1. Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called power ( ) that takes a double value for n and an int value for p, and returns the result as double value. Use a default argument of 2 for p, so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function.

```
#include<iostream.h>
#include <conio.h>
double power(double n, int p=2)
{
double res=1;
for(int i=1; i<=p;i++)
res=res*n;
return res;
}
int main()
{
double n,result;
int p;
cout<<"enter n and p value"<<endl;
cin>>n>>p;
result=power(n,p);
cout<<n<<" raised to the power "<<p<<" is "<<result<<endl;
result=power(n);
cout<<" By default argument"<<n<<" power 2 is "<<result<<endl;
getch();
return 0;
}
```

2.Write a C++ Program to accept an alphabet and check whether it is a vowel or a consonant. If it is a vowel, return its predecessor, else its successor. Use call-by-reference with reference arguments.

```

#include <iostream.h>
#include <conio.h>
void checkAlphabet(char alphabet,char& result)
{
if(alphabet=='a' || alphabet == 'e' ||
alphabet=='i'||alphabet=='o'||alphabet=='u'||
```

```
alphabet=='A'||alphabet=='E'||alphabet=='O'||alphabet=='U'||alphabet=='I')
{
result=alphabet-1;
cout<<alphabet<<"is vowel"<<"its predeccessor is"<<result<<endl;
}else{
result=alphabet+1;
cout<<alphabet<<"is consonant"<<"its successor is"<<result<<endl;
}
}
int main(){

char alphabet,result;
cout<<"Enter an alphabet:";

cin>>alphabet;
if(!((alphabet>='a'&&alphabet<='z')||(alphabet>='A' && alphabet<='Z')))
cout<<"enter character is not a alphabet"<<endl;
else
checkAlphabet(alphabet,result);
getch();
return 0;
}
```

3. Write a C++ Program to call a C function using an extern “C” linkage directive.Use compound statement linkage directive for #include.

```
#include <iostream.h>
#include <math.h>
#include <conio.h>
extern "C" {
double my_sqrt(double x) {
return sqrt(x);
}
}
int main() {
double x = 100;
double y = my_sqrt(x);
cout << "Square root of " << x << " is " << y << endl;
```

```
getch();
return 0;}
```

4. Write a C++ Program to accept a line of text and count the number of words,characters and digits in it.

```
#include <iostream.h>
#include <string.h>
#include <ctype.h>
#include <conio.h>
int main()
{
char line[500];
int wordCount = 0, charCount = 0, digitCount = 0;
cout << "Enter a line of text: ";
cin.getline(line,500);
int length = strlen(line);
for (int i = 0; i < length; i++)
{
if (isalpha(line[i]))
{
charCount++;
}
else if (isdigit(line[i]))
{
digitCount++;
}
if (isspace(line[i]) && !isspace(line[i-1]))
{
wordCount++;
}
}
// count the last word if it exists
if (length>0 && !isspace(line[length-1]))
{
wordCount++;
}
cout << "Word count: " << wordCount << endl;
cout << "Character count: " << charCount << endl;
```

```
cout << "Digit count: " << digitCount << endl;
getch();
return 0;
}
```

5. Write a C++ Program to store two binary numbers in arrays and perform bitwise AND, OR and XOR operations on these two numbers.

```
#include <iostream.h>
#include <conio.h>

#define MAX_SIZE 100 // Adjust this as needed

void printArray(int arr[], int n) {
    for (int i = 0; i < n; i++) {
    cout << arr[i] << " ";
    }
    cout << endl;
}

void andOperation(int arr1[], int arr2[], int n) {
    int result[MAX_SIZE];
    for (int i = 0; i < n; i++) {
    result[i] = arr1[i] & arr2[i];
    }
    cout << "Bitwise AND: ";
    printArray(result, n);
}

void orOperation(int arr1[], int arr2[], int n) {
    int result[MAX_SIZE];
    for (int i = 0; i < n; i++) {
    result[i] = arr1[i] | arr2[i];
    }
    cout << "Bitwise OR: ";
    printArray(result, n);
}
```

```
void xorOperation(int arr1[], int arr2[], int n) {
    int result[MAX_SIZE];
    for (int i = 0; i < n; i++) {
    result[i] = arr1[i] ^ arr2[i];
    }
    cout << "Bitwise XOR: ";
    printArray(result, n);
}

int main() {
    int n;
    cout << "Enter the number of bits (max " << MAX_SIZE << "): ";
    cin >> n;

    if (n > MAX_SIZE) {
    cout << "Error: Maximum size exceeded." << endl;
    return 1;
    }

    int arr1[MAX_SIZE], arr2[MAX_SIZE];
    cout << "Enter the first binary number: ";
    for (int i = 0; i < n; i++) {
    cin >> arr1[i];
    }
    cout << "Enter the second binary number: ";
    for (int in = 0; in < n; in++) {
    cin >> arr2[in];
    }

    cout << "Binary 1=";
    printArray(arr1, n);
    cout << "Binary 2=";
    printArray(arr2, n);

    andOperation(arr1, arr2, n);
    orOperation(arr1, arr2, n);
    xorOperation(arr1, arr2, n);
    getch();
    return 0;
}
```

6.Create two classes D1 and D2 which store the value of distances. D1 stores distances in meters and centimeters and D2 in feet and inches. Write a program to add objects of two classes D1 and D2 and the display the results in feet and inches using friend function.

```
#include <iostream.h> // Include iostream.h for Turbo C++
#include <conio.h>    // Include conio.h for getch()
class D2;             // Forward declaration

class D1 {
private:
    int meters;
    int centimeters;

public:
    void get_data() {
        cout << "Enter distance in meters and centimeters: ";
        cin >> meters >> centimeters;
    }
    friend void add(D1 obj1, D2 obj2);
};

class D2 {
private:
    int feet;
    int inches;

public:
    void get_data() {
        cout << "Enter distance in feet and inches: ";
        cin >> feet >> inches;
    }
    friend void add(D1 obj1, D2 obj2);
};

void add(D1 obj1, D2 obj2) {
    // Convert D1 to inches
    int d1_inches = obj1.meters * 39.37 + obj1.centimeters * 0.3937;
    // Add D1 and D2 in inches
    int total_inches = d1_inches + obj2.feet * 12 + obj2.inches;
    // Convert total inches to feet and inches
```

```
    int feet = total_inches / 12;
    int inches = total_inches % 12;
    // Display the result in feet and inches
    cout << "Total distance: " << feet << " feet, " << inches << " inches"
<< endl;
}

void main() {
    D1 d1;
    D2 d2;
    clrscr(); // Clear the screen (specific to Turbo C++)
    d1.get_data();
    d2.get_data();
    // Add the two objects and display the result
    add(d1, d2);
    getch(); // Pause the screen until a key is pressed
}
```

7. Given that an EMPLOYEE class contains the data members like E_Number, E_Name, Basic_salary, DA, HRA, Net_salary and the member functions like Read(), Calculate_Net_Sal(), and Display(). Write a C++ Program to read the data of N Employees and Compute the Net_Salary of each employee.

```
#include <iostream.h>  // Turbo C++ uses iostream.h
#include <conio.h>     // Include conio.h for clrscr() and getch()

class employee {
    int emp_number;
    char emp_name[20];
    float emp_basic;
    float emp_da;
    float emp_hra;
    float emp_net_sal;

public:
    void read();
    float calculate_net_salary();
    void display();
};
```

```
void employee::read() {
    cout << "\nEnter employee number: ";
    cin >> emp_number;
    cout << "\nEnter employee name: ";
    cin >> emp_name;
    cout << "\nEnter employee basic: ";
    cin >> emp_basic;
    cout << "\nEnter employee DA: ";
    cin >> emp_da;
    cout << "\nEnter employee HRA: ";
    cin >> emp_hra;
}

float employee::calculate_net_salary() {
    emp_net_sal = emp_basic + emp_da + emp_hra;
    return emp_net_sal;
}

void employee::display() {
    cout << "\n\n**** Details of Employee ****";
    cout << "\nEmployee Name : " << emp_name;
    cout << "\nEmployee number : " << emp_number;
    cout << "\nBasic salary : " << emp_basic;
    cout << "\nEmployee DA : " << emp_da;
    cout << "\nEmployee HRA : " << emp_hra;
    cout << "\nNet Salary : " << emp_net_sal;
    cout << "\n-------------------------------\n\n";
}

int main() {
    clrscr();  // Clear the screen
    int num_employees, i;
    cout << "How many employees do you want to enter? ";
    cin >> num_employees;

    employee emp[10];  // Use a fixed-size array since Turbo C++ doesn't
support dynamic arrays
```

```
    for (i = 0; i < num_employees; i++) {
    emp[i].read();
    emp[i].calculate_net_salary();
    }

    for (i = 0; i < num_employees; i++) {
    emp[i].display();
    }

    getch();
    return 0;  // Pause the screen until a key is pressed
}
```

8. Write a C++ program to overload the function Search() to search an integer key value and a key value of type double.

```
#include <iostream.h>
#include <conio.h>
// Function to search an integer key value
int Search(int arr[], int size, int key) {
for (int i = 0; i < size; i++) {
if (arr[i] == key) {
return i;
}
}
return -1;
}
// Function to search a key value of type double
int Search(double arr[], int size, double key) {
for (int i = 0; i < size; i++) {
if (arr[i] == key) {
return i;
}
}
return -1;
}
int main() {
int intArr[] = { 10, 20, 30, 40, 50 };
```

```
double doubleArr[] = { 1.1, 2.2, 3.3, 4.4, 5.5 };
// Searching for an integer key value
int intIndex = Search(intArr, 5, 30);
if (intIndex != -1) {
cout << "Integer key value found at index " << intIndex << endl;
}
else {
cout << "Integer key value not found" << endl;
}
// Searching for a key value of type double
int doubleIndex = Search(doubleArr, 5, 3.3);
if (doubleIndex != -1) {
cout << "Double key value found at index " << doubleIndex << endl;
}
else {
cout << "Double key value not found" << endl;
}
getch();
return 0;
}
```

9. Write a C++ program to nd the following using Function Template a) Successor value of any input of type integer, oat, char and double. b) Sum of all the elements of an array of integers or oats or doubles.

```
#include <iostream.h>
#include <conio.h>
template <class T>
T successor(T x) {
return x + 1;
}
template <class X>
X sum(X arr[], int size) {
X total = 0;
for (int i = 0; i < size; i++) {
total += arr[i];
}
return total;
}
```

```
int main() {
// Finding the successor value of any input
cout << "Successor of 5: " << successor(5) << endl;
cout << "Successor of 5.5: " << successor(5.5) << endl;
cout << "Successor of 'a': " << successor('a') << endl;
// Finding the sum of all the elements of an array
int int_arr[] = {1, 2, 3, 4, 5};
float float_arr[] = {1.5, 2.5, 3.5, 4.5, 5.5};
double double_arr[] = {1.0, 2.0, 3.0, 4.0, 5.0};
int int_arr_size = sizeof(int_arr) / sizeof(int);
int float_arr_size = sizeof(float_arr) / sizeof(float);
int double_arr_size = sizeof(double_arr) / sizeof(double);
cout << "Sum of int array: " << sum(int_arr, int_arr_size) << endl;
cout << "Sum of float array: " << sum(float_arr, float_arr_size) << endl;
cout << "Sum of double array: " << sum(double_arr, double_arr_size) <<
endl;
getch();
return 0;
}
```

10.Write a C++ Program to create a class as COMPLEX and implement the following by overloading the function ADD() which returns the Complex numbers a) ADD(C1, C2); C1 is an integer ; C2 is a Complex number. b) ADD(C1, C2); C1 and C2 are Complex numbers.

```
#include <iostream.h>
#include <conio.h>
#include <stdlib.h> // For abs()

class COMPLEX {
private:
    float real;
    float imag;

public:
    COMPLEX() { // Default constructor
        real = 0;
        imag = 0;
    }
```

```
    COMPLEX(float r, float i) { // Parameterized constructor
        real = r;
        imag = i;
    }

    COMPLEX(int r) { // Constructor with integer input
        real = r;
        imag = 0;
    }

    COMPLEX ADD(COMPLEX C) { // ADD() function for adding two Complex
numbers
        COMPLEX res;
        res.real = real + C.real;
        res.imag = imag + C.imag;
        return res;
    }

    COMPLEX ADD(int r) { // ADD() function for adding an integer and a
Complex number
        COMPLEX res;
        res.real = real + r;
        res.imag = imag;
        return res;
    }

    void display() { // Function to display Complex number
        if (imag < 0)
            cout << real << " - i" << abs(imag) << endl;
        else
            cout << real << " + i" << imag << endl;
    }
};

int main() {
    COMPLEX C1(4, 5);
    COMPLEX C2(3, -2);
    COMPLEX C3;

    cout << "C1 = ";
```

```
    C1.display();

    cout << "C2 = ";
    C2.display();

    C3 = C1.ADD(C2); // Adding two Complex numbers
    cout << "C1 + C2 = ";
    C3.display();

    C3 = C1.ADD(2); // Adding an integer and a Complex number
    cout << "C1 + 2 = ";
    C3.display();

    getch();
    return 0;
}
```