

C++ PROGRAMMING LAB



Prepared by:

Name of Student: Manmeet Singh

Roll No: 16

Batch: 2023-27

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Exp. No	List of Experiment
	Write a program to find the roots of a quadratic equation.
	Write a program to calculate the power of a number using a loop.
	Write a program to check if a given string, is a palindrome.
	Write a program that simulates a simple ATM machine, allowing users to check their balance, deposit, or withdraw money using a switch statement.
	Write a program that finds the largest among three numbers using nested if-else statements
	Write a program that determines the grade of a student based on their marks of 5 subjects using if-else-if ladder.
	Write a program to find the sum of digits of a number until it becomes a single-digit number.
	Write a program to print a Pascal's triangle using nested loops.
	Write a program to calculate the sum of series $1/1! + 2/2! + 3/3! + \dots + N/N!$ using nested loops.
0	Write a program to create an array of strings and display them in alphabetical order.
1	Write a program that checks if an array is sorted in ascending order.
2	Write a program to calculate the sum of elements in each row of a matrix.
3	Write a program to generate all possible permutations of a string.
4	<p>Create a C++ program to print the following pattern:</p> <pre> ***** * * * * * * ***** </pre>
5	<p>Write a C++ program to display the following pattern:</p> <pre> 1 232 34543 </pre>

	4567654 34543 232
6	<p>Write a program to creating an inventory management system for a small store. The system should use object-oriented principles in C++. Your program should have the following features:</p> <ul style="list-style-type: none"> • Create a Product class that represents a product in the inventory. Each Product object should have the following attributes: <ul style="list-style-type: none"> • Product ID (an integer) • Product Name (a string) • Price (a floating-point number) • Quantity in stock (an integer) • Implement a parameterized constructor for the Product class to initialize the attributes when a new product is added to the inventory.
7	Write a program to manage student records. Create a class Student with attributes such as name, roll number, and marks. Implement methods for displaying student details, adding new students, and calculating the average marks of all students in the record system.
8	Write a program that implements a basic calculator. Use a class Calculator with methods to perform addition, subtraction, multiplication, and division of two numbers. The program should allow the user to input two numbers and select an operation to perform.
9	Write a program to simulate a simple online shop. Create a class Product with attributes like name, price, and quantity in stock. Implement methods for adding products to the shopping cart, calculating the total cost, and displaying the contents of the cart.
0	Write a program to manage student grades for a classroom. Create a class Student with attributes for student name and an array to store grades. Implement methods for adding grades, calculating the average grade, and displaying the student's name and grades. Use constructors and destructors to initialize and release resources.

Name of Student: Manmeet Singh

Roll Number: 16

Experiment No: 1

Title: Write a program to find the roots of a quadratic equation.

Theory:

The C++ program uses the quadratic formula to calculate roots, distinguishing between real and complex roots based on the discriminant value in $ax^2 + bx + c = 0$.

Code:

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    double a, b, c, discriminant, root1, root2, realPart, imaginaryPart;

    cout << "Enter the value of a:";
    cin >> a;
    cout << "Enter the value of b: ";
    cin >> b;
    cout << "Enter the value of c: ";
    cin >> c;

    discriminant = b * b - 4 * a * c;

    if (discriminant > 0) {
        root1 = (-b + sqrt(discriminant)) / (2 * a);
        root2 = (-b - sqrt(discriminant)) / (2 * a);
        cout << "Roots are real and different." << endl;
        cout << "Root 1 = " << root1 << endl;
        cout << "Root 2 = " << root2 << endl;
    } else if (discriminant == 0) {
        root1 = root2 = -b / (2 * a);
        cout << "Roots are real and equal." << endl;
    }
```

```

        cout << "Root 1 = Root 2 = " << root1 << endl;
    } else {
        realPart = -b / (2 * a);
        imaginaryPart = sqrt(-discriminant) / (2 * a);
        cout << "Roots are complex and different." << endl;
        cout << "Root 1 = " << realPart << " + " << imaginaryPart << "i" << endl;
        cout << "Root 2 = " << realPart << " - " << imaginaryPart << "i" << endl;
    }

    return 0;
}

```

Output:

```

Enter the value of a:4
Enter the value of b: 6
Enter the value of c: 2
Roots are real and different.
Root 1 = -0.5
Root 2 = -1

```

Test Cases

1)

```

Enter the value of a:5
Enter the value of b: 8
Enter the value of c: 1
Roots are real and different.
Root 1 = -0.136675
Root 2 = -1.46332

```

2)

```
Enter the value of a:16
Enter the value of b: 78
Enter the value of c: 54
Roots are real and different.
Root 1 = -0.835499
Root 2 = -4.0395
```

Conclusion:

It efficiently computes roots of a quadratic equation, providing real or complex solutions based on the discriminant value, showcasing C++ mathematical operations.

Name of Student: Manmeet Singh

Roll Number: 16

Experiment No: 2

Title: Write a program to calculate the power of a number using a loop.

Theory:

The C++ program defines a function to calculate a number raised to an exponent using a loop and displays the result.

Code:

```
#include <iostream>

using namespace std;

int calculate_power(int base, int exponent) {
    int result = 1;
    for (int i = 0; i < exponent; i++) {
        result *= base;
    }
    return result;
}

int main() {
    int base, exponent;
    cout << "Enter base: ";
    cin >> base;
    cout << "Enter exponent: ";
    cin >> exponent;
    int power = calculate_power(base, exponent);
    cout << base << " raised to the power of " << exponent << " is: " << power << endl;
    return 0;
}
```

Output

```
Enter base: 2
Enter exponent: 3
2 raised to the power of 3 is: 8
```

Test Cases

1)

```
Enter base: 3  
Enter exponent: 2  
3 raised to the power of 2 is: 9
```

2)

```
Enter base: 4  
Enter exponent: 5  
4 raised to the power of 5 is: 1024
```

Conclusion

Efficiently computes the power of a number using a function, showcasing basic mathematical operations in C++.

Name of Student: Manmeet Singh

Roll Number: 16

Experiment No: 3

Title: Write a program to check if a given string, is a palindrome.

Theory:

Palindrome string is a string whose reverse is equal to the original string entered by the user

Code:

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

int main()
{
    string str, temp;
    int i = 0, j, z;

    cout << "Enter a string to check for Palindrome: ";
    cin >> str;

    temp = str;

    j = str.length() - 1;

    //Reversing the temp string.

    while (i < j)
    {
        z = str[i];
        str[i] = str[j];
        str[j] = z;
        i++;
        j--;
    }
}
```

```
    if (temp == str)
    {
        cout << "The string is a palindrome." << endl;
    }
    else
    {
        cout << "The string is not a palindrome." << endl;
    }

    return 0;
}
```

Output

```
Enter a string to check for Palindrome: naman
The string is a palindrome.
```

Test Case

1)

```
Enter a string to check for Palindrome: Aba
The string is not a palindrome.
```

2)

```
Enter a string to check for Palindrome: aba
The string is a palindrome.
```

Conclusion: The C++ code checks if a given string is a palindrome by comparing it to its reversed form, accurately determining palindrome status for the input string.

Name of Student: Manmeet Singh

Roll Number: 16

Experiment No: 4

Title: Write a program that simulates a simple ATM machine, allowing users to check their balance, deposit, or withdraw money using a switch statement.

Theory:

The C++ program simulates an ATM interface, allowing deposit, withdrawal, and balance check operations via a switch-case structure within a while loop.

Code

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

int main()
{
    int choice;
    float deposit = 0;
    float withdraw = 0;
    float balance = 0;
    char ch;
    cout << "\n\nWELCOME TO ATM MACHINE\n\nPress 1 to deposit\nPress 2 to withdraw\n\nPress 0 to exit the program\nEnter your choice: ";
    cin >> choice;
    ch = 'y';
    while (ch == 'y' || ch == 'Y')
    {
        switch (choice)
        {
            case 0:
                cout << "Thankyou";
                break;
            case 1:
                cout << "\nEnter the amount you want to deposit: ";
                cin >> deposit;
                cout << "\nYou deposited: " << deposit;
                balance += deposit;
                break;
            case 2:
                cout << "\nEnter the amount you want to withdraw: ";
                cin >> withdraw;
                cout << "\nYou withdrew: " << withdraw;
                balance -= withdraw;
                break;
            default:
                cout << "Invalid choice\n\n";
                break;
        }
        ch = 'n';
    }
    cout << "\n\nYour current balance is: " << balance;
    return 0;
}
```

```

case 2:
    cout << "\nEnter the amount you want to withdraw: ";
    cin >> withdraw;
    if (withdraw > balance)
    {
        cout << "\nInsufficient funds";
    }
    else
    {
        cout << "\nYou withdrew: " << withdraw;
        balance -= withdraw;
    }

case 3:
    cout << "\nYour balance is: " << balance;

default:
    cout << "\nDo you want to enter again? Y/N: ";
    cin >> ch;

if (ch == 'n' || ch == 'N')
{
    break;
}

```

```

}
```

```

cout << "program is over";
```

```

}
```

Output

WELCOME TO ATM MACHINE

Press 1 to deposit

Press 2 to withdraw

Press 3 to checkbalance

Press 0 to exit the program

Enter your choice: 1

Enter the amount you want to deposit: 500

You deposited: 500

Enter the amount you want to withdraw: 200

You withdrew: 200

Your balance is: 300

Do you want to enter again? Y/N: N

program is over

Test Cases

WELCOME TO ATM MACHINE

Press 1 to deposit

Press 2 to withdraw

Press 3 to checkbalance

Press 0 to exit the program

Enter your choice: 2

Enter the amount you want to withdraw: 1

Insufficient funds

Your balance is: 0

Do you want to enter again? Y/N: y

WELCOME TO ATM MACHINE

Press 1 to deposit

Press 2 to withdraw

Press 3 to checkbalance

Press 0 to exit the program

Enter your choice: 1

Enter the amount you want to deposit: 100

You deposited: 100

Enter the amount you want to withdraw: 50

```
You withdrew: 50  
Your balance is: 50  
Do you want to enter again? Y/N: n  
program is over%
```

Conclusion

The program models an ATM system enabling basic banking operations and prompting user interaction until exit, demonstrating a simple menu-driven design in C++.

Name of Student: Manmeet Singh

Roll Number: 16

Experiment No: 5

Title: Write a program that finds the largest among three numbers using nested if-else statements

Theory:

The C++ program compares three user-input numbers using nested if-else statements to identify and display the largest number among them.

Code

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

int main()
{
    int num1, num2, num3;
    cout << "Enter first number: ";
    cin >> num1;
    cout << "Enter second number: ";
    cin >> num2;
    cout << "Enter third number: ";
    cin >> num3;

    if(num1 > num2)
    {
        if(num1 > num3)
        {
            cout << num1 << " is the largest";
        }
        else
        {
            cout << num3 << " is the largest";
        }
    }
    else
    {
        if(num2 > num3)
        {
```

```
{  
    cout << num2 << " is the largest";  
}  
else  
{  
    cout << num3 << " is the largest";  
}  
}
```

Output

```
Largest Among three && 705  
Enter first number: 23  
Enter second number: 56  
Enter third number: 1  
56 is the largest%
```

Test Cases

1)

```
Enter first number: 34  
Enter second number: 56  
Enter third number: 3  
56 is the largest%
```

2)


```
Enter first number: 66  
Enter second number: 66  
Enter third number: 9  
66 is the largest%
```

Conclusion

This C++ code determines the largest among three numbers input by the user using nested if-else statements, checking pairwise for the maximum value, and outputting the result without considering cases where two or more numbers might be equal and the largest simultaneously.

Name of Student: Manmeet Singh

Roll Number: 16

Experiment No: 6

Title: Write a program that finds the largest among three numbers using nested if-else statements

Theory:

This C++ code calculates the average of five subject marks and assigns a grade based on predefined score ranges.

Code

```
#include <iostream>

int main() {
    float marks[5];
    float total = 0.0;
    float average;
    char grade;

    for (int i = 0; i < 5; ++i) {
        std::cout << "Enter marks for subject " << (i + 1) << ": ";
        std::cin >> marks[i];
        total += marks[i];
    }

    average = total / 5.0;

    if (average >= 90) {
        grade = 'A';
    } else if (average >= 80) {
        grade = 'B';
    } else if (average >= 70) {
        grade = 'C';
    } else if (average >= 60) {
        grade = 'D';
    }
```

```

    grade = 'D';
} else {
    grade = 'F';
}

std::cout << "Average marks: " << average << std::endl;
std::cout << "Grade: " << grade << std::endl;

return 0;
}

```

Output

```

7 _12072020
Enter marks for subject 1: 34
Enter marks for subject 2: 90
Enter marks for subject 3: 98
Enter marks for subject 4: 95
Enter marks for subject 5: 50
Average marks: 73.4
Grade: C

```

Test Cases

1)

```

7 _12072020
Enter marks for subject 1: 68
Enter marks for subject 2: 90
Enter marks for subject 3: 87
Enter marks for subject 4: 57
Enter marks for subject 5: 90
Average marks: 78.4
Grade: C

```

2)

```
Enter marks for subject 1: 90
Enter marks for subject 2: 90
Enter marks for subject 3: 78
Enter marks for subject 4: 67
Enter marks for subject 5: 100
Average marks: 85
Grade: B
```

Conclusion

Calculates average marks and assigns a grade based on predefined score ranges for five subjects, displaying both average and grade.

Name of Student: Manmeet Singh

Roll Number: 16

Experiment No: 7

Title: Write a program to find the sum of digits of a number until it becomes a single-digit number.

Theory:

This C++ code computes the sum of digits of a number until it reduces to a single digit by repeatedly summing its digits.

Code

```
#include <iostream>
using namespace std;
void SumUntillLastDigit(int n) {
    int sum = 0;
    while(n > 0 || sum > 9) {
        if(n == 0) {
            n = sum;
            sum = 0;
        }
        sum += n % 10;
        n /= 10;
    }
    cout << sum << endl;
}
int main() {
    int n ;
    cout << "Enter a number: ";
    cin >> n;
    SumUntillLastDigit(n);
    return 0;
}
```

Output

```
Enter a number: 34
7
```

Test Cases

1)

```
Enter a number: 56
2
```

2)

```
Enter a number: 99
9
```

Conclusion

Computes the sum of digits of a number until it's a single digit, showcasing the final summation.

Name of Student: Manmeet Singh

Roll Number: 16

Experiment No: 8

Title: Write a program to print a Pascal's triangle using nested loops.

Theory:

This C++ code uses nested loops to create a pyramid pattern with numbers and spaces. It prints ascending numbers from 1 to 'i' and then descending numbers from 'i-1' to 1, with spaces appropriately aligned.

Code

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

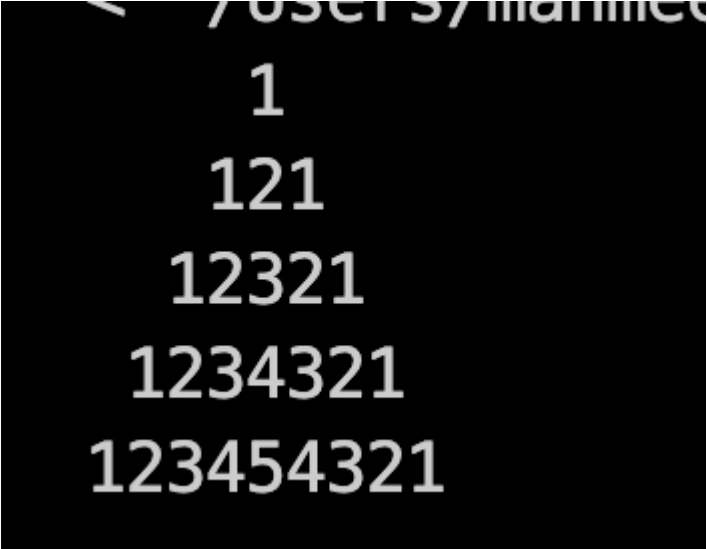
int main()
{
    int n = 5;
    int a;

    for(int i = 1; i <= n ; i++)

    {
        for(int z = 1; z <= n-i; z++)
        {
            cout << " ";
        }
        for(int k = 1; k <= i; k++)
        {
            cout << k;
        }
    }
}
```

```
}  
a = i - 1;  
for(int j = 1; j <= i-1; j++)  
{  
    cout << a;  
    a--;  
}  
  
cout << endl;  
}
```

Output



```
1  
121  
12321  
1234321  
123454321
```

Conclusion

Produces a pyramid pattern of numbers, ascending and descending, with spaces, forming a visual pattern.

Name of Student: Manmeet Singh

Roll Number: 16

Experiment No: 9

Title: Write a program to calculate the sum of series $1/1! + 2/2! + 3/3! + \dots + N/N!$ using nested loops.

Theory:

This C++ program computes the sum of a series by incrementally adding each number divided by its factorial up to 'n', where factorials are calculated within a nested loop structure.

Code

```
#include <iostream>
#include <iomanip>

using namespace std;

int main()
{
    int n;
    cout << "Enter the value of n: ";
    cin >> n;
    float sum = 0;

    for(int i = 1; i <= n; i++)
    {
        float fact = 1;

        for(int j = i; j >= 1; j--)
        {
            fact *= j;
        }

        sum += i/fact;
    }

    cout << "Sum of the series is: " << setprecision(3) << sum << endl;
}
```

Output

```
Enter the value of n: 6  
Sum of the series is: 2.72
```

Test Cases

1)

```
Desktop/Cpp_257/_1/SumOfSeriesw  
Enter the value of n: 4  
Sum of the series is: 2.67
```

2)

```
Enter the value of n: 8  
Sum of the series is: 2.72
```

Conclusion

Calculates the sum of a series based on factorials of numbers up to 'n' and displays the result with precision set to 3 decimal places.

Name of Student: Manmeet Singh

Roll Number: 16

Experiment No: 10

Title: Write a program to create an array of strings and display them in alphabetical order