences of digits EF, 56 and G, & should be deleted

The characters EF should be in the same format.

Example 1**Input:**

4523EF58G -> Value of STR1

Output:

452358 -> After removal of characters

‘EF’ and ‘G’

Example 2:**Input:**

E12F35G58 -> Value of STR1

Output:

E12F3558 -> After removal of character ‘G’



PROGRAM 3:

Given a string S(input) consisting of * and #. The length of the string is variable. The task is to find the minimum number of * and # required to make it a valid string. The string is considered valid if the number of * and # are equal. The * and # can be at any position in the string.

Note: The output will be a positive or negative integer based on number of * and # in the input string.

(* > #) : Positive integer

(# > *) : Negative integer

(#=*): 0

Example 1:**Input**

###*** → Value of S

Output

0 ---> number of * and # are equal

Example 2:**Input**

###***# → Value of S

Output

-1 ---> number of # is more than *



MODULE 8 – TECH 108 – INTRODUCTION TO C++ PROGRAMMING**THEORY CONCEPTS:**

1. Differences between C and C++
2. Basic structure of a C++ program

PROGRAM 1:

Sanjay is working on a printer device he wants to print whatever is typed in the word. Please help him by creating a software program to store the input and print the output.

Input Format

Input consists of a string

Output Format

Print the scanned string.

Sample Input

Hi, Greetings of the day!

Sample Output

Hi, Greetings of the day!



PROGRAM 2:

A young coder named Alka is trying to build a tool that will help people quickly find the sum of the digits of their favourite numbers. To test the tool, Alka decides to write a C++ program that will take a number as input and return the sum of its digits.

Input Format

Input consists of an Integer

Output Format

Print the sum of the digits.

Sample Input

12345

Sample Output

15



PROGRAM 3:

Likitha and her friend Manoj are planning a school event where they need to synchronise the schedules of two activities. To make sure the events align perfectly, Likitha decides to write a C++ program to calculate the Least Common Multiple (LCM) of the two activity durations.

Input Format

Input consists of two integers

Output Format

Print the LCM.

Sample Input

12 18

Sample Output

36



PROGRAM 4:

Sonia is organising a sports event and needs to arrange the scores of participants. She writes a C++ program that allows her to sort the scores in ascending order for the first half and descending order for the remaining half, so she can display the rankings accordingly.

Input Format

Input consists of size and followed by elements.

Output Format

Print the array in required format.

Sample Input

8

5 4 6 3 7 2 8 1

Sample Output

1 2 3 4 8 7 6 5



MODULE 9 – TECH 109 – BASICS OF OOP

THEORY CONCEPTS:

1. Introduction to Object-Oriented Programming (OOP)
2. Classes and objects, Constructors and Destructors
3. “this” pointer and static members, access specifiers (public, private, protected)

PROGRAM 1:

Hari is a class teacher, he wants to collect the students' details. Please help him to complete the process by writing the code.

Define a class Student with attributes name (string) and age (integer). Implement the following member functions: setData(string n, int a): Sets the name and age of the student. displayData(): Displays the name and age of the student.

Write a main function to create an object of Student, input data from the user, and display the student's information.

Input Format

Input consists of two data types,

First is the name & next is the age.

Output Format

Students name and age has to be displayed

Sample Input

Akshay

19

Sample Output

Student Name: Akshay

Student Age: 19



PROGRAM 2:

David and Emily are car enthusiasts who love collecting vintage cars. To keep track of their collection, David writes a C++ program where each car is represented as an object with details about its brand and manufacturing year. They can quickly input new cars and display their collection details at any time.

Define a class Car with attributes brand (string) and year (integer). Implement the following member functions:

Constructor to initialise brand and year.

display Info(): Displays the brand and year of the car.

Input Format

Input consists of two data types,

First is the name & next is the age.

Output Format

Students name and age has to be displayed

Sample Input

Compile time input

Sample Output

Car 1 Details:

Car Brand: Ford Mustang

Manufacturing Year: 1967

Car 2 Details:

Car Brand: Chevrolet Camaro

Manufacturing Year: 1969



MODULE 10 – TECH 110 - INHERITANCE

THEORY CONCEPTS:

1. Introduction to Inheritance
2. Derived Class and Base Class
3. Types of inheritance with example codes
4. Constructor and Destructor in derived classes

PROGRAM 1:

Mia just opened a new savings account with her local bank. To keep track of her transactions, she decided to write a simple C++ program to manage her account. The program helps her deposit money and ensures she maintains a minimum balance whenever she withdraws funds. Create a class “BankAccount” with fields “accountNumber”, “balance” and a method “deposit()” that adds an amount to the balance.

Create a subclass “SavingsAccount” that extends “BankAccount” and adds a field “minimumBalance” and a method “withdraw()” that subtracts an amount from the balance.

Create an object of the “SavingsAccount” class and call the “deposit()” and “withdraw()” methods.

Input Format

Values for banking!

Output Format

Balance.

Sample Input & Output:

Enter the Account Number: 123456

Enter the initial balance: 10000





Enter the minimum balance: 5000

Account Number: 123456

Balance: ₹10000

Enter the amount for deposit: 7500

Deposited: ₹7500

New balance: ₹17500

Enter the amount to withdraw: 13000

Withdrawal failed: Insufficient funds. Minimum balance must be maintained.

Enter the amount to withdraw: 20000

Withdrawal failed: Insufficient funds. Minimum balance must be maintained.



PROGRAM 2:

Write a program to implement the following logic using inheritance.

Create a parent class and implement the fun method. In the method, get the individual digits of the entered number, store it in an array, and find their sum.

For example in case of 1234, the individual digits are 4,3,2,1 and the final sum $\rightarrow (4+3)+(4+2)+(4+1)+(3+2)+(3+1)+(2+1) = 30$.

Create the main class that inherits the parent class and call the fun method inside the parent function.

Input Format

The input consists of an integer.

Output Format

The output prints the final sum.

Constraints

Integers only.

Sample Input

1234

Sample Output

30



MODULE 11 – TECH 111 - POLYMORPHISM

THEORY CONCEPTS:

1. Types of Polymorphism: compile-time and runtime
2. Function overloading, Operator overloading, Virtual functions and dynamic polymorphism
3. Types of inheritance with example codes
4. Constructor and Destructor in derived classes

PROGRAM 1:

Sarah is a mechanical engineer working on the design of various shapes for her company's manufacturing process. She needs to calculate the volume of different objects, including cubes, rectangular boxes, and cylinders. To streamline the calculations, Sarah writes a C++ program that uses function overloading to compute the volume of these shapes with varying numbers of arguments.

Function Overloading with Arguments:

Problem: Write a program that demonstrates the use of function overloading with different numbers of arguments. Create overloaded volume() functions in a class Volume to calculate the volume of:

A cube (single parameter: side length)

A rectangular box (three parameters: length, width, height)

A cylinder (two parameters: radius, height with default height value)

Input Format

Compile time inputs

Output Format

Volumes of cube, rectangle & cylinder.



Sample Input

Compile time inputs

Sample Output

volume of the cube with side:3 is: 27

volume of the rectangle with length:4,width:5,height:6 is: 120

volume of the cylinder with radius:2 and height:6 is: 75.36

PROGRAM 2:

Loan Calculator

For the clients, Bank XYZ need to calculate an education loan and a gold loan. As a result, they built a loan class with a

loadCalc(). The amount, year, and interest percent are required inputs for calculating an education loan. The amount and interest

% are necessary inputs for calculating a gold loan. Apply method overloading to calculate total amount need to be paid for

interest x. Print the monthly amount need to be paid by the client.

Note:

Education loan doesn't have interest upto 4 years. So apply interest if the year is greater than 4, otherwise interest is 0%.

Example1

Input

1

1000

5

2

Output

Amount with interest 1020.0

Explanation

Year greater than 5 , So interest applied .2% of 1000 is 20. So total amount to be paid is $1000+20=1020$

Example2

Input

1

1000

4

2

Output

Amount with interest 1000.0



PROGRAM 3:**Dynamic Polymorphism**

Write a program to illustrate dynamic polymorphism, create two classes Vehicle and Motorbike. Motorbike inherits the Vehicle class.

Create a method move() in the base class that takes a string as input and prints them using Best as prefix. Override the method move() in the derived class that also takes a string as input and prints them using Good as prefix.(Overrides the method in base class).

Example**Input**

HeroHonda

Suzuki

Output

Best HeroHonda

Good Suzuki



MODULE 12 – TECH 112 - ABSTRACTION, ENCAPSULATION, AND EXCEPTION

THEORY CONCEPTS:

1. Abstract classes and pure virtual functions
2. Encapsulation: data hiding
3. Exception=>try, catch, throw, Exception Classes

PROGRAM 1:

Create class money with two attributes: int rupee int paisa

Include getters, setters, Create the main class and initialize the values for the data members.

Get two amounts and print their sum

Sample Input

50 85

42 65

Sample Output

93.50



PROGRAM 2:**Data Type Exception**

Get an integer input,

If the input is 1, Throw an Integer exception.

If the input is 2, Throw a Character exception.

If the input is 3, Throw a double exception.

Input Format

The input consists of an integer.

Output Format

The output prints the corresponding exception.

Refer sample output for formatting exception.

Sample Input

1

Sample Input

2

Sample Input

3

Sample Output

Integer exception caught.

Sample Output

Character exception caught.

Sample Output

Double exception caught.



PROGRAM 3:

We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.

Sample input:

Enter marks for Student A in 3 subjects (out of 100):

85 90 78

Enter marks for Student B in 4 subjects (out of 100):

88 76 92 81

Sample output:

Percentage of Student A: 84.3333%

Percentage of Student B: 84.25%



MODULE 13 – TECH 113 - STRUCTURES AND UNIONS

THEORY CONCEPTS:

1. Structs: definition, declaration, initialization, accessing members, nested structures.
2. Unions: definition, declaration, accessing members
3. Differences between structs and unions

PROGRAM 1:

In an online grocery shop, customers want to purchase multiple items. Create a structure to store the Item code, Brand name, Item Name, Quantity, Price of the product. Generate the Bill

number, Display the purchased product, name, amount and quantity, and the total bill amount.

- a. Write a function MESSAGE() to alert the customer with the product name if the rate of a product is more than Rs.1000.
- b. Write a function VOUCHER() to generate the voucher for Rs.200 if the bill amount is greater than Rs.10000.

Input Format

- The first line of the input consists of the value of n.
- Next n inputs consist of the item code, brand name, item name, quantity, and price of the product(per item).

Output Format

- The output prints a message if the final amount of the product is greater than 1000.
- The next line prints the bill amount (Rounded off to two decimal places).
- The last line prints whether the customer gets a voucher or not.
- Refer sample input and output for formatting specifications.

Sample Input

2

101 philsbury flour 10 55

102 dettol soap 50 25



soap costs more than 1000

Sample Output

1800

No voucher

PROGRAM 2:

Five workers from an organization want to join in a Sukanya Samriddhi Yojana scheme in the post office. Create a structure to store the details of the employees like Policyholder name, Policy number, Mobile number, Child Name, Child Age, Maturity period (Minimum Maturity Period of 5 Years), and Monthly deposit amount(Fixed amount). Every month they have to pay a

fixed amount in that account. Based on the policy number a worker wants to check the amount accumulated after 2.5 yrs. Display the details of the Policyholder, Scheme started month and

Year, Maturity Period and the Total amount accumulated.

Input Format

The first line of the input consists of the value of n.

Next n inputs consist of the policy holder's name, policy number, mobile number, child name, child age, maturity period, and deposit amount.

The last line consists of the policy number to be searched.

Output Format

The output prints the policy holder's details and accumulated amount else not found.

Refer sample input and output for formatting specification

Sample Input

2

Alice 123456 9876543210 sam 3 3 5000

Bob 235689 8569745825 zara 4 4 6000

123456

Sample Output

Alice 123456 9876543210 sam 3 3 5000

Amount accumulated after 2.5 years: 150000



PROGRAM 3:

Write a program that uses a structure to represent complex numbers in mathematics.

```
struct complex {  
    float real;  
    float imaginary;  
};
```

Perform the following operations on complex numbers.

- Addition of 2 complex numbers
- Subtraction of 2 complex numbers
- Multiplication of 2 complex numbers
- Division of complex numbers
- Modulus of a given complex number

Input Format

The input consists of the real and imaginary parts of two complex numbers separated by a space.

Output Format

The output prints the results of the arithmetic operations.

Refer sample input and output for formatting specifications.

Sample Input

1 2 3 4

Sample Output

Sum = 4 + 6i

Difference = -2 + -2i

Difference = -5 + 10i

Division = 0.44 + 0.08i



MODULE 14 – TECH 114 - DYNAMIC MEMORY ALLOCATION

THEORY CONCEPTS:

1. malloc, calloc, realloc, free, new, delete.

PROGRAM 1:

A group of friends is planning a trip and wants to allocate a certain amount of money for each activity. Create a program that dynamically allocates memory for the expenses of various activities during the trip. The program should prompt the user to enter the number of activities and the cost for each activity. After all inputs are provided, display the total expenses for the trip.

Input Format:

The first line of input consists of the value of n, the number of activities.

The next n inputs consist of the costs for each activity.

Output Format:

Print the total expenses for the trip.

Sample Input:

```
3
500
200
300
```

Sample output:

```
Total expenses: 1000
```



PROGRAM 2:

Implement a student attendance system using dynamic memory allocation. Create a structure for students that includes their ID, name, and attendance percentage. Allow the user to enter details for a dynamic number of students and display those with attendance below a certain threshold.

Input Format:

The first line consists of an integer n , the number of students.

The next n lines consist of the student ID, name, and attendance percentage.

Output Format:

Print the details of students with attendance below 75%.

Sample Input

```
5
101 John 80.5
102 Alice 72.0
103 Bob 65.5
104 Emma 78.0
105 David 50.0
```

Sample output:

Students with attendance below 75%:

```
ID: 102, Name: Alice, Attendance: 72.0%
ID: 103, Name: Bob, Attendance: 65.5%
ID: 105, Name: David, Attendance: 50.0%
```



PROGRAM 3:

Write a program that dynamically allocates two matrices and performs their addition. Create a function to display the resulting matrix.

Input Format:

The first line consists of two integers m and n, the number of rows and columns.

The next m lines consist of n integers for the first matrix followed by m lines for the second matrix.

Output Format:

Print the resulting matrix after addition

Sample Input:

```
2 3
1 2 3
4 5 6
7 8 9
10 11 12
```

Sample Output:

Resulting matrix after addition:

```
8 9 10
14 16 18
```



MODULE 15 – TECH 115 - STANDARD TEMPLATE LIBRARY (STL)

THEORY CONCEPTS:

1. Introduction to STL, Containers: vector.
2. List
3. Set

PROGRAM 1:

You are tasked with creating a contact book application. Each contact has a name, phone number, and email. The user should be able to perform several operations, such as adding new contacts, searching for a contact by name, deleting a contact, and listing all contacts sorted by name.

Question:

Design a structure for the contact.

Use a vector to manage the contacts dynamically.

Implement functions for:

Adding a new contact.

Searching for a contact by name.

Deleting a contact.

Listing all contacts in alphabetical order.

Write a main function that demonstrates all these functionalities.



PROGRAM 2:

You need to design a system to manage a collection of movies. Each movie has a title, director, and release year. The system should allow users to add movies, remove movies, and search for movies by title.

Question:

Define a structure for the movie.

Use a set to store the movies to ensure that each movie is unique based on its title.

Implement functions for:

Adding a new movie (ensure no duplicates).

Removing a movie by title.

Searching for a movie by title.

Listing all movies sorted by title.

Write a demonstration in the main function that showcases each feature.

