3. Programs using Functions and Pointers

***CO1**: Impart the skills needed for Object – oriented programming and Console applications development*.

1. Define a function **hypotenuse** that calculates the hypotenuse of a right triangle when the other two sides are given. The function should take two double arguments and return the hypotenuse as a double. Test this function.

2. An integer is said to be a perfect number if the sum of its divisors, including 1 (but not the number itself), is equal to the number. For example, 6 is a perfect number, because 6 = 1 + 2 + 3. Write a function **isPerfect** that determines whether parameter number is a perfect number. Use this function prints all the perfect numbers in the range 1 and 1000.

3. Write a program with the following function prototype,
   a. ***void readArray(int a[],int size)*** – Function to read and store the data into an int array. The array to store the data and its size are passed as parameters. The parameter Size specifies the number of elements to be read.
   b. ***void printArray(int a[],int size)*** - Function to print the elements of an int array. The array to be printed and the number of elements are passed as parameters. The parameter Size specifies the number of elements to be printed
   c. ***int indexLargestElement(int a[],int size)*** - Function to find and return the index of the first largest element in an int array. The parameter Size specifies the number of elements in the array.
   d. ***void copyArray(int a[], int src, int b[],int tar, int numOfElements)*** - Function to copy some or all of the elements of one array into another array. Starting at the position specified by src, the elements of array 'a' are copied into array 'b' starting at the position specified by tar. The parameter numOfElements specifies the number of elements of array a to be copied into array b starting at the position specified by tar, the array b must have enough components to copy the elements of list1.

4. Write a program that uses a two-dimensional array to store the highest and lowest temperatures for each month of the year. The program should output the average high, average low and the highest and lowest temperatures for the year. Your program must consist of the following functions:
   a. Function ***getData*:** This function reads and stores data in the two dimensional array.
   b. Function ***printData*:** This function prints the data in the two dimensional array.

c. Function ***averageHigh***: This function calculates and returns the average high temperature for the year.

d. Function ***averageLow***: This function calculates and returns the average low temperature for the year.

e. Function ***indexHighTemp***: This function returns the index of the highest high temperature in the array.

f. Function ***indexLowTemp***: This function returns the index of the lowest low temperature in the array.

(These functions must all have the appropriate parameters.)

5. Write a ***recursive function*** power(base, exponent) that, when invoked, returns base $^{\text{exponent}}$. Test this function in main.

6. Write ***inline function*** to calculate the volume of cube and test it in main [volume=side^{3}].

7. Write a C++ program to read a matrix from the keyboard and display the same on the screen using functions. Make the row and column parameter of the matrix as a ***default argument*** so that it reads and prints a 2 x 2 matrix.

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

9. Write a program to declare and initialize a variable. Create a ***reference variable***. Display the value of actual variable using reference variable.

10. Write a C++ program to swap two numbers using ***reference variables***.

11. Write a function called **zero_small()** that has two integer arguments being ***passed by reference*** and sets the smaller of the two numbers to 0. Write the main program to access the function.

12. Create a method called **max().** ***Overload this method*** to find the maximum out of two integers, maximum out of two double values, maximum out of three integers, maximum out of n integers.

13. Write a program ***using pointers*** to dynamically allocate memory to store 'n' integer value. Randomly generate values in the range 1 to 10 and find the index of the last occurrence of the largest element in the array.