

COMPUTER PROGRAMMING (3+1 Credit Hours)

CSC-113

LAB JOURNAL



DEPARTMENT OF COMPUTER SCIENCE
BAHRIA UNIVERSITY, KARACHI, PAKISTAN

Student Name: MUHAMMAD HASAN

Enrolment Number: 02-135251-040

Class & Section: BS(IT) 1A

Fall/Spring: Spring

Year: 2025

Lab Day & Timing: Thursday 1430 - 1730

Course Teacher: Sir Marouf

Lab Engineer: Sir Bilal

BAHRIA UNIVERSITY KARACHI CAMPUS

Department Of Computer Science

Table of Contents

No.	TASKS	Pg. No.	Signature
Lab 1	Getting Started(Pseudo Code, Algorithms & Flow Chart)	4	
Lab 2	Variables and Data Types	12	
Lab 3	Decision Statements	17	
Lab 4	Loops	27	
Lab 5	Nested Loops	39	
Lab 6	Functions	45	
Lab 7	Functions Overloading and Recursion	61	
Lab 8	Arrays 1-D	77	
Lab 9	Arrays 2-D	87	
Lab 10	Pointers	94	
Lab 11	Structures	104	
Lab 12	Strings	118	
Lab 13	I/O Streams (File Handling)	124	



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

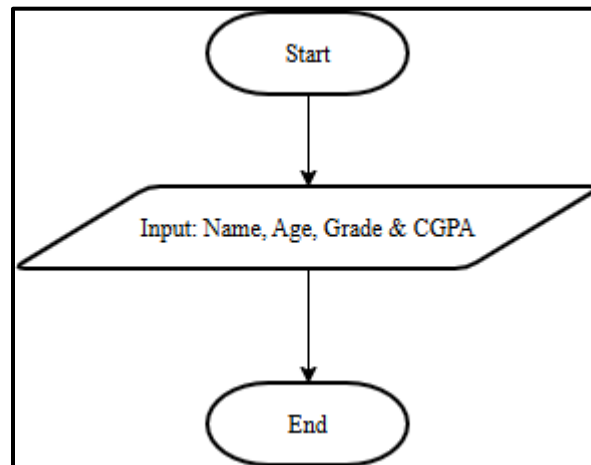
Lab 01 – Getting Started



Task 02:

Draw a flowchart & Write a C++ program to store statement record such as (Name, Age, Grade and CGPA).

Flowchart:



Code:

```
#include<iostream>
#include<string>
using namespace std;
int main(){
    string name;
    int age = 0;
    string grade;
    float cgpa = 0.00;
    cout << "Enter your name: ";
    getline(cin,name);
    cout << "Enter your age: ";
    cin >> age;
    cout << "Enter your grade: ";
    cin.ignore();
    getline(cin,grade);
    cout << "Enter your CGPA: ";
    cin >> cgpa;
}
```



Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

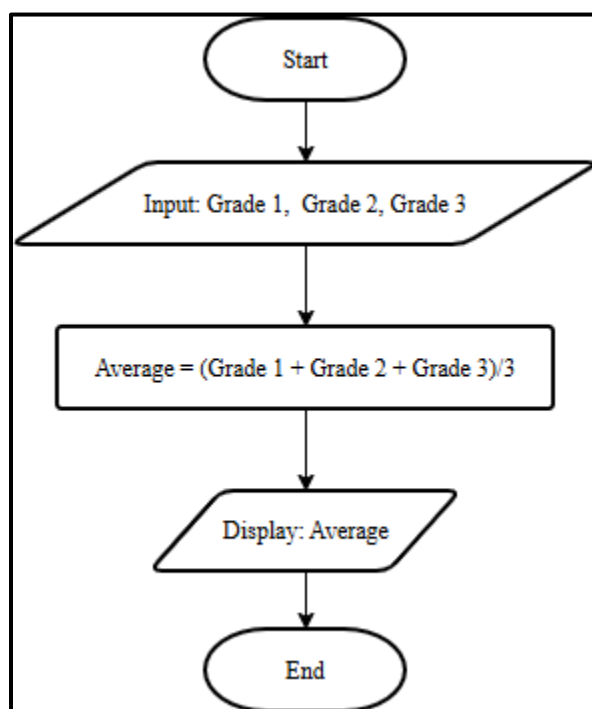
PS D:\Hasan\cpp\lab01> g++ task02.cpp
PS D:\Hasan\cpp\lab01> ./a.exe
Enter your name: Muhammad Hasan
Enter your age: 18
Enter your grade: A
Enter your CGPA: 4.00
PS D:\Hasan\cpp\lab01> 
```



Task 03:

Draw a flowchart and Write a C++ program to read student's three grade, calculate the average of the grade, and then display the average grade.

Flowchart:



Code:

```
#include<iostream>
using namespace std;
int main(){
    int g1 = 0, g2 = 0, g3 = 0; //g means Grade
    float avg = 0.00;
    cout << "Enter your first grade: ";
    cin >> g1;
    cout << "Enter your second grade: ";
    cin >> g2;
    cout << "Enter your third grade: ";
    cin >> g3;
    avg = (g1+g2+g3)/3.00; // calculating average
    cout << "Your average is: "<<avg;
}
```



Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

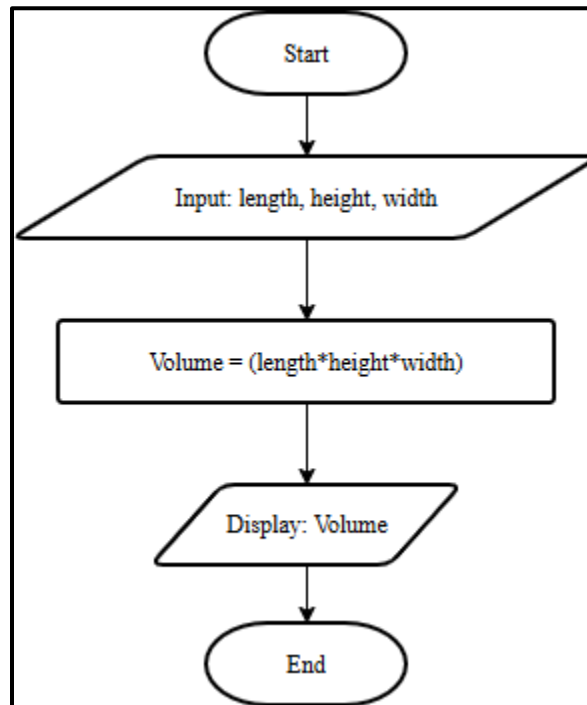
PS D:\Hasan\cpp\lab01> g++ task03.cpp
PS D:\Hasan\cpp\lab01> ./a.exe
Enter your first grade: 87
Enter your second grade: 88
Enter your third grade: 90
Your average is: 88.3333
PS D:\Hasan\cpp\lab01> 
```


Task 04:

Write program & draw a flowchart that reads the height, length, and width of the rectangular box, Calculates and displays the volume.

Note: Volume = lwh.

Flowchart:



Code:

```
#include<iostream>
using namespace std;
int main(){
    float length = 0.00, height = 0.00, width = 0.00, vol = 0.00;
    cout << "Enter height of rectangular box: ";
    cin >> height;
    cout << "Enter length of rectangular box: ";
    cin >> length;
    cout << "Enter width of rectangular box: ";
    cin >> width;
    vol = height*length*width;
    cout << "The volume of rectangular box is: "<<vol;
}
```



Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Hasan\cpp\lab01> g++ task04.cpp
PS D:\Hasan\cpp\lab01> ./a.exe
Enter height of rectangular box: 23
Enter length of rectangular box: 54.6
Enter width of rectangular box: 6
The volume of rectangular box is: 7534.8
PS D:\Hasan\cpp\lab01> 
```



Task 05:

Write a program to read the salary of an employee and prints particular designation of the concern person for example Manager Salary = 200k, Supervisor = 150k and Technician = 80k.

Code:

```
#include<iostream>
using namespace std;
int main(){
    int sal = 0;
    cout << "Enter the amount of your salary: ";
    cin >> sal;
    if(sal == 200){
        cout << "Your designation is Manager.";
    }
    else if(sal == 150){
        cout << "Your designation is Supervisor.";
    }
    else if(sal == 80){
        cout << "Your designation is Technician.";
    }
    else{
        cout << "Enter valid amount!";
    }
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter the amount of your salary: 100
Enter valid amount!
PS D:\Hasan\cpp\lab01> g++ task05.cpp
PS D:\Hasan\cpp\lab01> ./a.exe
Enter the amount of your salary: 200k
Your designation is Manager.
PS D:\Hasan\cpp\lab01>
```



Lab 02 – Variables and Data Types



Task 01:

Write a program that takes input as radius then calculate area of circle. (Hint: $A = \pi r^2$).

Code:

```
#include<iostream>
using namespace std;
int main(){
    float r = 0.00, a = 0.00, pi = 3.14;
    cout << "Enter radius of a circle: ";
    cin >> r;
    a = pi*(r*r);
    cout << "Area of this circle with radius '\"<<r<<\"' is: "<<a<<endl;
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

* History restored

PS D:\Hasan\cpp\lab02> g++ task01.cpp
PS D:\Hasan\cpp\lab02> ./a.exe
Enter radius of a circle: 22
Area of this circle with radius '22' is: 1519.76
PS D:\Hasan\cpp\lab02> 
```



Task 02:

Write a C++ program that reads a number in inches and converts it to meters. Note:
One inch is 0.0254 meter.

Code:

```
#include<iostream>
using namespace std;
int main(){
    float value_inch = 0.00, value_meter = 0.00;
    cout << "Enter number in inches: ";
    cin >> value_inch;
    value_meter = value_inch*0.0254;
    cout << "Value in meters is: "<<value_meter<<endl;
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter radius of a circle: 22
Area of this circle with radius '22' is: 1519.76
PS D:\Hasan\cpp\lab02> g++ task02.cpp
PS D:\Hasan\cpp\lab02> ./a.exe
Enter number in inches: 23
Value in meters is: 0.5842
PS D:\Hasan\cpp\lab02> 
```



Task 03:

Write a C++ program that prompt input roll number, student name and marks of three subjects:

1. Computer Programming = CP
2. Object Oriented Programming= OOP
3. Data Structures & Algorithms= DSA

Calculate total marks, percentage and division of student.

$$\text{Marks percentage} = \text{marks obtained} / \text{total} * 100$$

Code:

```
#include<iostream>
#include<string>
using namespace std;
int main(){
    string name;
    int rn = 0; // rn = roll number
    float cp = 0.00, oop = 0.00, dsa = 0.00, per = 0.00;
    cout << "Enter your name: ";
    getline(cin,name);
    cout << "Enter your roll number: ";
    cin >> rn;
    cout << "Enter your marks in Computer Programming CP: ";
    cin >> cp;
    cout << "Enter your marks in Object Oriented Programming OOP: ";
    cin >> oop;
    cout << "Enter your marks in Data Structures and Algorithms DSA: ";
    cin >> dsa;
    per = ((cp+oop+dsa)/(300))*100;
    cout << "Your percentage is: "<<per<<endl;
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter your name: Muhammad Hasan
Enter your roll number: 040
Enter your marks in Computer Programming CP: 98
Enter your marks in Object Oriented Programming OOP: 89
Enter your marks in Data Structures and Algorithms DSA: 98
Your percentage is: 95
PS D:\Hasan\cpp\lab02> |
```



Task 04:

Write a program that reads a temperature in degrees Celsius and prints out the corresponding temperature in degrees Fahrenheit ($F = (C * 9 / 5) + 32$).

Code:

```
#include<iostream>
using namespace std;
int main(){
    float temp_cel = 0.00, temp_fer = 0.00;
    cout << "Enter temperature in Celcius: ";
    cin >> temp_cel;
    temp_fer = ((temp_cel*9/5) + 32);
    cout << "Temperature in Fahrenheit is: "<<temp_fer<<endl;
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter your marks in Data Structures and Algorithms DSA: 98
Your percentage is: 95
PS D:\Hasan\cpp\lab02> g++ task04.cpp
PS D:\Hasan\cpp\lab02> ./a.exe
Enter temperature in Celcius: 434
Temperature in Fahrenheit is: 813.2
PS D:\Hasan\cpp\lab02> 
```




Lab 03 – Decision Statements



Task 01: Character Classification

Write a C++ program that checks whether a given character is a vowel, consonant, or a digit. The program should prompt the user to enter a character and then display a message indicating the classification.

Code:

```
#include<iostream>
using namespace std;
int main(){
    char a;
    cout << "Enter any character: ";
    cin >> a;
    if(a=='a' || a=='A' || a=='e' || a=='E' || a=='i' || a=='I' || a=='o' ||
a=='O' || a=='u' || a=='U'){
        cout << "\"<a<<\"' is a vowel."<<endl;
    }
    else if(a>='0' && a<='9'){
        cout << "\"<a<<\"' is a digit."<<endl;
    }
    else{
        cout << "\"<a<<\"' is consonent."<<endl;
    }
}
```



Output 1:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

* History restored

PS D:\Hasan\cpp\lab03> g++ task01.cpp
PS D:\Hasan\cpp\lab03> ./a.exe
Enetr any character: 2
'2' is a digit.
PS D:\Hasan\cpp\lab03> |
```

Output 2:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enetr any character: 2
'2' is a digit.
PS D:\Hasan\cpp\lab03> g++ task01.cpp
PS D:\Hasan\cpp\lab03> ./a.exe
Enetr any character: a
'a' is a vowel.
PS D:\Hasan\cpp\lab03> |
```



Task 02: Shopping Cart

You are developing an online shopping application for a retail store. The store wants to offer discounts to customers based on their total purchase amount. If the total purchase amount exceeds 15000 Rupees, a 10% discount will be applied to the customer's order.

Write a C++ program that prompts the user to enter the total purchase amount. Based on the entered amount, apply the appropriate discount using if-else statements and display the discounted amount to the customer.

Code:

```
#include<iostream>
using namespace std;
int main(){
    int ta = 0; //ta -> total amount
    float fa = 0.00; //fa -> final amount
    cout << "Enter the total amount of your purchased items: ";
    cin >> ta;
    if(ta>15000){
        cout << "Congragulations! you are getting a 10% discount."<<endl;
        fa = ta*0.9; //if getting 10% discount, means he's 90% amount
        cout << "Your discounted amount is: "<<fa<<endl;
    }
    else{
        fa = ta;
        cout << "Your total amount is: "<<fa<<endl;
    }
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

'a' is a vowel.
PS D:\Hasan\cpp\lab03> g++ task02.cpp
PS D:\Hasan\cpp\lab03> ./a.exe
Enter the total amount of your purchased items: 20000
Congragulations! you are getting a 10% discount.
Your discounted amount is: 18000
PS D:\Hasan\cpp\lab03> 
```



Task 03: Age Analysis

Assume that three friends are planning to do zip line adventure. Where age will be evaluated first to send the younger one first. Write a C++ program where compare age of 3 friends. Find the youngest one from all three.

Code:

```
#include<iostream>
using namespace std;
int main(){
    int age_fr1 = 0, age_fr2 = 0, age_fr3 = 0;
    cout << "Enter age of 1st friend: ";
    cin >> age_fr1;
    cout << "Enter age of 2nd friend: ";
    cin >> age_fr2;
    cout << "Enter age of 3rd friend: ";
    cin >> age_fr3;
    if(age_fr1<age_fr2 && age_fr1<age_fr3){
        cout << "1st friend is youngest, with age of "<<age_fr1<<" years."<<endl;
    }
    else if(age_fr2<age_fr1 && age_fr2<age_fr3){
        cout << "2nd friend is youngest, with age of "<<age_fr2<<" years."<<endl;
    }
    else{
        cout << "3rd friend is youngest, with age of "<<age_fr3<<" years."<<endl;
    }
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Hasan\cpp\lab03> g++ task03.cpp
PS D:\Hasan\cpp\lab03> ./a.exe
Enter age of 1st friend: 33
Enter age of 2nd friend: 44
Enter age of 3rd friend: 12
3rd friend is youngest, with age of 12 years.
PS D:\Hasan\cpp\lab03> |
```



Task 04: Medical Diagnosis System

Write a C++ program that takes the user's input for body temperature and uses if-else statements to determine and display a basic medical diagnosis such as "Hypothermia," "Normal Body Temperature," "Low-Grade Fever," or "High Fever" based on different temperature ranges. Students can run the program with various temperature inputs to observe the different diagnoses.

Temp less than 36.5	Hypothermia
Temp in between 36.5 and 37.5	Normal Body Temperature
Temp in between 37.6 and 38.5	Low grade Fever
More than 38.5	High Fever

Code:

```
#include<iostream>
using namespace std;
int main(){
    float temp = 0.00;
    cout << "Enter your body temperature: ";
    cin >> temp;
    if(temp<36.5){
        cout << "You are diagnosed with Hypothemia."<<endl;
    }
    else if(temp>=36.5 && temp<=37.5){
        cout << "You temperature is Normal."<<endl;
    }
    else if(temp>=37.6 && temp<=38.5){
        cout << "You are diagnosed with Low Grade Fever."<<endl;
    }
    else{
        cout << "You are diagnosed with High Fever."<<endl;
    }
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter your body temperature: 23
You are diagnosed with Hypothemia.
PS D:\Hasan\cpp\lab03> g++ task04.cpp
PS D:\Hasan\cpp\lab03> ./a.exe
Enter your body temperature: 87
You are diagnosed with High Fever.
PS D:\Hasan\cpp\lab03> 
```



Task 05:

Imagine you are developing a program for a small business that sells handmade crafts. The owner recently bought a beautiful vase at a cost price (CPrice) and is trying to sell it at a selling price (SPrice). Your task is to create a program that will help the owner determine whether they made a profit, or incurred a loss.

Task Requirements:

- a) Prompt the user to enter the Cost Price (CPrice) of the vase.
- b) Prompt the user to enter the Selling Price (SPrice) of the vase.

Profit Calculation:

$$\text{Profit} = \text{SPrice} - \text{Cprice}$$

- c) Print a message displaying the profit amount to the user.

Loss Calculation:

$$\text{Loss} = \text{CPrice} - \text{Sprice}$$

- d) Print a message displaying the loss amount to the user.
- e) If CPrice is equal to SPrice, output a message indicating that there is no profit or loss.

Code:

```
#include<iostream>
using namespace std;
int main(){
    int cp = 0, sp = 0; // cp -> cost price & sp -> selling price
    cout << "Enter the cost price of vase: ";
    cin >> cp;
    cout << "Enter the selling price of vase: ";
    cin >> sp;
    if(sp>cp){
        int profit = 0;
        profit = sp-cp;
        cout << "You have made a profit of "<<profit<<" rupees."<<endl;
    }
    else if(cp>sp){
        int loss = 0;
        loss = cp-sp;
        cout << "You have made a loss of "<<loss<<" rupees."<<endl;
    }
    else{
        cout << "Selling price is same as cost price."<<endl;
    }
}
```



Output 1:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

You are diagnosed with High Fever.
PS D:\Hasan\cpp\lab03> g++ task05.cpp
PS D:\Hasan\cpp\lab03> ./a.exe
Enter the cost price of vase: 45
Enter the selling price of vase: 5
You have made a loss of 40 rupees.
PS D:\Hasan\cpp\lab03> 
```

Output 2:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

You have made a loss of 40 rupees.
PS D:\Hasan\cpp\lab03> g++ task05.cpp
PS D:\Hasan\cpp\lab03> ./a.exe
Enter the cost price of vase: 89
Enter the selling price of vase: 2000
You have made a profit of 1911 rupees.
PS D:\Hasan\cpp\lab03> 
```




Task 06:

Write a C++ program that takes a digit (0-9) as input from the user and print its spelling using a switch statement.

Code:

```
#include<iostream>
using namespace std;
int main(){
    int dig = 0;
    cout << "Enter any digit between 0 & 9: ";
    cin >> dig;
    switch(dig){
        case 0:
            cout << "Its spelling is \'Zero\'"<<endl;
            break;
        case 1:
            cout << "Its spelling is \'One\'"<<endl;
            break;
        case 2:
            cout << "Its spelling is \'Two\'"<<endl;
            break;
        case 3:
            cout << "Its spelling is \'Three\'"<<endl;
            break;
        case 4:
            cout << "Its spelling is \'Four\'"<<endl;
            break;
        case 5:
            cout << "Its spelling is \'Five\'"<<endl;
            break;
        case 6:
            cout << "Its spelling is \'Six\'"<<endl;
            break;
        case 7:
            cout << "Its spelling is \'Seven\'"<<endl;
            break;
        case 8:
            cout << "Its spelling is \'Eight\'"<<endl;
            break;
        case 9:
            cout << "Its spelling is \'Nine\'"<<endl;
            break;
```



```
        default:  
        cout << "Invalid input!"<<endl;  
        break;  
    }  
}
```

Output 1:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
  
Enter the selling price of vase: 2000  
You have made a profit of 1911 rupees.  
PS D:\Hasan\cpp\lab03> g++ task06.cpp  
PS D:\Hasan\cpp\lab03> ./a.exe  
Enter any digit between 0 & 9: 3  
Its spelling is 'Three'  
PS D:\Hasan\cpp\lab03> |
```

Output 2:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
  
Its spelling is 'Three'  
PS D:\Hasan\cpp\lab03> g++ task06.cpp  
PS D:\Hasan\cpp\lab03> g++ task06.cpp  
PS D:\Hasan\cpp\lab03> ./a.exe  
Enter any digit between 0 & 9: 0  
Its spelling is 'Zero'  
PS D:\Hasan\cpp\lab03> |
```



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

Lab 04 - Loops



Task 01:

/*A child is learning about the alphabet and wants to explore how letters are represented in computers using ASCII values. To make the learning experience fun and interactive, the child's parent decides to create a simple C++ program using for loop that prints ASCII values and their equivalent characters for the uppercase and lowercase in English alphabet.
e.g. (A to Z).*/

```
#include<iostream>
using namespace std;
int main(){
    for(int i = 97; i <= 122; i++){
        cout << "ASCII value "<<i<<". Corresponding character
"<<char(i)<<". "<<endl; //done it only for lowercase alphabets
    }
}
```



Output:

```
PROBLEMS  OUTPUT  TERMINAL  ...  powershell + - [ ] [X] ... < X
PS D:\Hasan\cpp\university\lab04> g++ task01.cpp
PS D:\Hasan\cpp\university\lab04> ./a.exe
ASCII value 97. Corresponding character a.
ASCII value 98. Corresponding character b.
ASCII value 99. Corresponding character c.
ASCII value 100. Corresponding character d.
ASCII value 101. Corresponding character e.
ASCII value 102. Corresponding character f.
ASCII value 103. Corresponding character g.
ASCII value 104. Corresponding character h.
ASCII value 105. Corresponding character i.
ASCII value 106. Corresponding character j.
ASCII value 107. Corresponding character k.
ASCII value 108. Corresponding character l.
ASCII value 109. Corresponding character m.
ASCII value 110. Corresponding character n.
ASCII value 111. Corresponding character o.
ASCII value 112. Corresponding character p.
ASCII value 113. Corresponding character q.
ASCII value 114. Corresponding character r.
ASCII value 115. Corresponding character s.
ASCII value 116. Corresponding character t.
ASCII value 117. Corresponding character u.
ASCII value 118. Corresponding character v.
ASCII value 119. Corresponding character w.
ASCII value 120. Corresponding character x.
ASCII value 121. Corresponding character y.
ASCII value 122. Corresponding character z.
PS D:\Hasan\cpp\university\lab04> [ ]
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} C++ [] [X] Go Live windows-gcc-x86 []



Task 02:

/*Inventory Management

You work for a small retail store that needs a program to manage its inventory. The store wants a program that allows them to enter the quantity of each item in stock and calculates the total value of the inventory. Each item has an associated price.

Write a C++ program by implementing loop concept on this task

Requirements

```
| Prompt the user to enter the number of different items in stock.  
| For each item, prompt the user to enter the price per unit and the quantity  
in stock.  
| Calculate and print the total value of the inventory.*/
```

```
#include<iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    int total_items = 0, qt = 0;
```

```
    float price = 0.00, total_price = 0.00, final_price = 0.00;
```

```
    cout << "Enter the total number of products you have purchased: ";
```

```
    cin >> total_items;
```

```
    for(int i = 1; i <= total_items; i++){
```

```
        cout << "Enter the price of your item no. "<<i<<": ";
```

```
        cin >> price;
```

```
        cout << "Enter the quantity of your item no. "<<i<<": ";
```

```
        cin >> qt;
```

```
        cout << "-----"<<endl;
```

```
        total_price = (price*qt);
```

```
        final_price = final_price + total_price;
```

```
    }
```

```
    cout << "Final payable amount is: "<<final_price;
```

```
}
```



Output:

```
PS D:\Hasan\cpp\university\lab04> g++ task02.cpp
PS D:\Hasan\cpp\university\lab04> ./a.exe
Enter the total number of products you have purchased: 4
Enter the price of your item no. 1: 10
Enter the quantity of your item no. 1: 2
-----
Enter the price of your item no. 2: 400
Enter the quantity of your item no. 2: 1
-----
Enter the price of your item no. 3: 34
Enter the quantity of your item no. 3: 2
-----
Enter the price of your item no. 4: 90
Enter the quantity of your item no. 4: 5
-----
Final payable amount is: 938
PS D:\Hasan\cpp\university\lab04> |
```



Task 03:

```
/*Prime Checker
A group of students is tasked with creating a C++ project titled "Prime Number
Addition." The goal is to input 10 integers provided by the user, and the program
will then identify prime numbers among these inputs and calculate their sum.
Using a for loop, the program will prompt the user to input each integer
individually. After gathering the inputs, it will calculate the sum of the prime
numbers and display the result*/

#include<iostream>
using namespace std;
int main(){
    int n = 0;
    cout << "Enter number: ";
    cin >> n;
    int count = 0;
    for(int i = 1; i <= n; i++){
        if(n%i == 0){
            count++;
        }
    }
    if(count == 2){
        cout << "It is a prime number."<<endl;
    }
    else{
        cout << "It is not a prime number."<<endl;
    }
}
```

Output:

```
PS D:\Hasan\cpp\university\lab04> ./a.exe
Enter number: 7
It is a prime number.
PS D:\Hasan\cpp\university\lab04> ./a.exe
Enter number: 2
It is a prime number.
PS D:\Hasan\cpp\university\lab04> |
```




Task 04:

```
/*GCD HCF FINDER

As part of a school project, you're developing a mathematical utility program in C++. One of the essential features you aim to implement is finding the greatest common divisor (GCD), also referred to as the highest common factor (HCF), of two given numbers.

This functionality is designed to aid students in comprehending the concept of common divisors between two numbers using C++ and a while loop.*/

#include<iostream>
using namespace std;
int main(){
    int n1 = 0, n2 = 0, hcf = 0;
    cout << "Enter first number: ";
    cin >> n1;
    cout << "Enter second number: ";
    cin >> n2;
    int i = 1;
    while(i <= n1 && i <= n2){
        if(n1%i == 0 && n2%i == 0){
            hcf = i;
        }
        i++;
    }
    cout << "The HCF is: "<<hcf<<endl;
}
```

Output:

```
PS D:\Hasan\cpp\university\lab04> g++ task04.cpp
PS D:\Hasan\cpp\university\lab04> ./a.exe
Enter first number: 13
Enter second number: 16
The HCF is: 1
PS D:\Hasan\cpp\university\lab04> ./a.exe
Enter first number: 24
Enter second number: 12
The HCF is: 12
PS D:\Hasan\cpp\university\lab04> ./a.exe
Enter first number: 40
Enter second number: 73
The HCF is: 1
PS D:\Hasan\cpp\university\lab04> 
```



Task 05:

```
/* Academic Excellence Awards Ceremony

At an Academic Excellence Awards Ceremony, the organizers want to recognize
students' achievements in mathematics by calculating the factorial of their
scores.
The factorial value will be used to determine the complexity level of problems
they solved.*/

#include<iostream>
using namespace std;
int main(){
    int total_stu = 0;
    cout << "Enter no. of students: ";
    cin >> total_stu;
    cout << endl;
    int count = 1;
    while(count <= total_stu){
        int marks = 0, fact = 1;
        cout << "Enter marks of student no. "<<count<<": ";
        cin >> marks;
        for(int i = 1; i <= marks; i++){
            fact = fact*i;
        }
        cout << "Factorial of marks is: "<<fact<<endl;
        cout << endl;
        count++;
    }
}
```

Output:

```
PS D:\Hasan\cpp\university\lab04> g++ task05.cpp
PS D:\Hasan\cpp\university\lab04> ./a.exe
Enter no. of students: 3

Enter marks of student no. 1: 15
Factorial of marks is: 2004310016

Enter marks of student no. 2: 10
Factorial of marks is: 3628800

Enter marks of student no. 3: 8
Factorial of marks is: 40320

PS D:\Hasan\cpp\university\lab04> |
```



Task 06:

```
/*Multiplication Visualization

To enhance the multiplication concept of your students you decide to make program
which help then to visualize the how multiplication is done. Write C++
programming language using do-while loop for visualization of multiplication of
table of any number enter by user*/

#include<iostream>
using namespace std;
int main(){
    int num = 0, lim = 0;
    cout << "Enter number of which you want to see table: ";
    cin >> num;
    cout << "Enter number till which you want to multiplication: ";
    cin >> lim;
    for(int i = 1; i <= lim; i++){
        cout<<num <<" * "<<i<<" = "<<num*i<<endl;
    }
}
```

Output:

```
PS D:\Hasan\cpp\university\lab04> g++ task06.cpp
PS D:\Hasan\cpp\university\lab04> ./a.exe
Enter number of which you want to see table: 6
Enter number till which you want to multiplication: 9
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54
PS D:\Hasan\cpp\university\lab04> |
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Go Live windows-gcc-x86



Task 07:

```
/*Employee Payroll Calculation
```

In a small company, you've been tasked with developing a C++ program to calculate the payroll for employees.

Each employee has a unique hourly wage and works a different number of hours per week.

Your program needs to calculate the weekly pay for each employee based on their hourly wage and hours worked.*/

```
#include<iostream>
using namespace std;
int main(){
    int total_emp = 0;
    cout << "Enter total number of employees: ";
    cin >> total_emp;
    for(int i = 1; i <= total_emp; i++){
        int hrs = 0, sal = 0, final_sal = 0;
        cout << endl;
        cout << "Employee no. "<<i<<endl;
        cout << "Enter hourly salary: ";
        cin >> sal;
        cout << "Enter hours worked: ";
        cin >> hrs;
        final_sal = hrs*sal;
        cout << endl;
        cout << "Salary of employee no. "<<i<<" is "<<final_sal<<endl;
        cout << endl;
    }
}
```



Output:

```
PS D:\Hasan\cpp\university\lab04> g++ task07.cpp
```

```
PS D:\Hasan\cpp\university\lab04> ./a.exe
```

```
Enter total number of employees: 4
```

```
Employee no. 1
```

```
Enter hourly salary: 4000
```

```
Enter hours worked: 9
```

```
Salary of employee no. 1 is 36000
```

```
Employee no. 2
```

```
Enter hourly salary: 55
```

```
Enter hours worked: 56
```

```
Salary of employee no. 2 is 3080
```

```
Employee no. 3
```

```
Enter hourly salary: 44
```

```
Enter hours worked: 44
```

```
Salary of employee no. 3 is 1936
```

```
Employee no. 4
```

```
Enter hourly salary: 2000
```

```
Enter hours worked: 18
```

```
Salary of employee no. 4 is 36000
```

```
PS D:\Hasan\cpp\university\lab04> 
```



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

Lab 05 – Nested Loops



Task 01:

```
// 1. Filled stars
// 2. Hollow stars
// 3. Pyramid

#include<iostream>
using namespace std;
int main(){
    int choice = 0;
    cout << "Welcome to star pattern generator."<<endl;
    cout << "Choose the type of star pattern you want to print."<<endl;
    cout << "1. Filled Stars"<<endl;
    cout << "2. Hollow Stars"<<endl;
    cout << "3. Pyramid"<<endl;
    cout << "Enter your choice: ";
    cin >> choice;

    //Filled stars (L)
    if(choice == 1){
        int rows = 0;
        cout << "Enter no. of rows: ";
        cin >> rows;
        for(int i = 1; i <= rows; i++){
            for(int j = 1; j <= i; j++){
                cout << "*";
            }
            cout << endl;
        }
    }

    //Hollow L shape
    else if(choice == 2){
        int rows = 0;
        cout << "Enter no. of rows: ";
        cin >> rows;
        for(int i = 1; i <= rows; i++){
            for(int j = 1; j <= i; j++){
                if(i == 1 || i == rows || j == 1 || j == i){
                    cout << "*";
                }
                else{
                    cout << " ";
                }
            }
        }
    }
```



```
    }  
    cout << endl;  
}  
}  
  
//normal pyramid  
else if(choice == 3){  
    int rows = 0;  
    cout << "Enter no. of rows: ";  
    cin >> rows;  
    for(int i = 1; i <= rows; i++){  
        for(int j = rows; j >= i; j--){  
            cout << " ";  
        }  
        for(int k = 1; k <= i; k++){  
            cout << "* ";  
        }  
        cout << endl;  
    }  
}  
  
else{  
    cout << "Invalid choice"<<endl;  
}  
}
```




Output:

```
PS D:\Hasan\cpp\university\lab05> g++ task01.cpp
PS D:\Hasan\cpp\university\lab05> ./a.exe
Welcome to star pattern generator.
Choose the type of star pattern you want to print.
1. Filled Stars
2. Hollow Stars
3. Pyramid
Enter your choice: 1
Enter no. of rows: 6
*
**
***
****
*****
*****
```

```
PS D:\Hasan\cpp\university\lab05> g++ task01.cpp
PS D:\Hasan\cpp\university\lab05> ./a.exe
Welcome to star pattern generator.
Choose the type of star pattern you want to print.
1. Filled Stars
2. Hollow Stars
3. Pyramid
Enter your choice: 2
Enter no. of rows: 5
*
**
* *
* *
*****
```

```
PS D:\Hasan\cpp\university\lab05> g++ task01.cpp
PS D:\Hasan\cpp\university\lab05> ./a.exe
Welcome to star pattern generator.
Choose the type of star pattern you want to print.
1. Filled Stars
2. Hollow Stars
3. Pyramid
Enter your choice: 3
```

```
Enter your choice: 3
Enter no. of rows: 5
*
* *
* * *
* * * *
* * * * *
```

```
PS D:\Hasan\cpp\university\lab05> |
```



Task 02:

```
//Floyd's triangle

#include<iostream>
using namespace std;
int main(){
    int rows = 0, p = 1; //p is for number on pattern
    cout << "Enter no. of rows: ";
    cin >> rows;
    for(int i = 1; i <= rows; i++){
        for(int j = 1; j <= i; j++){
            cout << p<<" ";
            p++;
        }
        cout << endl;
    }
}
```

Output:

```
PS D:\Hasan\cpp\university\lab05> g++ task02.cpp
PS D:\Hasan\cpp\university\lab05> ./a.exe
Enter no. of rows: 5
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
PS D:\Hasan\cpp\university\lab05> |
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Go Live windows-gcc-x86



Task 03:

```
// Decreasing L
// Hollow square
// Combination of L & square

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

    // for inverted triangle
    cout << "1. Inverted triangle."<<endl;
    for(int i = 1; i <= 5; i++){
        for(int j = 5; j >= i; j--){
            cout << "* ";
        }
        cout << endl;
    }

    //for hollow square
    cout << "2. Hollow square."<<endl;
    for(int i = 1; i <= 5; i++){
        for(int j = 1; j <= 5; j++){
            if(i == 1 || i == 5 || j == 1 || j == 5){
                cout << "* ";
            }
            else{
                cout << "  ";
            }
        }
        cout << endl;
    }

    cout << "3. Combination of L & square."<<endl;
    for(int i = 1; i <= 5; i++){
        for(int j = 1; j <= i; j++){ // for simple L
            cout << "* ";
        }
        for(int k = 5; k > i; k--){ // for spaces after L
            cout << "  ";
        }
        for(int k = 1; k <= 5; k++){ // for square
            if(i == 1 || i == 5 || k == 1 || k == 5){
                cout << "* ";
            }
        }
    }
```



```
        else{  
            cout << "  ";  
        }  
    }  
    cout << endl;  
}  
}
```

Output:

```
PS D:\Hasan\cpp\university\lab05> g++ task03.cpp  
PS D:\Hasan\cpp\university\lab05> ./a.exe  
1. Inverted triangle.  
* * * * *  
* * * *  
* * *  
* *  
*  
  
2. Hollow square.  
* * * * *  
*       *  
*       *  
*       *  
* * * * *  
  
3. Combination of L & square.  
*       * * * * *  
* *       *       *  
* * *       *       *  
* * * * * *       *  
* * * * * * * * * *
```

PS D:\Hasan\cpp\university\lab05>

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Go Live windows-gcc-x86



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

Lab 06 – Functions



Task 01:

```
/*Basic Calculator

You are creating a basic calculator program for students to practice arithmetic
operations. The calculator should allow users to perform addition, subtraction,
multiplication, division, and power on two numbers entered via user input. Each
operation will be executed using separate functions*/

#include<iostream>
using namespace std;

void addition(float a, float b){
    cout << "Result: "<<a<<" + "<<b<<" = "<<(a+b)<<endl;
}
void subtraction(float a, float b){
    cout << "Result: "<<a<<" - "<<b<<" = "<<(a-b)<<endl;
}
void multiply(float a, float b){
    cout << "Result: "<<a<<" * "<<b<<" = "<<(a*b)<<endl;
}
void divide(float a, float b){
    if(b==0){
        cout << "Error: division by zero is not possible."<<endl;
    }
    else{
        cout << "Result: "<<a<<" / "<<b<<" = "<<(a/b)<<endl;
    }
}
void power(float a, float b){
    float result = 1;
    float temp = b; //b will ultimately become 0, but i have to write actual
value of b in result statement
    while(b != 0){
        result = result*a;
        b--;
    }
    cout << "Result: "<<a<<" ^ "<<temp<<" = "<<result<<endl;
}

int main(){
    float num1 = 0, num2 = 0;
    char op;
    cout << "***Simple Calculator***"<<endl;
```



```
cout << "Enter first number: ";
cin >> num1;
cout << "Enter second number: ";
cin >> num2;
cout << "Enter operator (+, -, *, /, ^) ";
cin >> op;
if(op == '+'){
    addition(num1, num2);
}
else if(op == '-'){
    subtraction(num1, num2);
}
else if(op == '*'){
    multiply(num1, num2);
}
else if(op == '/'){
    divide(num1, num2);
}
else if(op == '^'){
    power(num1, num2);
}
else{
    cout << "invalid Operator!"<<endl;
}
}
```



Output:

```
PROBLEMS OUTPUT TERMINAL ... powershell + v [icon] [icon] ... < X
PS D:\Hasan\cpp\university\lab06> g++ task01.cpp
PS D:\Hasan\cpp\university\lab06> ./a.exe
***Simple Calculator***
Enter first number: 2.5
Enter second number: 2.5
Enter operator (+, -, *, /, ^) +
Result: 2.5 + 2.5 = 5
PS D:\Hasan\cpp\university\lab06> g++ task01.cpp
PS D:\Hasan\cpp\university\lab06> ./a.exe
***Simple Calculator***
Enter first number: 67
Enter second number: 60
Enter operator (+, -, *, /, ^) -
Result: 67 - 60 = 7
PS D:\Hasan\cpp\university\lab06> ./a.exe
***Simple Calculator***
Enter first number: 50.9
Enter second number: 2
Enter operator (+, -, *, /, ^) *
Result: 50.9 * 2 = 101.8
PS D:\Hasan\cpp\university\lab06> ./a.exe
***Simple Calculator***
Enter first number: 2
Enter second number: 2
Enter operator (+, -, *, /, ^) ^
Result: 2 ^ 2 = 4
PS D:\Hasan\cpp\university\lab06> 6
6
PS D:\Hasan\cpp\university\lab06> ./a.exe
***Simple Calculator***
Enter first number: 6
Enter second number: 0
Enter operator (+, -, *, /, ^) /
Error: division by zero is not possible.
PS D:\Hasan\cpp\university\lab06> [ ]

Ln 26, Col 22 Spaces: 4 UTF-8 CRLF {} C++ [icon] [icon] Go Live windows-gcc-x86 [icon]
```




Task 02:

/*ATM System

Create a simple ATM system that allows users to check their balance, deposit funds, and withdraw funds. Use functions to perform each operation and use if-else ladders to validate user input and ensure that withdrawals do not exceed the available balance. Use loops to allow multiple transactions until the user chooses to exit.*/

```
#include<iostream>
```

```
using namespace std;
```

```
void deposit(float &totalBalance){
```

```
    float amount = 0.00;
```

```
    cout << "Enter amount you want to deposit: ";
```

```
    cin >> amount;
```

```
    totalBalance += amount;
```

```
    cout << endl;
```

```
    cout << "Your new balance is: "<<totalBalance<<endl;
```

```
    cout << endl;
```

```
}
```

```
void withdraw(float &totalBalance){
```

```
    float amount = 0.00;
```

```
    cout << "Enter amount you want to withdraw: ";
```

```
    cin >> amount;
```

```
    if(amount>totalBalance){
```

```
        cout << endl;
```

```
        cout << "Insufficient balance."<<endl;
```

```
        cout << endl;
```

```
    }
```

```
    else{
```

```
        totalBalance -= amount;
```

```
        cout << endl;
```

```
        cout << "Your new balance is: "<<totalBalance<<endl;
```

```
        cout << endl;
```

```
    }
```

```
}
```

```
void checkbalance(float &totalBalance){
```

```
    cout << endl;
```

```
    cout << "Your total balance is: "<<totalBalance<<endl;
```

```
    cout << endl;
```

```
}
```



```
int main(){
    float totalBalance = 0.00;
    int choice = 0;
    cout << "Welcome to Bank"<<endl;
    do{
        cout << "*****Menu*****"<<endl;
        cout << "To deposit amount, press '1' "<<endl;
        cout << "To withdraw amount, press '2' "<<endl;
        cout << "To check balance, press '3' "<<endl;
        cout << "To exit, press '4' "<<endl;
        cout << "Enter choice: ";
        cin >> choice;
        if(choice == 1){
            deposit(totalBalance);
        }
        else if(choice == 2){
            withdraw(totalBalance);
        }
        else if(choice == 3){
            checkbalance(totalBalance);
        }
        else if(choice == 4){
            cout << endl;
            cout << "Ending program with total balance: "<<totalBalance<<endl;
        }
        else{
            cout << "invalid choice."<<endl;
        }
    }while(choice!=4);
}
```



Output:

```
PS D:\Hasan\cpp\university\lab06> g++ task02.cpp
```

```
PS D:\Hasan\cpp\university\lab06> ./a.exe
```

```
Welcome to Bank
```

```
*****Menu*****
```

```
To deposit amount, press '1'
```

```
To withdraw amount, press '2'
```

```
To check balance, press '3'
```

```
To exit, press '4'
```

```
Enter choice: 3
```

```
Your total balance is: 0
```

```
*****Menu*****
```

```
To deposit amount, press '1'
```

```
To withdraw amount, press '2'
```

```
To check balance, press '3'
```

```
To exit, press '4'
```

```
Enter choice: 1
```

```
Enter amount you want to deposit: 100
```

```
Your new balance is: 100
```

```
*****Menu*****
```

```
To deposit amount, press '1'
```

```
To withdraw amount, press '2'
```

```
To check balance, press '3'
```

```
To exit, press '4'
```

```
Enter choice: 2
```

```
Enter amount you want to withdraw: 80
```

```
Your new balance is: 20
```



Your new balance is: 20

*****Menu*****

To deposit amount, press '1'

To withdraw amount, press '2'

To check balance, press '3'

To exit, press '4'

Enter choice: 2

Enter amount you want to withdraw: 100

Insufficient balance.

*****Menu*****

To deposit amount, press '1'

To withdraw amount, press '2'

To check balance, press '3'

To exit, press '4'

Enter choice: 3

Your total balance is: 20

*****Menu*****

To deposit amount, press '1'

To withdraw amount, press '2'

To check balance, press '3'

To exit, press '4'

Enter choice: 4

Ending program with total balance: 20

PS D:\Hasan\cpp\university\lab06> |



Task 03:

```
/* Number Analysis

Write a C++ program that includes functions to find the least significant digit
and most significant digit of a three-digit number. Additionally, implement
functions to calculate the sum of digits and determine whether the number is a
palindrome*/

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

void plaindrome(char a, char c){
    if(a == c){
        cout << "It is a Palindromic number."<<endl;
    }
    else{
        cout << "It is not a Palindromic number."<<endl;
    }
}

int main(){
    int num = 0; //user will input an integer
    string numStr; //we will convert it into string
    char first, last;//to store first and last digit

    cout << "Enter a three digit number: ";
    cin >> num;

    numStr = to_string(num);//converting into string
    first = numStr[0];//extracting
    last = numStr[2];
    cout << "First digit is: "<<first<<endl;
    cout << "Last digit is: "<<last<<endl;

    plaindrome(first, last);
}
```



Output:

```
PS D:\Hasan\cpp\university\lab06> g++ task03.cpp
PS D:\Hasan\cpp\university\lab06> ./a.exe
Enter a three digit number: 300
First digit is: 3
Last digit is: 0
It is not a Palindromic number.
PS D:\Hasan\cpp\university\lab06> ./a.exe
Enter a three digit number: 343
First digit is: 3
Last digit is: 3
It is a Palindromic number.
PS D:\Hasan\cpp\university\lab06> |
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Go Live windows-gcc-x86



Task 04:

```
/*Personal Finance Tracker
```

Develop a C++ program for personal finance tracking. The program should allow users to record financial transactions, categorize expenses, calculate total expenses and income, and view their budget.

Instructions:

Present a menu with the following options:

1. Record Transaction: Users can record a transaction by entering the category and amount.
2. View Expenses: Display the total expenses recorded.
3. View Income: Display the total income recorded.
4. View Budget: Display the budget, calculated as the difference between total income and total expenses.
5. Exit: Terminate the program.

When recording a transaction, prompt the user to input the category and amount. Positive amounts denote income, while negative amounts denote expenses.

Categorize each transaction as income or expense based on the amount entered.

Display the category and amount of each transaction. Calculate and display the total expenses and income. Calculate and display the budget, which is the difference between total income and total expenses. Continue displaying the menu and prompting for options until the user chooses to exit.*/

```
#include<iostream>
```

```
using namespace std;
```

```
//function to add and subtract money
```

```
void addTransactions(float &income, float &expense){  
    float groceries = 0.00, utility = 0.00, loan = 0.00, temp_income = 0.00;  
    int choice = 0;  
    cout << "-----Add transaction ----- "<<endl;  
    cout << "Record Income. Press '1' "<<endl;  
    cout << "Record Groceries. Press '2' "<<endl;  
    cout << "Record Utility bills. Press '3' "<<endl;  
    cout << "Record Loan. Press '4' "<<endl;  
    cout << endl;  
    cout << "Enter choice: ";  
    cin >> choice;  
    cout << endl;  
    if(choice == 1){  
        cout << "Enter your income: ";  
        cin >> temp_income;  
        cout << "Recorded!"<<endl;  
    }  
}
```



```
else if(choice == 2){
    cout << "Enter your groceries expense: ";
    cin >> groceries;
    cout << "Recorded!"<<endl;
}
else if(choice == 3){
    cout << "Enter your utility bill expense: ";
    cin >> utility;
    cout << "Recorded!"<<endl;
}
else if(choice == 4){
    cout << "Enter your loan amount: ";
    cin >> loan;
    cout << "Recorded!"<<endl;
}
else{
    cout << "Invalid choice."<<endl;
}
income += temp_income;
expense += (groceries + utility + loan);
}

void viewExpense(float expense){
    cout << endl;
    cout << "Your total amount of expenses is: "<<expense<<endl;
    cout << endl;
}

void viewBudget(float income, float expense){
    float budget = income - expense;
    cout << endl;
    cout << "Your total income is: "<<income<<endl;
    cout << "Your total expenses are: "<<expense<<endl;
    cout << endl;
    cout << "Your available budget is: "<<budget<<endl;
    cout << endl;
}

int main(){
    float income = 0.00, expense = 0.00;
    int choice = 0;
    cout << "*****PERSONAL FINANCE TRACKER*****"<<endl;
    do{
        cout << "
                MENU
            " <<endl;
        cout << "To record transaction, Press '1' " <<endl;
```




```
cout << "To view expense, Press '2' "<<endl;
cout << "To view budget, Press '3' "<<endl;
cout << "To exit, Press '4' "<<endl;
cout << "Enter choice: ";
cin >> choice;
cout << endl;
if(choice == 1){
    addTransactions(income, expense);
}
else if(choice == 2){
    viewExpense(expense);
}
else if(choice == 3){
    viewBudget(income, expense);
}
else if(choice == 4){
    cout << "terminating program..."<<endl;
}
else{
    cout << "Invalid choice"<<endl;
}
}while(choice != 4);
}
```



Output:

```
PROBLEMS OUTPUT TERMINAL ... powershell + - [ ] [X] ... < X

PS D:\Hasan\cpp\university\lab06> g++ task04.cpp
PS D:\Hasan\cpp\university\lab06> ./a.exe
****PERSONAL FINANCE TRACKER****
-----MENU-----
To record transaction, Press '1'
To view expense, Press '2'
To view budget, Press '3'
To exit, Press '4'
Enter choice: 1

-----Add transaction-----
Record Income. Press '1'
Record Groceries. Press '2'
Record Utility bills. Press '3'
Record Loan. Press '4'

Enter choice: 1

Enter your income: 50000
Recorded!

-----MENU-----
To record transaction, Press '1'
To view expense, Press '2'
To view budget, Press '3'
To exit, Press '4'
Enter choice: 1

-----Add transaction-----
Record Income. Press '1'
Record Groceries. Press '2'
Record Utility bills. Press '3'
Record Loan. Press '4'

Enter choice: 2

Enter your groceries expense: 10000
Recorded!

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} C++ [ ] [X] Go Live windows-gcc-x86 [ ]
```



```
PROBLEMS OUTPUT TERMINAL ... powershell + v [icon] [icon] ... < X

Enter choice: 2

Enter your groceries expense: 10000
Recorded!
-----MENU-----
To record transaction, Press '1'
To view expense, Press '2'
To view budget, Press '3'
To exit, Press '4'
Enter choice: 1

-----Add transaction-----
Record Income. Press '1'
Record Groceries. Press '2'
Record Utility bills. Press '3'
Record Loan. Press '4'

Enter choice: 3

Enter your utility bill expense: 10000
Recorded!
-----MENU-----
To record transaction, Press '1'
To view expense, Press '2'
To view budget, Press '3'
To exit, Press '4'
Enter choice: 2

Your total amount of expenses is: 20000

-----MENU-----
To record transaction, Press '1'
To view expense, Press '2'
To view budget, Press '3'
To exit, Press '4'
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} C++ [icon] [icon] Go Live windows-gcc-x86 [icon]



```
Your total amount of expenses is: 20000
```

```
-----MENU-----
```

```
To record transaction, Press '1'
```

```
To view expense, Press '2'
```

```
To view budget, Press '3'
```

```
To exit, Press '4'
```

```
Enter choice: 3
```

```
Your total income is: 50000
```

```
Your total expenses are: 20000
```

```
Your available budget is: 30000
```

```
-----MENU-----
```

```
To record transaction, Press '1'
```

```
To view expense, Press '2'
```

```
To view budget, Press '3'
```

```
To exit, Press '4'
```

```
Enter choice: 4
```

```
terminating program...
```

```
PS D:\Hasan\cpp\university\lab06> |
```



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

Lab 07 – Function Overloading and Recursion



Task 01:

Write a recursive function for the calculation of power.

Code:

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

int power(int base, int exp){
    if(exp == 0) return 1;
    return base*power(base, exp - 1);
}

int main(){
    int base = 0, exp = 0;
    cout << "Enter base: ";
    cin >> base;
    cout << "Enter power: ";
    cin >> exp;

    cout << base << " to the power " << exp << " is: " << power(base, exp) <<
endl;
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task01.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Enter base: 3
Enter power: 4

3 to the power 4 is: 81
```

Task 02:**Number Comparison Utility**

You are designing a program to assist users in analyzing a set of numbers. The application should allow users to enter between two to four numbers and then determine the smallest and largest numbers among them. Additionally, users should be able to compare different combinations of numbers, such as comparing two, three, or four numbers simultaneously. This flexibility enables users to make quick comparisons and analyze data effectively

Code:

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

int findSmall(int a, int b){
    if(a <= b) return a;
    else return b;
}

int findSmall(int a, int b, int c){
    if(a <= b && a <= c) return a;
    else if(b <= a && b <= c) return b;
    else return c;
}

int findSmall(int a, int b, int c, int d){
    if(a <= b && a <= c && a <= d) return a;
    else if(b <= a && b <= c && b <= d) return b;
    else if(c <= a && c <= b && c <= d) return c;
    else return d;
}

int findLarg(int a, int b){
    if(a >= b) return a;
    else return b;
}

int findLarg(int a, int b, int c){
    if(a >= b && a >= c) return a;
    else if(b >= a && b >= c) return b;
    else return c;
}

int findLarg(int a, int b, int c, int d){
    if(a >= b && a >= c && a >= d) return a;
```



```
    else if(b >= a && b >= c && b >= d) return b;
    else if(c >= a && c >= b && c >= d) return c;
    else return d;
}

int main(){
    int a[4];

    cout << "Enter 4 numbers: " << endl;
    for(int i = 0; i < 4; i++) cin >> a[i];

    cout << endl;
    cout << "*****" << endl;
    cout << "Smallest among first two is: " << findSmall(a[0], a[1]) << endl;
    cout << "Largest among first two is: " << findLarg(a[0], a[1]) << endl;

    cout << "*****" << endl;
    cout << "Smallest among first three is: " << findSmall(a[0], a[1], a[2]) <<
endl;
    cout << "Largest among first three is: " << findLarg(a[0], a[1], a[2]) <<
endl;

    cout << "*****" << endl;
    cout << "Smallest among all is: " << findSmall(a[0], a[1], a[2], a[3]) <<
endl;
    cout << "Largest among all is: " << findLarg(a[0], a[1], a[2], a[3]) << endl;
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task02.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Enter 4 numbers:
56
34
66
3

*****
Smallest among first two is: 34
Largest among first two is: 56
*****
Smallest among first three is: 34
Largest among first three is: 66
*****
Smallest among all is: 3
Largest among all is: 66
```


Task 03:**Recursive Fibonacci number Calculation**

Imagine you are developing a program to calculate Fibonacci numbers for a mathematics class. Your task is to create a C++ program that allows students to input a positive number and calculates the corresponding Fibonacci number using a recursive function. Here's how the program works:

The program prompts the user to enter a positive number.

- The user inputs a positive number, and the program stores it in a variable called N.
- The program calls a recursive function called fib() to calculate the Fibonacci number for the input value of N.

The fib() function calculates the Fibonacci number using the following rules:

- For $n = 0$, the Fibonacci number is 0.
- For $n = 1$, the Fibonacci number is 1.
- For $n > 1$, the Fibonacci number is calculated as the sum of the Fibonacci numbers of the previous:

two terms: $\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$

Code:

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

int fib(int n){
    if(n == 0 || n == 1) return n;
    return fib(n-1) + fib(n-2);
}

int main(){
    int n = 0;
    cout << "Enter a positive number: ";
    cin >> n;

    cout << "Fibonacci no. for " << n << " is: " << fib(n) << endl;
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task03.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Enter a positive number: 5

Fibonacci no. for 5 is: 5
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Enter a positive number: 7

Fibonacci no. for 7 is: 13
```

Task 04:**Growth Simulator**

Write a C++ program that uses recursion to calculate the population size of rabbits after a specified number of months.

The program should prompt the user to enter the initial number of rabbit pairs and the number of months for which they want to simulate the growth.

Implement a recursive function to calculate the population size based on the following rules:

- Each pair of rabbits reproduces every month.
- Each pair produces exactly one new pair of offspring.
- Rabbits mature in one month and start reproducing.
- Assume no rabbits die during the simulation.
- After computing the population size, display the result to the user.
- Ensure that the program handles valid inputs and gracefully handles errors.

Code:

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

int growthSim(int month){
    if(month == 1 || month == 2) return 1;
    return growthSim(month - 1) + growthSim(month - 2);
}

int main(){
    int initialPairs = 0, month = 0;
    cout << "Welcome to the Rabbit population Growth Simulator" << endl << endl;

    cout << "Enter the initial number of rabbit pairs: ";
    cin >> initialPairs;
    cout << "Enter the number of months for simulation: ";
    cin >> month;

    cout << "After " << month << " months, Rabbits population will be: " <<
    initialPairs*growthSim(month) << endl;
}
```



Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task04.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Welcome to the Rabbit population Growth Simulator

Enter the initial number of rabbit pairs: 3
Enter the number of months for simulation: 4

After 4 months, Rabbits population will be: 9
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Welcome to the Rabbit population Growth Simulator

Enter the initial number of rabbit pairs: 2
Enter the number of months for simulation: 6

After 6 months, Rabbits population will be: 16
```

Task 05:

Finding the Maximum

Imagine you are developing a program to assist engineers in analyzing sensor data from three different sources. The program needs to identify the highest recorded values from each sensor to determine potential anomalies in the data. For this purpose, you decide to create a function that finds the maximum of three floating-point numbers, as well as find the maximum of two integer point numbers.

Code:

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

float findMax(float a, float b, float c){
    if(a >= b && a >= c) return a;
    else if(b >= a && b >= c) return b;
    else return c;
}

int findMax(int a, int b){
    if(a >= b) return a;
    else return b;
}

int main(){
    int choice = 0;
    float fVal[3];
    int intVal[2];

    do{
        cout << "1. Compare 3 float values." << endl;
        cout << "2. Compare 2 integer values." << endl;
        cout << "3. Exit." << endl;
        cout << "Enter choice: ";
        cin >> choice;

        if(choice == 1){
            cout << "Enter values: ";
            cin >> fVal[0] >> fVal[1] >> fVal[2];
            cout << "Largest among them is: " << findMax(fVal[0], fVal[1],
fVal[2]) << endl;
        }
        else if(choice == 2){
            cout << "Enter value: ";
```



```
        cin >> intVal[0] >> intVal[1];
        cout << "Largest among them is: " << findMax(intVal[0], intVal[1]) <<
endl;
    }
    else if(choice == 3) cout << "Closong program..." << endl;
    else cout << "Invalid choice!" << endl;

    }while(choice != 3);
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task05.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe

1. Compare 3 float values.
2. Compare 2 integer values.
3. Exit.
Enter choice: 1

Enter values: 45.5
45.6
32

Largest among them is: 45

2. Compare 2 integer values.
3. Exit.
Enter choice: 3

Closong program...
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task05.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe

1. Compare 3 float values.
2. Compare 2 integer values.
3. Exit.
Enter choice: 1

Enter values: 45.5
45.6
34

Largest among them is: 45.6
```



Largest among them is: 45.6

1. Compare 3 float values.
2. Compare 2 integer values.
3. Exit.

Enter choice: 2

Enter value: 3

78

Largest among them is: 78

1. Compare 3 float values.
2. Compare 2 integer values.
3. Exit.

Enter choice: 3

Closing program...

Task 06:

Calculating Area with Different Shapes

Picture yourself developing a geometry calculator program using function overloading. This application allows users to calculate the area of various shapes such as circles, rectangles, and triangles. Users can input the required dimensions for each shape, and the program intelligently selects the appropriate function to calculate and display the area.

Code:

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

float cirArea(float r){
    return (3.14*(r*r));
}

float recArea(float l, float w){
    return l*w;
}

float triArea(float b, float h){
    return (1/2)*b*h;
}

int main(){
    int choice = 0;

    do{
        cout << endl;
        cout << "1. Calculate area of circle." << endl;
        cout << "2. Calculate area of rectangle." << endl;
        cout << "3. Calculate area of triangle." << endl;
        cout << "4. Exit." << endl;
        cout << "Enter choiec: ";
        cin >> choice;

        switch(choice){
            case 1:{
                cout << endl;
                float radius = 0.00;
                cout << "Enter radius: ";
                cin >> radius;
                cout << "Area of circle is: " << cirArea(radius) << endl;
                break;
            }
        }
    } while(choice < 4);
}
```



```
}

case 2:{
    cout << endl;
    float length = 0.00, width = 0.00;
    cout << "Enter length: ";
    cin >> length;
    cout << "Enter width: ";
    cin >> width;
    cout << "Area is: " << recArea(length, width) << endl;
    break;
}

case 3:{
    cout << endl;
    float base = 0.00, height = 0.00;
    cout << "Enter base: ";
    cin >> base;
    cout << "Enter height: ";
    cin >> height;
    cout << "Area is: " << triArea(base, height) << endl;
    break;
}

case 4: cout << "\nExiting program." << endl;
break;

default: cout << "\nInvalid choice!" << endl;
break;
}
}while(choice != 4);
}
```




Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task06.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
```

1. Calculate area of circle.
2. Calculate area of rectangle.
3. Calculate area of triangle.
4. Exit.

Enter choiec: 1

Enter radius: 45

Area of circle is: 6358.5

1. Calculate area of circle.
2. Calculate area of rectangle.
3. Calculate area of triangle.
4. Exit.

Enter choiec: 3

Enter base: 5

Enter height: 6

Area is: 0

1. Calculate area of circle.
2. Calculate area of rectangle.
3. Calculate area of triangle.
4. Exit.

Enter choiec: 4

Exiting program.

Task 07:

Fuel Efficiency Calculator

You're building a fuel efficiency calculator for a vehicle management system. The goal is to let users check fuel efficiency for cars, trucks, and airplanes using function overloading.

What You Have to Do:

Use function overloading to create separate fuel efficiency functions for: Cars, Trucks & Airplanes

Parameters by Vehicle Type:

Cars:

- int: engine displacement
- double: fuel consumption rate
- double: vehicle weight

Trucks:

- int: cargo weight
- float: engine power
- double: aerodynamic drag coefficient

Airplanes:

- string: aircraft type
- double: cruising altitude
- float: airspeed

User Interaction:

- Ask the user which vehicle they're calculating for
- Get inputs for the required parameters
- Calculate and display the fuel efficiency with correct units

Code:

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

float calEfficiency(int disp, double consump, double weight){
    return ((disp / 1000.0) * (100.0 / consump) / (weight / 1000.0));
}

float calEfficiency(int cargoWeight, float pow, double aeroDynamicDrag){
    return (pow * 0.8) / ((cargoWeight / 1000.0) * aeroDynamicDrag + 1.0);
}

float calEfficiency(string type, double altitude, float speed){
    return (speed * 0.1) / (altitude / 10000.0 + 1.0);
}
```



```
int main(){
    int choice = 0;
    do{
        cout << endl;
        cout << "1. Calculate efficiency of car." << endl;
        cout << "2. Calculate efficiency of truck." << endl;
        cout << "3. Calculate efficiency of Airplane." << endl;
        cout << "4. Exit." << endl;
        cout << "Enter choice: ";
        cin >> choice;
        if(choice == 1){
            int displacement = 0;
            double fuelCon = 0.00, weight = 0.00;
            cout << "\nEnter displacement: ";
            cin >> displacement;
            cout << "Enter fuel consumption: ";
            cin >> fuelCon;
            cout << "Enter vehicle weight: ";
            cin >> weight;
            cout << "\nFuel Efficiency is: " << calEfficiency(displacement,
fuelCon, weight) << endl;
        }
        else if(choice == 2){
            int cargoWeight = 0;
            float enginePow = 0.00;
            double dragCoefficient = 0.00;
            cout << "\nEnter Cargo weight: ";
            cin >> cargoWeight;
            cout << "Enter engine power: ";
            cin >> enginePow;
            cout << "Enter aerodynamic drag coefficient: ";
            cin >> dragCoefficient;
            cout << "\nFuel Efficiency is: " << calEfficiency(cargoWeight,
enginePow, dragCoefficient) << endl;
        }
        else if(choice == 3){
            string type;
            double altitude = 0.00;
            float speed = 0.00;

            cin.ignore();
            cout << "\nEnter aircraft type: ";
            getline(cin, type);
            cout << "Enter cruising altitude: ";
            cin >> altitude;
```



```
        cout << "Enter airspeed: ";  
        cin >> speed;  
  
        cout << "\nFuel Efficiency is: " << calEfficiency(type, altitude,  
speed) << endl;  
    }  
  
    else if(choice == 4) cout << "\nExting program." << endl;  
    else cout << "\nInvalid choice!" << endl;  
}while(choice != 4);  
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task07.cpp  
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
```

1. Calculate efficiency of car.
2. Calculate efficiency of truck.
3. Calculate efficiency of Airplane.
4. Exit.

Enter choice: 2

Enter Cargo weight: 45
Enter engine power: 887
Enter aerodynamic drag coefficient: 54

Fuel Efficiency is: 206.88

1. Calculate efficiency of car.
2. Calculate efficiency of truck.
3. Calculate efficiency of Airplane.
4. Exit.

Enter choice: 3

Enter aircraft type: Cargo
Enter cruising altitude: 6
Enter airspeed: 89

Fuel Efficiency is: 8.89466

1. Calculate efficiency of car.
2. Calculate efficiency of truck.
3. Calculate efficiency of Airplane.
4. Exit.

Enter choice: 4

Exting program.



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

Lab 08 – Arrays 1-D



Task 01:

Write a C program to convert an array into ascending order.

Code:

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

int main(){
    int a[5];

    for(int i = 0, sNo = 1; i < 5; i++, sNo++){
        cout << "Enter the element no. "<<sNo<<" of array: ";
        cin >> a[i];
    }

    cout << "Original array:" << endl;
    for(int i = 0; i < 5; i++){
        cout << a[i] << " ";
    }

    cout << endl;
    cout << "Sorted array in ascending order:" << endl;

    //sorting
    for(int i = 0; i < (5-1); i++){
        for(int j = 0; j < 5-1-i; j++){
            if(a[j] > a[j+1]){
                int temp = a[j+1];
                a[j+1] = a[j];
                a[j] = temp;
            }
        }
    }

    //displaying sorted array
    for(int i = 0; i < 5; i++){
        cout << a[i] << " ";
    }
}
```



Output:

```
PS D:\Hasan\cpp\university\lab08-1 D Array> g++ task01.cpp
PS D:\Hasan\cpp\university\lab08-1 D Array> ./a.exe
Enter the element no. 1 of array: 6
Enter the element no. 2 of array: 3
Enter the element no. 3 of array: 8
Enter the element no. 4 of array: 2
Enter the element no. 5 of array: 0
Original array:
6 3 8 2 0
Sorted array in ascending order:
0 2 3 6 8
PS D:\Hasan\cpp\university\lab08-1 D Array> |
```

```
PS D:\Hasan\cpp\university\lab08-1 D Array> ./a.exe
Enter the element no. 1 of array: 89
Enter the element no. 2 of array: 3
Enter the element no. 3 of array: 24
Enter the element no. 4 of array: 6
Enter the element no. 5 of array: 76
Original array:
89 3 24 6 76
Sorted array in ascending order:
3 6 24 76 89
PS D:\Hasan\cpp\university\lab08-1 D Array> |
```



Task 02:

Write a C program to search an element entered by user from array and display the searched element and its location.

Code:

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

int main(){
    int size = 0;
    int target = 0;
    bool found = false;

    cout << "Enter the no. of elements in array: ";
    cin >> size;

    int a[size];

    for(int i = 0, sNo = 1; i < size; i++, sNo++){
        cout << "Enter the element no. "<<sNo<<" of array: ";
        cin >> a[i];
    }

    cout << "Enter the element to search in aarray: ";
    cin >> target;

    for(int i = 0, sNo = 1; i < size; i++, sNo++){
        if(a[i] == target){
            found = true;
            cout << "Target element '"<<a[i]<<"' found at place no. '"<<sNo<<"'."
<< endl;
        }
    }
    if(!found){
        cout << "Targeted element not found." << endl;
    }
}
```




Output:

```
PS D:\Hasan\cpp\university\lab08-1 D Array> g++ task02.cpp
PS D:\Hasan\cpp\university\lab08-1 D Array> ./a.exe
Enter the no. of elements in array: 5
Enter the element no. 1 of array: 2
Enter the element no. 2 of array: 8
Enter the element no. 3 of array: 4
Enter the element no. 4 of array: 9
Enter the element no. 5 of array: 2

Enter the element to search in array: 8

Target element '8' found at place no. '2'.
PS D:\Hasan\cpp\university\lab08-1 D Array> ./a.exe
Enter the no. of elements in array: 3
Enter the element no. 1 of array: 6
Enter the element no. 2 of array: 78
Enter the element no. 3 of array: 23

Enter the element to search in array: 9

Targeted element not found.
PS D:\Hasan\cpp\university\lab08-1 D Array> |
```



Task 03:

Write a C++ program to find total number elements repeated in an array also print all unique elements in an array.

Code:

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

int main(){
    int a[5] = {0};
    int unique[5] = {0};
    int temp = 0, repeatCount = 0, uniqueCount = 0;

    //getting elements from user
    for(int i = 0; i < 5; i++){
        cout << "Enter element no. '"<<(i+1)<<": ";
        cin >> a[i];
    }

    for(int i = 0; i < 5; i++){
        bool isRepeated = false;
        temp = a[i]; // holding the element
        for(int j = 0; j < 5; j++){
            if(i == j) continue; // not comparing element with itself
            if(temp == a[j]){
                isRepeated = true;
                break;
            }
        }

        if(isRepeated){ //checking that if the element is already counted
            bool alreadyCounted = false;
            for(int k = 0; k < i; k++){ //running the till before the holded
element
                if(temp == a[k]){
                    alreadyCounted = true;
                    break;
                }
            }
            if(!alreadyCounted){
                repeatCount++;
            }
        }
    }
}
```



```
// if it is not repeated
if(!isRepeated){
    unique[uniqueCount] = temp;
    uniqueCount++;
}
}

cout << "Total number of repeated elements is: "<<repeatCount<<"." << endl;
cout << "Unique elements are: ";
for(int i = 0; i < uniqueCount; i++){
    cout << unique[i] << " ";
}
}
```

Output:

```
PS D:\Hasan\cpp\university\lab08-1 D Array> g++ task03.cpp
PS D:\Hasan\cpp\university\lab08-1 D Array> ./a.exe
Enter element no. '1': 7
Enter element no. '2': 5
Enter element no. '3': 3
Enter element no. '4': 9
Enter element no. '5': 2

Total number of repeated elements is: 0.

Unique elements are: 7 5 3 9 2
PS D:\Hasan\cpp\university\lab08-1 D Array> ./a.exe
Enter element no. '1': 3
Enter element no. '2': 5
Enter element no. '3': 3
Enter element no. '4': 5
Enter element no. '5': 9

Total number of repeated elements is: 2.

Unique elements are: 9
PS D:\Hasan\cpp\university\lab08-1 D Array> 
```



Task 04:

Write a program in C++ to identify array in which no zero present, and print those numbers. If user input a value without zero program should terminate.

Code:

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

int main(){
    int max = 100;
    int validCount = 0;
    int a[max], valid[max];

    for(int i = 0; i < max; i++){
        bool zeroFound = false;
        do{
            cout << "Enter any integer without zero: ";
            cin >> a[i];

            if(a[i] == 0){
                cout << "You entered zero!" << endl;
                zeroFound = true;
                break;
            }
        }while(a[i] == 0); // if user directly enters zero. because we will not
        //able to check single zero with while loop

        int temp = a[i];
        int lastDigit = 0;

        while(temp != 0){
            lastDigit = temp%10; //getting the last digit
            if(lastDigit == 0){
                cout << "You entered integer with zero!" << endl;
                zeroFound = true;
                break;
            }
            else{
                temp /= 10; //removing the lastdigit if it is not equal to zero.
            }
        }

        if(zeroFound){
```



```
        break;
    }
    else{
        valid[validCount] = a[i];
        validCount++;
    }
}

cout << "Valid inputs: ";
for(int i = 0; i < validCount; i++){
    cout << valid[i] << " ";
}
}
```

Output:

```
PS D:\Hasan\cpp\university\lab08-1 D Array> g++ task04.cpp
PS D:\Hasan\cpp\university\lab08-1 D Array> ./a.exe
Enter any integer without zero: 23
Enter any integer without zero: 56
Enter any integer without zero: 85
Enter any integer without zero: 3
Enter any integer without zero: 5
Enter any integer without zero: 43
Enter any integer without zero: 2
Enter any integer without zero: 906

You entered integer with zero!

Valid inputs: 23 56 85 3 5 43 2
PS D:\Hasan\cpp\university\lab08-1 D Array> |
```

Ln 54, Col 16 Spaces: 4 UTF-8 CRLF {} C++ Go Live windows-gcc-x86



Task 05:

Write a C++ program that asks user to enter 10 integer values. Store those values in one dimensional array. Create another one dimensional array of same size, and store the values of first array in reverse order. Print the result on Screen

Code:

```
#include<iostream>
using namespace std;
int main(){
    int n = 5;
    int a[n], rev[n];

    for(int i = 0; i < n; i++){
        cout << "Enter element no. '"<<(i+1)<<"' : ";
        cin >> a[i];
    }
    // reversing
    for(int i = (n-1), j = 0; i >= 0; i--, j++){
        rev[j] = a[i];
    }
    cout << "Matrix - Original" << endl;
    cout << "-----" << endl;
    for(int i = 0; i < n; i++){
        cout << a[i] << " ";
    }
    cout << endl << endl;
    cout << "Matrix - Reverse" << endl;
    cout << "-----" << endl;
    for(int i = 0; i < n; i++){
        cout << rev[i] << " ";
    }
}
```

Output:

```
PS D:\Hasan\cpp\university\lab08-1 D Array> ./a.exe
Enter element no. '1': 4
Enter element no. '2': 5
Enter element no. '3': 6
Enter element no. '4': 7
Enter element no. '5': 8
Matrix - Original
-----
4 5 6 7 8
Matrix - Reverse
-----
8 7 6 5 4
```



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

Lab 09 –Arrays 2-D



Task 01:

Write a program to create a 2D array of size 3x3. The program takes input for each cell in the array and then calculates and displays the sum of each row.

Code:

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

int main(){
    int a[3][3];
    int rowSum[3] = {0};
    for(int i = 0; i < 3; i++){
        cout << "Row no. " <<(i+1)<<": " << endl;
        for(int j = 0; j < 3; j++){
            cout << "Enter value of coloumn no. " <<(j+1)<<": ";
            cin >> a[i][j];
        }
    }

    for(int i = 0; i < 3; i++){
        for(int j = 0; j < 3; j++){
            rowSum[i] += a[i][j];
        }
    }

    for(int i = 0; i < 3; i++){
        cout << "Sum of all elements of row no. " <<(i+1)<<": " <<rowSum[i] <<
endl;
    }
}
```




Output:

```
PS D:\Hasan\cpp\university\lab09-2 D Array> g++ task01.cpp
PS D:\Hasan\cpp\university\lab09-2 D Array> ./a.exe
Row no. 1:
    Enter value of coloumn no. 1: 1
    Enter value of coloumn no. 2: 2
    Enter value of coloumn no. 3: 3
Row no. 2:
    Enter value of coloumn no. 1: 9
    Enter value of coloumn no. 2: 8
    Enter value of coloumn no. 3: 7
Row no. 3:
    Enter value of coloumn no. 1: 4
    Enter value of coloumn no. 2: 6
    Enter value of coloumn no. 3: 5

Sum of all elements of row no. '1': 6
Sum of all elements of row no. '2': 24
Sum of all elements of row no. '3': 15
PS D:\Hasan\cpp\university\lab09-2 D Array> |
```



Task 02:

Write a program that takes a 3x3 matrix as input and asks for a number entered and prints out its position in the matrix. It displays not found if the number is not in the matrix.

Code:

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

int main(){
    int a[3][3];
    int target = 0;
    bool found = false;

    for(int i = 0; i < 3; i++){
        cout << "\nRow no. " << (i+1) << endl;
        for(int j = 0; j < 3; j++){
            cout << "\tEnter value coloumn no. '"<<(j+1)<<"': ";
            cin >> a[i][j];
        }
    }

    cout << "\nEnter element to find its position: ";
    cin >> target;

    int i, j;
    for(i = 0; i < 3; i++){
        for(j = 0; j < 3; j++){
            if(target == a[i][j]){
                found = true;
                break;
            }
        }
        if(found){
            break;
        }
    }
    if(found){
        cout << "\nTarget: " <<target<<" found at position: ["<<i<<"]["<<j<<"]."
<< endl;
    }
    else{
        cout << "\nTarget not found." << endl;
    }
}
```



Output:

```
PS D:\Hasan\cpp\university\lab09-2 D Array> g++ task02.cpp
PS D:\Hasan\cpp\university\lab09-2 D Array> ./a.exe
```

```
Row no. 1
Enter value coloumn no. '1': 4
Enter value coloumn no. '2': 5
Enter value coloumn no. '3': 7
```

```
Row no. 2
Enter value coloumn no. '1': 3
Enter value coloumn no. '2': 5
Enter value coloumn no. '3': 3
```

```
Row no. 3
Enter value coloumn no. '1': 6
Enter value coloumn no. '2': 7
Enter value coloumn no. '3': 7
```

```
Enter element to find its position: 7
```

```
Target: 7 found at position: [0][2].
```

```
PS D:\Hasan\cpp\university\lab09-2 D Array> 
```



Task 03:

Write a program which calculates the transpose of a 3x3 matrix.

Code:

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

int main(){
    int a[3][3], aTrans[3][3];

    for(int i = 0; i < 3; i++){
        cout << "\nRow no. " << (i+1) << endl;
        for(int j = 0; j < 3; j++){
            cout << "\tEnter value coloumn no. '"<<(j+1)<<"': ";
            cin >> a[i][j];
        }
    }

    for(int i = 0; i < 3; i++){
        for(int j = 0; j < 3; j++){
            aTrans[i][j] = a[j][i];
        }
    }

    cout << "\nOriginal - Matrix" << endl;
    cout << "-----" << endl;
    for(int i = 0; i < 3; i++){
        for(int j = 0; j < 3; j++){
            cout << a[i][j] << " ";
        }
        cout << endl;
    }

    cout << "\nTranspose - Matrix" << endl;
    cout << "-----" << endl;
    for(int i = 0; i < 3; i++){
        for(int j = 0; j < 3; j++){
            cout << aTrans[i][j] << " ";
        }
        cout << endl;
    }
}
```



Output:

```
PS D:\Hasan\cpp\university\lab09-2 D Array> g++ task03.cpp
PS D:\Hasan\cpp\university\lab09-2 D Array> ./a.exe
```

```
Row no. 1
    Enter value coloumn no. '1': 4
    Enter value coloumn no. '2': 6
    Enter value coloumn no. '3': 89
```

```
Row no. 2
    Enter value coloumn no. '1': 4
    Enter value coloumn no. '2': 2
    Enter value coloumn no. '3': 5
```

```
Row no. 3
    Enter value coloumn no. '1': 7
    Enter value coloumn no. '2': 8
    Enter value coloumn no. '3': 9
```

Original - Matrix

```
-----
4 6 89
4 2 5
7 8 9
```

Transpose - Matrix

```
-----
4 4 7
6 2 8
89 5 9
```

```
PS D:\Hasan\cpp\university\lab09-2 D Array> |
```



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

Lab 10 - Pointers



Task 01:

You're managing student information for a school. Your task is to implement a function that swaps the information of two students. This task will help you understand how pointers can be applied to real-world problems.

Code:

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

int main(){
    string name[2], grade[2];
    int age[2];

    string *pName = name, *pGrade = grade;
    int *pAge = age;

    for(int i = 0; i < 2; i++){
        cout << "Student no. " << i+1 << endl;
        cout << "\tEnter name: ";
        getline(cin, name[i]);
        cout << "\tEnter age: ";
        cin >> age[i];
        cout << "\tEnter grade: ";
        cin >> grade[i];
        cin.ignore();
    }

    cout << "Before swaping:" << endl;
    for(int i = 0; i < 2; i++){
        cout << "\tStudent no. "<<i+1<<" Name: "<<name[i]<<" Age: "<<age[i]<<"
Garde: "<<grade[i] << endl;
    }

    //swaping
    string tempName = *pName;
    *pName = *(pName+1);
    *(pName+1) = tempName;

    int tempAge = *pAge;
    *pAge = *(pAge+1);
    *(pAge+1) = tempAge;
```



```
string tempGrade = *pGrade;
*pGrade = *(pGrade+1);
*(pGrade+1) = tempGrade;

cout << "After swaping:" << endl;
for(int i = 0; i < 2; i++){
    cout << "\tStudent no. "<<i+1<<" Name: "<<name[i]<<" Age: "<<age[i]<<"
Garde: "<<grade[i] << endl;
}
}
```

Output:

```
PS D:\Hasan\cpp\00. university\lab10-Pointers> g++ task01.cpp
PS D:\Hasan\cpp\00. university\lab10-Pointers> ./a.exe
Student no. 1
    Enter name: Hasan
    Enter age: 18
    Enter grade: 80
Student no. 2
    Enter name: Anas
    Enter age: 19
    Enter grade: 70
Before swaping:
    Student no. 1 Name: Hasan Age: 18 Garde: 80
    Student no. 2 Name: Anas Age: 19 Garde: 70
After swaping:
    Student no. 1 Name: Anas Age: 19 Garde: 70
    Student no. 2 Name: Hasan Age: 18 Garde: 80
PS D:\Hasan\cpp\00. university\lab10-Pointers> █
```




Task 02:

Imagine you're managing expenses for a grocery shopping trip with your friends. Your task is to develop a function that computes the total expenditure and average spending per person based on the amounts spent by each individual on groceries. Implement the function to accept an array of integers representing the amounts spent by each person on groceries, using call by reference with pointers.

Code:

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

void avgAndTotal(int *a, int size){
    int total = 0;
    for(int i = 0; i < size; i++) total += *(a + i);

    float avg = total/float(size);
    cout << "Total expenditure is: " << total << endl;
    cout << "Average per person os: " << avg << endl;
}

int main(){
    int person = 0;
    cout << "Enter no. of persons: ";
    cin >> person;
    int a[person];
    for(int i = 0; i < person; i++){
        cout << "Enter expense of person no. "<<i+1<<": ";
        cin >> a[i];
    }
    avgAndTotal(a, person);
}
```

Output:



```
PS D:\Hasan\cpp\00. university\lab10-Pointers> g++ task02.cpp
PS D:\Hasan\cpp\00. university\lab10-Pointers> ./a.exe
Enter no. of persons: 4
Enter expense of person no. 1: 57
Enter expense of person no. 2: 8
Enter expense of person no. 3: 45
Enter expense of person no. 4: 23

Total expenditure is: 133

Average per person os: 33.25
PS D:\Hasan\cpp\00. university\lab10-Pointers> █
```



Task 03:

You're tasked with organizing a shelf of books in a library. Your task is to create a function that arranges the books in ascending order based on their heights. You can use pointers to efficiently manage their positions.

Code:

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

void organizeBooks(float *a, int size){
    cout << "\nHeights of Books before organizing:" << endl;
    for(int i = 0; i < size; i++){
        cout << *(a+i) << "cm ";
    }
    cout << endl;

    //organizing
    for(int i = 0; i < size; i++){
        for(int j = 0; j < (size-i-1); j++){
            if(*(a+j) > *(a+j+1)){
                float temp = *(a+j);
                *(a+j) = *(a+j+1);
                *(a+j+1) = temp;
            }
        }
    }

    cout << "\nHeights of Books after organizing:" << endl;
    for(int i = 0; i < size; i++){
        cout << *(a+i) << "cm ";
    }
    cout << endl;
}

int main(){
    int noOfBooks = 0, bookCount = 0, choice = 0;
    cout << "Welcome to Book Shelf Organizer" << endl << endl;
    cout << "Enter the no. of books: ";
    cin >> noOfBooks;

    float shelf[noOfBooks];
    do{
        cout << "\n---MENU---" << endl;
```



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

```
cout << "1. Add book." << endl;
```



```
cout << "2. See organized books." << endl;
cout << "3. Exit." << endl;
cout << "Enter choice: ";
cin >> choice;
if(choice == 1){
    if(bookCount >= noOfBooks){
        cout << "\nBook shelf is full." << endl;
    }
    else{
        cout << "\nEnter height of book (cm): ";
        cin >> shelf[bookCount];
        cout << "Book added!" << endl;
        bookCount++;
    }
}
else if(choice == 2){
    organizeBooks(shelf, bookCount);
}
else if(choice == 3){
    cout << "\nClosing program." << endl;
}
else{
    cout << "\nInvalid choice!" << endl;
}

}while(choice != 3);
}
```



Output:

```
PS D:\Hasan\cpp\00. university\lab10-Pointers> g++ task03.cpp
PS D:\Hasan\cpp\00. university\lab10-Pointers> ./a.exe
Welcome to Book Shelf Organizer

Enter the no. of books: 3

---MENU---
1. Add book.
2. See organized books.
3. Exit.
Enter choice: 1

Enter height of book (cm): 56
Book added!

---MENU---
1. Add book.
2. See organized books.
3. Exit.
Enter choice: 1

Enter height of book (cm): 3.4
Book added!

---MENU---
1. Add book.
2. See organized books.
3. Exit.
Enter choice: 2

Heights of Books before organizing:
56cm 3.4cm

Heights of Books after organizing:
3.4cm 56cm

---MENU---
```



```
---MENU---
1. Add book.
2. See organized books.
3. Exit.
Enter choice: 1

Enter height of book (cm): 89
Book added!

---MENU---
1. Add book.
2. See organized books.
3. Exit.
Enter choice: 1

Book shelf is full.

---MENU---
1. Add book.
2. See organized books.
3. Exit.
Enter choice: 2

Heights of Books before organizing:
3.4cm 56cm 89cm

Heights of Books after organizing:
3.4cm 56cm 89cm

---MENU---
1. Add book.
2. See organized books.
3. Exit.
Enter choice: 3

Closing program.
```



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

Lab 11 – Structures



Task 01:

Make a structure BookRec, which stores the following details of a book: title of the book, Author's name, its publisher and price in it. Write a program to take input of 2 books and display it on the console.

Code:

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

struct bookRec{
    string title, author, pub;
    float price = 0.00;
};

int main(){
    bookRec bd;

    for(int i = 0; i < 2; i++){
        cout << "\nEnter title: ";
        getline(cin, bd.title);
        cout << "Enter the name author: ";
        getline(cin, bd.author);
        cout << "Enter the name of publisher: ";
        getline(cin, bd.pub);
        cout << "Enter price: ";
        cin >> bd.price;
        cin.ignore();
    }

    cout << "\nDetails: " << endl;
    for(int i = 0; i < 2; i++){
        cout << "\nBook no. " << (i+1) << endl;
        cout << "\tTitle: " << bd.title << endl;
        cout << "\tAuthor: " << bd.author << endl;
        cout << "\tPublisher: " << bd.pub << endl;
        cout << "\tPrice: " << bd.price << endl;
    }
}
```



Output:

```
PS D:\Hasan\cpp\university\lab11-Structures> g++ task01.cpp
PS D:\Hasan\cpp\university\lab11-Structures> ./a.exe

Enter title: Random Book
Enter the name author: No one
Enter the name of publisher: 0
Enter price: 100

Enter title: Second Book
Enter the name author: Anonymous
Enter the name of publisher: Gaba Publisher
Enter price: 800

Details:

Book no. 1
    Title: Second Book
    Author: Anonymous
    Publisher: Gaba Publisher
    Price: 800

Book no. 2
    Title: Second Book
    Author: Anonymous
    Publisher: Gaba Publisher
    Price: 800
PS D:\Hasan\cpp\university\lab11-Structures> |
```



Task 02:

You are tasked with developing a software system to manage a library of books. The system should store information about each book and its author, allow users to search for books by title or author name, There should be at least 15 user input and display the result

Code:

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

struct bookDetails{
    string title, author;
    int pubYear = 0;
};

/*1. to add books*/
void addBook(bookDetails bd[], int size, int count){
    cout << "Enter title: ";
    getline(cin, bd[count].title);
    cout << "Enter the name of author: ";
    getline(cin, bd[count].author);
    cout << "Enter the year of publication: ";
    cin >> bd[count].pubYear;
    cin.ignore();
}

/*2. to search by title*/
void searchByTitle(bookDetails bd[], int size, int count){
    string targetTitle;
    bool found = false;
    cout << "Enter title of book which you want to find: ";
    getline(cin, targetTitle);

    for(int i = 0; i < count; i++){
        if(bd[i].title == targetTitle){
            cout << endl;
            cout << "Book found! Details." << endl;
            cout << "Title          Author          Publication Year" << endl;
            for(int i = 0; i < 46; i++) cout << "-";
            cout << endl;
            cout << left;
            cout
            <<setw(15)<<bd[i].title<<setw(15)<<bd[i].author<<setw(16)<<bd[i].pubYear<<endl;
        }
    }
}
```



```
        found = true;
        break;
    }
}
if(!found){
    cout << "\nBook not found." << endl << endl;
}
}

/*3. to search by author*/
void searchByAuthor(bookDetails bd[], int size, int count){
    string targetAuthor;
    bool found = false;
    cout << "Enter title of book which you want to find: ";
    getline(cin, targetAuthor);

    for(int i = 0; i < count; i++){
        if(bd[i].author == targetAuthor){
            cout << endl;
            cout << "Book found! Details." << endl;
            cout << "Title          Author          Publication Year" << endl;
            for(int i = 0; i < 46; i++) cout << "-";
            cout << endl;
            cout << left;
            cout
            <<setw(15)<<bd[i].title<<setw(15)<<bd[i].author<<setw(16)<<bd[i].pubYear<<endl;
            found = true;
            break;
        }
    }
    if(!found){
        cout << "\nBook not found." << endl << endl;
    }
}

/*4. to view sorted books*/
void viewSortedbooks(bookDetails bd[], int size, int count){

    for(int i = 0; i < count; i++){
        for(int j = 0; j < (count-1-i); j++){

            // int(bd[j].title.substr(0,1)) > int(bd[j+1].title.substr(0,1)) ->
            this will consider character as string
            if(int(bd[j].title[0]) > int(bd[j+1].title[0])){
```



```
        string Title = bd[j].title;
        bd[j].title = bd[j+1].title;
        bd[j+1].title = Title;

        string Author = bd[j].author;
        bd[j].author = bd[j+1].author;
        bd[j+1].author = Author;

        int tempYear = bd[j].pubYear;
        bd[j].pubYear = bd[j+1].pubYear;
        bd[j+1].pubYear = tempYear;
    }
}

cout << "Details of all books." << endl;
cout << "S.No.    Title            Author            Publication Year" << endl;
for(int i = 0; i < 54; i++) cout << "-";
cout << endl;
cout << left;

for(int i = 0, sNo = 1; i < count; i++, sNo++){
    cout
<<setw(8)<<sNo<<setw(15)<<bd[i].title<<setw(15)<<bd[i].author<<setw(16)<<bd[i].pubYear<<endl;
}
}

/*5. to display all books*/
void displayAllBooks(bookDetails bd[], int size, int count){
    cout << "Details of all books." << endl;
    cout << "S.No.    Title            Author            Publication Year" << endl;
    for(int i = 0; i < 54; i++) cout << "-";
    cout << endl;
    cout << left;

    for(int i = 0, sNo = 1; i < count; i++, sNo++){
        cout
<<setw(8)<<sNo<<setw(15)<<bd[i].title<<setw(15)<<bd[i].author<<setw(16)<<bd[i].pubYear<<endl;
    }
}

int main(){
```



```
bookDetails bd[15];
int bookCount = 0;
bool firstBookAdded = false;
int choice = 0;
cout << "***Library Management System***" << endl;
do{
    cout << endl;
    cout << "\t---MENU---" << endl;
    cout << "1. Add a new book." << endl;
    cout << "2. Search book by title." << endl;
    cout << "3. Search book by author." << endl;
    cout << "4. Sort books by title." << endl;
    cout << "5. Display all books." << endl;
    cout << "6. Exit" << endl;
    cout << "Enter choice: ";
    cin >> choice;
    cin.ignore();

    if(choice == 1){
        if(bookCount >= 15){
            cout << "\nLibrary filled!" << endl;
        }
        else{
            cout << endl;
            addBook(bd, 15, bookCount);
            bookCount++;
            firstBookAdded = true;
        }
    }

    else if(choice == 2){
        if(!firstBookAdded){
            cout << "\nPlease add book first!" << endl;
        }
        else{
            cout << endl;
            searchByTitle(bd, 15, bookCount);
        }
    }

    else if(choice == 3){
        if(!firstBookAdded){
            cout << "\nPlease add book first!" << endl;
        }
        else{
```



```
        cout << endl;
        searchByAuthor(bd, 15, bookCount);
    }
}

else if(choice == 4){
    if(!firstBookAdded){
        cout << "\nPlease add book first!" << endl;
    }
    else{
        cout << endl;
        viewSortedbooks(bd, 15, bookCount);
    }
}

else if(choice == 5){
    if(!firstBookAdded){
        cout << "\nPlease add book first!" << endl;
    }
    else{
        cout << endl;
        displayAllBooks(bd, 15, bookCount);
    }
}

else if(choice == 6){
    cout << "\nClosing program..." << endl;
}

else{
    cout << "\nInvalid choice!" << endl;
}
}while(choice != 6);
}
```



Output:

```
PS D:\Hasan\cpp\university\lab11-Structures> ./a.exe  
***Library Management System***
```

```
---MENU---
```

1. Add a new book.
 2. Search book by title.
 3. Search book by author.
 4. Sort books by title.
 5. Display all books.
 6. Exit
- Enter choice: 3

Please add book first!

```
---MENU---
```

1. Add a new book.
 2. Search book by title.
 3. Search book by author.
 4. Sort books by title.
 5. Display all books.
 6. Exit
- Enter choice: 1

Enter title: First Book
Enter the name of author: Hasan
Enter the year of publication: 2000

```
---MENU---
```

1. Add a new book.
 2. Search book by title.
 3. Search book by author.
 4. Sort books by title.
 5. Display all books.
 6. Exit
- Enter choice: 1

Enter title: Second Book



5. Display all books.

6. Exit

Enter choice: 1

Enter title: Second Book

Enter the name of author: No one

Enter the year of publication: 9900

---MENU---

1. Add a new book.

2. Search book by title.

3. Search book by author.

4. Sort books by title.

5. Display all books.

6. Exit

Enter choice: 2

Enter title of book which you want to find: First Book

Book found! Details.

Title	Author	Publication Year
-------	--------	------------------

First Book	Hasan	2000
------------	-------	------

---MENU---

1. Add a new book.

2. Search book by title.

3. Search book by author.

4. Sort books by title.

5. Display all books.

6. Exit

Enter choice: 3

Enter title of book which you want to find: No one



Enter choice: 3

Enter title of book which you want to find: No one

Book found! Details.

Title	Author	Publication Year

Second Book	No one	9900

---MENU---

1. Add a new book.
2. Search book by title.
3. Search book by author.
4. Sort books by title.
5. Display all books.
6. Exit

Enter choice: 4

Details of all books.

S.No.	Title	Author	Publication Year

1	First Book	Hasan	2000
2	Second Book	No one	9900

---MENU---

1. Add a new book.
2. Search book by title.
3. Search book by author.
4. Sort books by title.
5. Display all books.
6. Exit

Enter choice: 5



```
Enter choice: 5

Details of all books.
S.No.  Title      Author      Publication Year
-----
1      First Book   Hasan       2000
2      Second Book   No one      9900

---MENU---
1. Add a new book.
2. Search book by title.
3. Search book by author.
4. Sort books by title.
5. Display all books.
6. Exit
Enter choice: 6

Closing program...
PS D:\Hasan\cpp\university\lab11-Structures> |
```

Ln 209, Col 5 Spaces: 4 UTF-8 CRLF {} C++ Go Live windows-gcc-x86



Task 03:

Write a C++ program that compute Net Salary of Employee. Program contains two user defined functions empSalary() and display().

1. Create a structure of Employee that contains following data members:
 - EmployeeNumber, Name, BasicSalary, HouseAllowance, MedicalAllowance, Tax, GrossPay and NetSalary - Employee number, name and basic salary must be taken input from the user.
2. empSalary() compute salary with given criteria:
 - House Allowance = 12% of Basic Salary
 - Medical Allowance = 8% of Basic Salary
 - Tax = 5% of Basic Salary
 - GrossSalary = Basic + House Allowance + Medical Allowance
 - NetSalary = GrossSalary – Tax
3. display() for displaying details of Employee

Code:

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

struct employee{
    string name;
    int empNum = 0;
    float basicSal = 0.00, houseAll = 0.00, medAll = 0.00, tax = 0.00, grossSal = 0.00, netSal = 0.00;
};

void calculateSal(employee &eDetails){
    eDetails.houseAll += (eDetails.basicSal*(12/100.0));
    eDetails.medAll += (eDetails.basicSal*(8/100.0));
    eDetails.tax += (eDetails.basicSal*(5/100.0));
    eDetails.grossSal = (eDetails.basicSal + eDetails.houseAll + eDetails.medAll + 0.0);
    eDetails.netSal = (eDetails.grossSal - eDetails.tax) + 0.00;
}

void display(employee ed){
    cout << "\nPay Slip:" << endl << endl;
    cout << "Emp No. Name Basic Salary House Allowance(12%) Medical Allowance(8%) Tax(5%) Gross Salary Net Salary\n";
    for(int i = 0; i < 115; i++) cout << "-";
    cout << endl;
```



```
    cout << left;

    cout<<setw(9)<<ed.empNum<<setw(15)<<ed.name<<setw(15)<<ed.basicSal<<setw(22)<
<ed.houseAll<<setw(22)<<ed.medAll<<setw(9)<<ed.tax<<setw(13)<<ed.grossSal<<setw(1
0)<<ed.netSal<<endl;
}

int main(){
    employee eDetail;

    cout << "Enter employee number: ";
    cin >> eDetail.empNum;
    cin.ignore();
    cout << "Enter employee name: ";
    getline(cin, eDetail.name);
    cout << "Enter basic salary: ";
    cin >> eDetail.basicSal;

    calculateSal(eDetail);
    display(eDetail);
}
```

Output:

```
PS D:\Hasan\cpp\university\lab11-Structures> ./a.exe
Enter employee number: 1234
Enter employee name: Muhammad Hasan
Enter basic salary: 120000

Pay Slip:

Emp No.   Name           Basic Salary   House Allowence(12%)   Medical Allowence(8%)   Tax(5%)   Gross Salary   Net Salary
-----
1234      Muhammad Hasan  120000         14400                  9600                   6000      144000         138000
PS D:\Hasan\cpp\university\lab11-Structures> |
```



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

Lab 12 – Strings



Task 01:

Imagine a greeting card company that allows customers to personalize their greeting cards with custom messages. One of the features provided by the company is the ability to replace specific words in their messages with other words. This is particularly useful for creating a consistent theme across multiple cards, such as changing "Birthday" to "Anniversary" or "Congratulations" to "Well Done". The company has tasked you with developing a software solution that will allow customers to easily make these changes to their messages.

Code:

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

int main() {
    string message, oldWord, newWord;
    cout << "Enter your message: ";
    getline(cin, message);
    cout << endl;

    cout << "Enter the word you want to replace: ";
    cin >> oldWord;
    cout << "Enter the new word: ";
    cin >> newWord;

    int pos = message.find(oldWord);
    if (pos != -1) {
        message.replace(pos, oldWord.length(), newWord);
        cout << "\nUpdated Message: " << message << endl;
    }
    else {
        cout << "Word not found!" << endl;
    }
}
```



Output:

```
PS D:\Hasan\cpp\00. university\lab12-Strings> g++ task01.cpp
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe
Enter your message: Lab 11: Strings

Enter the word you want to replace: Strings
Enter the new word: Pointers

Updated Message: Lab 11: Pointers
PS D:\Hasan\cpp\00. university\lab12-Strings> |
```

Task 02:

You are working on a customer service platform where agents often need to combine various pieces of information to form a complete response to customer inquiries. To streamline this process, you will create a feature that allows agents to input different parts of their response separately and then concatenate them into a single, coherent message.

Develop a C++ program that allows customer service agents to input multiple parts of a response message, concatenate these parts into a single message, and display the final message.

Code:

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

int main() {
    string part1, part2, part3, part4, finalPart;

    cout << "Enter the first part of the response: ";
    getline(cin, part1);
    cout << "Enter the second part of the response: ";
    getline(cin, part2);
    cout << "Enter the third part of the response: ";
    getline(cin, part3);
    cout << "Enter the fourth part of the response: ";
    getline(cin, part4);

    finalPart = part1 + " " + part2 + " " + part3 + " " + part4;
    cout << "\nFinal Response: " << finalPart << endl;
}
```




Output:

```
PS D:\Hasan\cpp\00. university\lab12-Strings> g++ task02.cpp
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe
Enter the first part of the response: ThankYou
Enter the second part of the response: for
Enter the third part of the response: coming
Enter the fourth part of the response: Bye!

Final Response: ThankYou for coming Bye!
```

Task 03:

Imagine we're developing a program tailored for creating user accounts, where the system necessitates specific string manipulations to ensure the usernames adhere to the required format.

- Convert the string to uppercase.
- Replace all spaces with underscores ('_').
- Reverse the string.
- Return the modified string.

Code:

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

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

    for (int i = 0; i < username.length(); i++) {
        if (username[i] > 96 && username[i] < 123) {
            username[i] -= 32;
        }
    }

    for (int i = 0; i < username.length(); i++) {
        if (username[i] == ' ') {
            username[i] = '_';
        }
    }

    int start = 0, end = username.length() - 1;
    while (start < end) {
        char temp = username[start];
        username[start] = username[end];
        username[end] = temp;
        start++;
        end--;
    }
}
```



```
        end--;  
    }  
    cout << "\nModified Username: " << username << endl;  
}
```

Output:

```
PS D:\Hasan\cpp\00. university\lab12-Strings> g++ task03.cpp  
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe  
Enter a username: muhammad hasan  
  
Modified Username: NASAH_DAMMAHUM  
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe  
Enter a username: hasan  
  
Modified Username: NASAH
```

Task 04:

Write a program that takes a paragraph as input and formats it into a well-structured document. The program should capitalize the first letter of each sentence, ensure proper spacing between words, and handle special cases such as abbreviations and proper nouns.

Requirements:

- Prompt the user to enter a sentence.
- Read the input sentence from the user.
- Convert the first letter of each word to uppercase.
- Print the modified sentence with the capitalized words.

Code:

```
#include<iostream>  
#include<string>  
using namespace std;  
  
int main() {  
    string sentence;  
    cout << "Enter a sentence: ";  
    getline(cin, sentence);  
  
    if (sentence[0] > 96 && sentence[0] < 123) {  
        sentence[0] -= 32;  
    }  
    cout << "\nFormatted Sentence: " << sentence << endl;  
}
```



Output:

```
PS D:\Hasan\cpp\00. university\lab12-Strings> g++ task04.cpp
PS D:\Hasan\cpp\00. university\lab12-Strings> ./a.exe
Enter a sentence: welcome to my program.

Formatted Sentence: Welcome to my program.
```



Bahria University
Discovering Knowledge

CSL 113: Computer Programming Lab
Semester: BSIT 01

Lab 13 – File Handling



Task 01:

You are tasked with creating a program to back up the library's catalog. The program should copy the contents of the main catalog file to a backup file, ensuring that the library's data is securely preserved in case of any unforeseen events.

Code:

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

int main(){
    string row;
    cout << "Catalog backup process initiated." << endl;

    cout << "Opening main catalog to get data." << endl;
    ifstream mainFileIn("mainCatalog.txt", ios::in);
    if(!mainFileIn.is_open()){
        cout << "Error in opening main catalog." << endl;
    }
    else{
        cout << "Main catalog opened successfully." << endl;
    }

    cout << "\nOpening backup catalog to send data." << endl;
    ofstream backupFileOut("backupCatalog.txt", ios::app);
    if(!backupFileOut) cout << "Error in opening backup catalog." << endl;
    else{
        cout << "Backup catalog opened successfully." << endl;
    }

    while(!mainFileIn.eof()){
        getline(mainFileIn, row);
        backupFileOut << row << endl;
    }
    mainFileIn.close();
    backupFileOut.close();
    cout << "\nData copied successfully." << endl;
}
```



Output:

```
task01.cpp  mainCatalog.txt  backupCatalog.txt
mainCatalog.txt
1  C++ is a very good language.
2  You should learn C++.
3  |
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\Hasan\cpp\university\lab13-File Handling> g++ task01.cpp
PS D:\Hasan\cpp\university\lab13-File Handling> ./a.exe
Catalog backup process initiated.
Opening main catalog to get data.
Main catalog opened successfully.

Opening backup catalog to send data.
Backup catalog opened successfully.

Data copied successfully.
PS D:\Hasan\cpp\university\lab13-File Handling> |
```

```
task01.cpp  mainCatalog.txt  backupCatalog.txt
backupCatalog.txt
1  C++ is a very good language.
2  You should learn C++.
3
4
```



Task 02:

You're developing a motivational quote generator where users can input a four-word quote, which will be permanently stored. Since users may prefer mixed-case quotes, you'll present the final output for review before saving it. After storing the words, the program will display the quote reconstructed with all words in uppercase, providing a visual representation of how the quote would look with case conversion.

Code:

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

int main(){
    string word[4];
    char choice;

    // getting quote fom user word by word
    cout << "Enter your 4-words based quote:" << endl;
    for(int i = 0; i < 4; i++){
        cout << "\tEnter word no. "<<(i+1)<<": ";
        getline(cin, word[i]);
    }

    //saving user's entered quote into file
    ofstream userOut("userQuote.txt", ios::app);
    if(!userOut.is_open()) cout << "Error in opening file to save user quote." <<
endl;
    else{
        for(int i = 0; i < 4; i++){
            userOut << word[i] << " ";
        }
        userOut << endl;
        cout << "\nUser quote saved successfully!" << endl;
        userOut.close();
    }
    // converting the quote unto uppercase
    for(int i = 0; i < 4; i++){
        for(int j = 0; j < word[i].length(); j++){
            if(word[i][j] >= 97 && word[i][j] <= 122){
                word[i][j] -= 32;
            }
        }
    }
}
```



```
// displaying the uppercase quote, & asking for verification
cout << "\nHere's the upgraded version of your quote: " << endl;
for(int i = 0; i < 4; i++){
    cout << word[i] << " ";
}
cout << endl;
do{
    cout << "\nIs this how you want to save quote (uppercase)? (y/n): ";
    cin >> choice;

    if(choice != 'y' && choice != 'n'){
        cout << "\nPlease enter valid choice! (y/n)" << endl;
    }

}while(choice != 'y' && choice != 'n');

// if approved by user, saving it into file
if(choice == 'y'){
    ofstream upperOut("uppercaseQuote.txt", ios::app);
    if(!upperOut.is_open()) cout << "Error in opening file to save uppercase
quote!" << endl;
    else{
        for(int i = 0; i < 4; i++){
            upperOut << word[i] << " ";
        }
        upperOut << endl;
        cout << "The uppercase quote has been saved successfully!" << endl;
        upperOut.close();
    }
}
else{
    cout << "\nTerminating program without saving..." << endl;
}
}
```




Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Opening backup catalog to send data.
Backup catalog opened successfully.

Data copied successfully.
PS D:\Hasan\cpp\university\lab13-File Handling> g++ task02.cpp
PS D:\Hasan\cpp\university\lab13-File Handling> ./a.exe
Enter your 4-words based quote:
    Enter word no. 1: my
    Enter word no. 2: nAme
    Enter word no. 3: is
    Enter word no. 4: Hasan

User quote saved successfully!

Here's the upgraded version of your quote:
MY NAME IS HASAN

Is this how you want to save quote (uppercase)? (y/n): y
The uppercase quote has been saved successfully!
PS D:\Hasan\cpp\university\lab13-File Handling> █
```

```
task02.cpp ×  userQuote.txt ×  uppercaseQuote.txt
userQuote.txt
1  my nAme is Hasan
2
```

```
task02.cpp  userQuote.txt  uppercaseQuote.txt ×
uppercaseQuote.txt
1  MY NAME IS HASAN
2
```



Task 04:

Imagine you're a teacher managing student grades for your class. You need a program to update the grades of your students stored in a text file. The program will prompt you to enter a number representing the amount to increase the grades by. It will then read the student data from the file, increment each student's grade by the specified amount, and display the updated grades.

Code:

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

struct student{
    string name;
    int marks = 0;
};

void addData(){
    student s;
    cout << endl;
    cout << "Enter student's name: ";
    getline(cin, s.name);
    cout << "Enter marks: ";
    cin >> s.marks;
    cin.ignore();

    ofstream fileOut("studentData.txt", ios::app);
    if(!fileOut.is_open()) cout << "Error in opening file to send student data."
<< endl;
    else{
        fileOut << s.name << "|" << s.marks << endl;
        fileOut.close();
        cout << "Student rexorded siuccessfully!" << endl;
    }
}

void updateData(){
    student oldData;
    int increment = 0;
    string row;
    cout << endl;
    cout << "Enter marks which you want to add: ";
    cin >> increment;
```



```
ifstream stuIn("studentData.txt", ios::in);
ofstream tempOut("temp.txt", ios::app);

if(!stuIn.is_open()) cout << "Error in opening file to read student data." <<
endl;
else{
    while(!stuIn.eof()){
        getline(stuIn, row);
        if(row.empty()) continue;

        int pos1 = row.find('|');

        oldData.name = row.substr(0, pos1);
        oldData.marks = stoi(row.substr(pos1 + 1));

        tempOut << oldData.name << "|" << (oldData.marks + increment) <<
endl;
    }

    tempOut.close();
    stuIn.close();

    remove("studentData.txt");
    rename("temp.txt", "studentData.txt");
}

int main(){
    int choice = 0;
    cout << "***Welcome to Grades Updating App***" << endl;
    do{
        cout << "\n\t---MENU---" << endl;
        cout << "1. Add data." << endl;
        cout << "2. Update data." << endl;
        cout << "3. Exit." << endl;
        cout << "Enter choice: ";
        cin >> choice;
        cin.ignore();

        switch(choice){
            case 1:
                addData();
                break;
            case 2:
                updateData();
                break;
            case 3:
                return 0;
            default:
                cout << "Invalid choice. Please enter a valid choice." << endl;
        }
    } while(choice < 4);
}
```



```
        break;
    case 3:
        cout << "\nClosing program..." << endl;
        break;
    default:
        cout << "\nInvalid choice!" << endl;
        break;
    }
}while(choice != 3);
}
```



Output:

```
PS D:\Hasan\cpp\university\lab13-File Handling> ./a.exe
***Welcome to Grades Updating App***

    ---MENU---
1. Add data.
2. Update data.
3. Exit.
Enter choice: 1

Enter student's name: Muhammad Hasan
Enter marks: 10
Student rexorded siuccessfully!

    ---MENU---
1. Add data.
2. Update data.
3. Exit.
Enter choice: 2

Enter marks which you want to add: 5

    ---MENU---
1. Add data.
2. Update data.
3. Exit.
Enter choice: 1

Enter student's name: Random
Enter marks: 20
Student rexorded siuccessfully!

    ---MENU---
1. Add data.
2. Update data.
3. Exit.
Enter choice: 2
```



```
Enter choice: 2

Enter marks which you want to add: 5

    ---MENU---
1. Add data.
2. Update data.
3. Exit.
Enter choice: 3

Closing program...
PS D:\Hasan\cpp\university\lab13-File Handling> |
```

The screenshot shows a code editor with two tabs: 'task04.cpp' and 'studentData.txt'. The 'studentData.txt' tab is active, displaying the following content:

```
studentData.txt
1  Muhammad Hasan|20
2  Random|25
3  
```