K. R. MANGALAM UNIVERSITY

THE COMPLETE WORLD OF EDUCATION

Object Oriented Programming Using C++ (ENCS-102)

Assignment Submitted to

K. R. Mangalam University

for

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

Shikha Sharma (2301010087)

Course Teacher

Mr. Shamim Ahmad

School of Engineering & Technology

K. R. MANGALAM UNIVERSITY

Sohna, Haryana 122103, India

Index

Function Questions

Q1. Write a program that calculates 6^5. Declare your own function to do this.

Code:

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

int power(int x, int y)
{
    int i = 0;
    int pow = 1;
    while (i < y)
    {
        pow = pow * x;
        i++;
    }
    return pow;
}
int main()
{
    int n1, n2;
    n1 = 6;
    n2 = 5;
    cout << "The power of 6^5 is : " << power(n1, n2);
}
```

Output:

```
The power of 6^5 is : 7776
```

Q2. Write a program that asks a name to say hello. Use your own function, that receives a string of characters (name) and prints on screen the hello message. (Doesn't returns anything- void type)

Code:

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

void greet(string x)
{
    cout << "Hello " << x;
}

int main()
{
    string name;
    cout << "Enter your name: ";
    cin >> name;
    greet(name);
}
```

Output:

```
Enter your name: Akash
Hello Akash
```

Q3. Write a program that asks for two numbers, compare them and show the maximum. Declare a function called max_two that compares the numbers and returns the maximum.

Code:

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

int max_two(int x, int y)
{
    (x > y) ? cout << x << " is greater." : cout << y << " is greater.";
}
int main()
{
    int a, b;
```

```
    cout << "Enter two numbers: ";
    cin >> a >> b;
    max_two(a, b);
}
```

Output:

```
Enter two numbers: 23 45
45 is greater.
```

Q4. Write a program that asks the user for an integer number and find the sum of all natural numbers upto that number.

Code:

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

int sumN(int x)
{
    int sum = 0;
    for (int i = 1; i <= x; i++)
    {
        sum = sum + i;
    }
    return sum;
}

int main()
{
    int a;
    cout << "Enter the number: ";
    cin >> a;
    cout << "The sum of all natural numbers till " << a << " is : " <<
sumN(a);
}
```

Output:

```
Enter the number: 10
The sum of all natural numbers till 10 is : 55
```

Q5. Write a program that performs arithmetic division. The program will use two integers, a and b (obtained by the user) and will perform the division a/b, store the result in another integer c and show the result of the division using cout. In a similar way, extend the program to add, subtract, multiply, do modulo and power using integers a and b. Modify your program so that when it starts, it asks the user which type of calculation it should do, then asks for the 2 integers, then runs the user selected calculation and outputs the result in a user friendly formatted manner.

Code:

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

int sum(float x, float y)
{
    return x + y;
}
float subtract(float x, float y)
{
    return x - y;
}
float multiply(float x, float y)
{
    return x * y;
}
int modulo(int x, int y)
{
    return x % y;
}
```

```
int pow(int x, int y)
{
    int p = 1, i = 0;
    while (i < y)
    {
        p = p * x;
        i++;
    }
    return p;
}
int main()
{
    int n;
    float a, b;
    cout << "Choose one of the operations." << endl;
    cout << "_____\n";
    cout << "1 - Addition\n2 - Subtraction\n3 - Multiplication\n4 -
Remainder\n5 - Power\n6 - Exit" << endl;
    cout << "_____\n";
    cin >> n;
    cout << "Enter two numbers : ";
    cin >> a >> b;
    switch (n){
    case 1:
        cout << "The Addition of " << a << " and " << b << " = " << sum(a,
b) << endl;
        break;

    case 2:
        cout << "The Subtraction of " << a << " and " << b << " = " <<
subtract(a, b) << endl;
        break;

    case 3:
```

```
        cout << "The Multiplication of " << a << " and " << b << " = " <<
multiply(a, b) << endl;
        break;

    case 4:
        cout << "The Remainder of " << a << " and " << b << " = " <<
modulo(a, b) << endl;
        break;
    case 5:
        cout << "The Power of " << a << " and " << b << " = " << pow(a, b)
<< endl;
        break;
    case 6:
        return 0;
        break;
    default:
        cout << "Press a valid number!" << endl;
        break;
    }
}
```

Output:

```
Choose one of the operations.
----------------------------------------------------------------
1 - Addition
2 - Subtraction
3 - Multiplication
4 - Remainder
5 - Power
6 - Exit
----------------------------------------------------------------
3
Enter two numbers : 12 45
The Multiplication of 12 and 45 = 540
```

Q6. Basically the same as exercise 5, but this time, the function that adds the numbers should be void, and takes a third, pass by reference parameter; then puts the sum in that.

Code:

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

int sum(float x, float y, float &result)
{
    result = x + y;
}
float subtract(float x, float y)
{
    return x - y;
}
float multiply(float x, float y)
{
    return x * y;
}
int modulo(int x, int y)
{
    return x % y;
}
int pow(int x, int y)
{
    int p = 1, i = 0;
    while (i < y)
    {
        p = p * x;
        i++;
    }
    return p;
}
int main()
```

```
{
    int n;
    float a, b, result = 0;
    cout << "Choose one of the operations." << endl;
    cout << "_____\n";
    cout << "1 - Addition\n2 - Subtraction\n3 - Multiplication\n4 -
Remainder\n5 - Power\n6 - Exit" << endl;
    cout << "_____\n";
    cin >> n;
    cout << "Enter two numbers : ";
    cin >> a >> b;
    switch (n)
    {
    case 1:
        sum(a, b, result);
        cout << "The Addition of " << a << " and " << b << " = " << result
<< endl;
        break;

    case 2:
        cout << "The Subtraction of " << a << " and " << b << " = " <<
subtract(a, b) << endl;
        break;

    case 3:
        cout << "The Multiplication of " << a << " and " << b << " = " <<
multiply(a, b) << endl;
        break;

    case 4:
        cout << "The Remainder of " << a << " and " << b << " = " <<
modulo(a, b) << endl;
        break;

    case 5:
```

```
        cout << "The Power of " << a << " and " << b << " = " << pow(a, b)
<< endl;
        break;

    case 6:
        return 0;
        break;

    default:
        cout << "Press a valid number!" << endl;
        break;
    }
}
```

Output:

```
Choose one of the operations.
---------------------------------------------------------------
1 - Addition
2 - Subtraction
3 - Multiplication
4 - Remainder
5 - Power
6 - Exit
---------------------------------------------------------------
1
Enter two numbers : 32 45
The Addition of 32 and 45 = 77
```

Q7. Write a recursive function that finds the #n integer of the Fibonacci sequence. Then build a minimal program to test it. For reference see Wikipedia : Fibonacci number.

For any possible natural number "n", the following applies fib(n+2)= fib(n+1)+ fib(n). Also, the following are predefined fib(0)=0 fib(1)=1

Code:

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

```
int fibbo(int x)
{
    if (x <= 0 || x == 1)
    {
        return x;
    }
    return fibbo(x - 1) + fibbo(x - 2);
}

int main()
{
    int n;
    cout << "Enter the number till you want the fibonacci numbers : ";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        cout << fibbo(i) << "\t";
    }
}
```

Output:

```
Enter the number till you want the fibonacci numbers : 10
0       1       1       2       3       5       8       13
21      34
```

Q8. Basically the same as exercise 7, although this time you mustn't use recursion.

For extra exercise, give a big number (like 1000000) to both exercise 3 and 4 solutions and compare the execution times. Ponder on the results.

Code:

```
#include <iostream>
using namespace std;
void fibo(int x)
{
    int a = 0, b = 1, c;
```

```
    cout << a << "\t" << b << "\t";
    for (int i = 2; i < x; i++)
    {
        c = a + b;
        a = b;
        b = c;
        cout << c << "\t";
    }
}

int main()
{
    int n;
    cout << "Enter the number till you want the fibonacci numbers:  ";
    cin >> n;
    fibo(n);
}
```

Output:

```
Enter the number till you want the fibonacci numbers:  12
0       1       1       2       3       5       8       13
21      34      55      89
```

Q9. Create a calculator that takes a number, a basic math operator (+,-,*,/,^) and a second number all from user input, and have it print the result of the mathematical operation. The mathematical operations should be wrapped inside of functions.

Code:

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

float add(float x, float y)
{
    return x + y;
}
```

```
float sub(float x, float y)
{
    return x - y;
}
float dot(float x, float y)
{
    return x * y;
}
float divide(float x, float y)
{
    return x / y;
}
int pow(int x, int y)
{
    int p = 1, i = 0;
    while (i < y)
    {
        p = p * x;
        i++;
    }
    return p;
}

int main()
{
    float a, b;
    char opr;
    cout << "Enter first number : ";
    cin >> a;
    cout << "Enter the operator (+,-,*,/,^): ";
    cin >> opr;
    cout << "Enter second number : ";
    cin >> b;
    if (opr == '+')
    {
```

```
        cout << "The sum of " << a << " and " << b << " is : " << add(a,
b) << endl;
    }
    else if (opr == '-')
    {
        cout << "The subtraction of " << a << " and " << b << " is : " <<
sub(a, b) << endl;
    }
    else if (opr == '*')
    {
        cout << "The multiplication of " << a << " and " << b << " is : "
<< dot(a, b) << endl;
    }
    else if (opr == '/')
    {
        cout << "The division of " << a << " and " << b << " is : " <<
divide(a, b) << endl;
    }
    else if (opr == '^')
    {
        cout << "The power of " << a << " and " << b << " is : " << pow(a,
b) << endl;
    }
    else
    {
        cout << "Enter a valid operator!" << endl;
    }
}
```

Output:

```
Enter first number : 12
Enter the operator (+,-,*,/,^): ^
Enter second number : 2
The power of 12 and 2 is : 144
```

Q10. Write a program to print the sum of two numbers entered by the user by defining your own function.

Code:

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

int sum(int x, int y)
{
    return x + y;
}

int main()
{
    int a, b;
    cout << "Enter two number to add : ";
    cin >> a >> b;
    cout << "The sum of " << a << " and " << b << " is : " << sum(a, b) <<
endl;
}
```

Output:

```
Enter two number to add : 12 34
The sum of 12 and 34 is : 46
```

Q11. Define a function that returns the product of two numbers entered by the user.

Code:

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

int prod(int x, int y)
{
    return x * y;
}
int main()
```

```
{
    int a, b;
    cout << "Enter two number to Multiply : ";
    cin >> a >> b;
    cout << "The product of " << a << " and " << b << " is : " << prod(a,
b) << endl;
}
```

Output:

```
Enter two number to Multiply : 12 6
The product of 12 and 6 is : 72
```

Q12. Write a program to print the circumference and area of a circle of radius entered by the user by defining your own function.

Code:

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

float area(float x)
{
    return 3.14 * x * x;
}
float circum(float x)
{
    return 2 * 3.14 * x;
}

int main()
{
    float radi;
    cout << "Enter the radius : ";
    cin >> radi;
    cout << "The area of the circle is : " << area(radi) << endl;
    cout << "The circumference of the circle is : " << circum(radi) <<
endl;
}
```

Output:

```
Enter the radius : 5
The area of the circle is : 78.5
The circumference of the circle is : 31.4
```

Q13. Define two functions to print the maximum and the minimum number respectively among three numbers entered by the user.

Code:

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

int max(int x, int y, int z){
    // nested ternary oprs.
    int maxum = (x > y && x > z) ? x : (y > x && y > z) ? y : (z > x && z > y) ? z : 0;
    return maxum;
}
int min(int x, int y, int z){
    int minum = (x < y && x < z) ? x : (y < x && y < z) ? y : (z < x && z < y) ? z : 0;
    return minum;
}
int main()
{
    int a, b, c;
    cout << "Enter three numbers : ";
    cin >> a >> b >> c;
    cout << "The maximum number is : " << max(a, b, c) << endl;
    cout << "The minimum number is : " << min(a, b, c) << endl;
}
```

Output:

```
Enter three numbers : 123 43 310
The maximum number is : 310
The minimum number is : 43
```

Q14. Define a program to find out whether a given number is even or odd.

Code:

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

int main()
{
    int a;
    cout << "Enter the number : ";
    cin >> a;
    if (a % 2 == 0)
        cout << a << " is an Even number.";
    else
        cout << a << " is an Odd number.";
}
```

Output:

```
Enter the number : 12
12 is an Even number.
```

Q15. A person is eligible to vote if his/her age is greater than or equal to 18. Define a function to find out if he/she is eligible to vote.

Code:

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

void eligible(int x)
{
    (x >= 18) ? cout << "You are eligible to vote." << endl : cout << "You
are not eligible to vote." << endl;
}

int main()
{
    int n;
```

```
    while (true)
    {
        cout << "Enter your age : ";
        cin >> n;
        if (n == 0)
            break;
        eligible(n);
        cout << "Press 0 to exit!" << endl;
    }
}
```

Output:

```
Enter your age : 15
You are not eligible to vote.
Press 0 to exit!
Enter your age : 32
You are eligible to vote.
Press 0 to exit!
Enter your age : 0
```

Q16. Define a function to find out if a number is prime or not.

Code:

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

void isPrime(int x)
{
    int count = 0;
    for (int i = 1; i <= x; i++)
    {
        if (x % i == 0)
        {
            count++;
        }
    }
    if (count == 2)
```

```
        cout << x << " is a prime number." << endl;
    else
        cout << x << " is not a prime number." << endl;
}
int main()
{
    int  n;
    while (true)
    {
        cout << "Enter the number : ";
        cin >> n;
        if (n == 0)
            break;
        isPrime(n);
        cout << "Press 0 for exit!" << endl;
    }
}
```

Output:

```
Enter the number : 12
12 is not a prime number.
Press 0 for exit!
Enter the number : 23
23 is a prime number.
Press 0 for exit!
Enter the number : 0
```

Q17. Write a program which will ask the user to enter his/her marks (out of 100). Define a function that will display grades according to the marks entered as below:

Marks	Grade
91-100	AA
81-90	AB
71-80	BB
61-70	BC

51-60 CD

41-50 DD

<=40 Fail

Code:

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

void grade(int x)
{
    if (x >= 91 && x <= 100)
    {
        cout << "According to your marks your grade is : "
             << "AA" << endl;
    }
    else if (x >= 81 && x <= 90)
    {
        cout << "According to your marks your grade is : "
             << "AB" << endl;
    }
    else if (x >= 71 && x <= 80)
    {
        cout << "According to your marks your grade is : "
             << "BB" << endl;
    }
    else if (x >= 61 && x <= 70)
    {
        cout << "According to your marks your grade is : "
             << "BC" << endl;
    }
    else if (x >= 51 && x <= 60)
    {
        cout << "According to your marks your grade is : "
             << "CD" << endl;
    }
```

```
    else if (x >= 41 && x <= 50)
    {
        cout << "According to your marks your grade is : "
             << "DD" << endl;
    }
    else if (x <= 40 && x > 0)
    {
        cout << "According to your marks you are 'Fail'!! " << endl;
    }
    else
    {
        cout << "Invalid marks, enter marks between (1 - 100)!" << endl;
    }
}

int main()
{
    int num;
    while (true)
    {
        cout << "Enter your marks between (1 - 100) : ";
        cin >> num;
        if (num == 0)
            break;
        grade(num);
    }
}
```

Output:

```
Enter your marks between (1 - 100) : 80
According to your marks your grade is : BB
Enter your marks between (1 - 100) : 92
According to your marks your grade is : AA
Enter your marks between (1 - 100) : 0
```

Q18. Write a program to print the factorial of a number by defining a function named 'Factorial'.

Factorial of any number n is represented by n! and is equal to 1*2*3* . *(n-1)*n. E.g.-

4! = 1*2*3*4 = 24

3! = 3*2*1 = 6

2! = 2*1 = 2

Also,

1! = 1

0! = 0

Code:

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

// Function to calculate factorial
unsigned long long Factorial(int n) {
    if (n == 0 || n == 1)
        return 1;
    else
        return n * Factorial(n - 1);
}

int main() {
    int num;
    cout << "Enter a number to find its factorial: ";
    cin >> num;

    cout << "The factorial of " << num << " is " << Factorial(num) << endl;

    return 0;
}
```

Output:

```
Enter a number to find its factorial: 5
The factorial of 5 is 120
```

Q19. Write a program in C++ to show the simple structure of a function.

Code:

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

void myFunction(); // Function declaration

int main()
{
    myFunction(); // Function call
    return 0;
}

void myFunction() // Function definition
{
    // Function body
    cout << "This is my function!" << endl;
}
```

Output:

```
This is my function!
```

Q20. Write a program in C++ to find the square of any number using the function.

Test Data :

Input any number for square : 20

Expected Output :

The square of 20 is : 400.00

Code:

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

using namespace std;

float square(float x)
{
    return x * x;
}
int main()
{
    float n;
    cout << "Enter any number: ";
    cin >> n;
    float res = square(n);
    cout << "Square : " << fixed << setprecision(2) << res;
}
```

Output:

```
Enter any number: 20
Square : 400.00
```

Q21. Write a program in C++ to swap two numbers using a function.

Code:

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

```
void swap(int x, int y)
{
    int z;
    z = x;
    x = y;
    y = z;
    cout << "After swapping : n1 = " << x << "," << "n2 = " << y;
}
int main()
{
    int a, b;
    cout << "Enter the value of n1 and n2 : ";
    cin >> a >> b;
    cout << "Before Swapping : n1 = " << a << "," << "n2 = " << b;
    cout << endl;
    swap(a, b);
}
```

Output:

```
Enter the value of n1 and n2 : 4 5
Before Swapping : n1 = 4,n2 = 5
After swapping : n1 = 5,n2 = 4
```

Q22. Write a program in C++ to check if a given number is even or odd using the function.

Code:

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

int oddEven(int x)
{
    return (x % 2 == 0) ? 1 : 0;
}
int main()
{
    int n;
```

```
    while (true)
    {
        cout << "Enter the number : ";
        cin >> n;
        if (oddEven(n))
        {
            cout << n << " is an Even number." << endl;
        }
        else
        {
            cout << n << " is an Odd number." << endl;
        }
    }
}
```

Output:

```
Enter the number : 23
23 is an Odd number.
Enter the number : 44
44 is an Even number.
```

Q23.Write a program in C++ to find the sum of the series 1!/1+2!/2+3!/3+4!/4+5!/5 using the function.

Expected Output :

The sum of the series is : 34

Code:

```
#include <iostream>
using namespace std;
int series(int x)
{
    int fact = 1;
    int sum = 0;
    for (int i = 1; i <= x; i++)
    {
        fact = fact * i;
        sum = sum + (fact / i);
```

```
    }
    return sum;
}

int main()
{
    int n;
    cout << "Enter the last digit of the series : ";
    cin >> n;
    cout << "The sum of the series is : " << series(n) << endl;
}
```

Output:

```
Enter the last digit of the series : 5
The sum of the series is : 34
```

Q24. Write a program in C++ to convert a decimal number to a binary number using the function.

Code:

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

void bin(int x)
{
    int binNum[32];
    int i = 0;
    while (x > 0)
    {
        binNum[i] = x % 2;
        x = x / 2;
        i++;
    }
    for (int j = i - 1; j >= 0; j--)
    {
        cout << binNum[j];
    }
```

```
}
int main()
{
    int n;
    cout << "Enter the decimal form of number : ";
    cin >> n;
    cout << "The binary representation of decimal number " << n << " is :
0b";
    bin(n);
}
```

Output:

```
Enter the decimal form of number : 65
The binary representation of decimal number 65 is : 0b1000001
```

Q25. Write a program in C++ to check whether a number is a prime number or not using the function.

Code:

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

bool isPrime(int n) {
    if (n <= 1) {
        return false;
    }
    for (int i = 2; i * i <= n; i++) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}

int main() {
    int number;
    cout << "Enter the number: ";
```

```
    cin >> number;

    if (isPrime(number)) {
        cout << number << " is a Prime number!";
    } else {
        cout << number << " is not a Prime number!";
    }
    return 0;
}
```

Output:

```
Enter the number: 78
78 is not a Prime number!
```

Q26. Write a program in C++ to get the largest element of an array using the function.

Test Data :

Input the number of elements to be stored in the array : 5

Input 5 elements in the array :

element – 0 : 1

element – 1 : 2

element – 2 : 3

element – 3 : 4

element – 4 : 5

Expected Output :

The largest element in the array is : 5

Code:

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

int maxum(int x)
{
    int maxNum[x];
```

```
    cout << "Enter the elements of the array : ";
    for (int i = 0; i < x; i++)
    {
        cin >> maxNum[i];
    }
    int max = maxNum[0];
    for (int i = 0; i < x; i++)
    {
        cout << "Element - " << i << " : " << maxNum[i] << endl;
        if (max < maxNum[i])
            max = maxNum[i];
    }
    return max;
}

int main()
{
    int n;
    cout << "Enter the size of array : ";
    cin >> n;
    cout << "The largest number in the array is : " << maxum(n);
}
```

Output:

```
Enter the size of array : 5
Enter the elements of the array : 34 3 67 45 1
Element - 0 : 34
Element - 1 : 3
Element - 2 : 67
Element - 3 : 45
Element - 4 : 1
The largest number in the array is : 67
```

Q27. Write a program in C++ to check Armstrong and Perfect numbers using the function.

Code:

```
#include <iostream>
```

```
using namespace std;

int perfect(int x)
{
    int sum = 0, num = x;
    for (int i = 1; i <= x / 2; i++)
    {
        if (x % i == 0)
        {
            sum = sum + i;
        }
    }
    return (x == sum);
}

int armstrng(int x)
{
    int sum = 0, num, rem;
    num = x;
    while (num != 0)
    {
        rem = num % 10;
        sum = sum + (rem * rem * rem);
        num = num / 10;
    }
    return (x == sum);
}

int main()
{
    int n;
    cout << "Enter the number : ";
    cin >> n;
    if (perfect(n))
    {
```

```
        cout << n << " is a Perfect number." << endl;
    }
    else
    {
        cout << n << " is not a Perfect number." << endl;
    }

    if (n < 10 && n > 0)
    {
        cout << n << " is an Armstrong number." << endl;
    }
    else if (armstrng(n))
    {
        cout << n << " is an Armstrong number." << endl;
    }
    else
    {
        cout << n << " is not an Armstrong number." << endl;
    }
}
```

Output:

```
Enter the number : 371
371 is not a Perfect number.
371 is an Armstrong number.
```

Q28. Write a program in C++ to print all perfect numbers in a given range using the function.

Code:

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

void perfect(int x)
{
    int number = 1, i, sum;
```

```
    while (number <= x)
    {
        sum = 0;
        for (i = 1; i <= number / 2; i++)
        {
            if (number % i == 0)
            {
                sum = sum + i;
            }
        }
        if (sum == number)
        {
            cout << number << " is a perfect number." << endl;
        }
        number++;
    }
}

int main()
{
    int n;
    cout << "Enter the number till you want perfect numbers : ";
    cin >> n;
    perfect(n);
}
```

Output:

```
Enter the number till you want perfect numbers : 10000
6 is a perfect number.
28 is a perfect number.
496 is a perfect number.
8128 is a perfect number.
```

Q29. Write a program in C++ to check whether two given strings are an anagram.

Test Data :

Input the first String : spare

Input the second String : pears

Expected Output :

spare and pears are Anagram.

Code:

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

void anagram(char x[20], char y[20])
{
    int len1, len2, i, j, found = 0, not_found = 0;
    len1 = strlen(x);
    len2 = strlen(y);

    if (len1 == len2)
    {
        for (i = 0; i < len1; i++)
        {
            found = 0;
            for (j = 0; j < len1; j++)
            {
                if (x[i] == y[j])
                {
                    found = 1;
                    break;
                }
            }
            if (found == 0)
            {
                not_found = 1;
            }
```

```
        }
        if (not_found == 1)
        {
            cout << "Strings are not anagram.";
        }
        else
        {
            cout << "Strings are anagram.";
        }
    }
    else
    {
        cout << "Length of strings should be same!!";
    }
}

int main()
{
    char a[20], b[20];
    cout << "Enter the first string : ";
    cin >> a;
    cout << "Enter the second string : ";
    cin >> b;
    anagram(a, b);
}
```

Output:

```
Enter the first string : spare
Enter the second string : pears
Strings are anagram.
```

Q30. Write a C++ program to find the maximum and minimum of some values using a function that returns an array.

Code:

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

```cpp
void maxim(int x)
{
    int maxNum[x];
    cout << "Enter the elements of the array : ";
    for (int i = 0; i < x; i++)
    {
        cin >> maxNum[i];
    }
    int max = maxNum[0], min = maxNum[0];
    for (int i = 0; i < x; i++)
    {
        if (max < maxNum[i])
            max = maxNum[i];

        if (min > maxNum[i])
            min = maxNum[i];
    }
    cout << "The maximum value in the array is : " << max << endl;
    cout << "The minimum value in the array is : " << min << endl;
}

int main(){
    int n;
    cout << "Enter the size of array : ";
    cin >> n;
    maxim(n);
}
```

Output:

```
Enter the size of array : 5
Enter the elements of the array : 23 45 2 78 98
The maximum value in the array is : 98
The minimum value in the array is : 2
```