**Task 1:**

**void Power (double num, double exp, double &result);**

Calculate the power using built-in pow function of cmath library and print it in main using result reference variable.

**int Carrys (int num1, int num2);**

Calculate the count of carry generated in the process of addition of num1 and num2.

For example:

Num1 = 456

Num2 = 99

Number of Carry: 2

**Task 2:**

Declare an integer array of fixed size 5.

Give them random initial values (using random () function) from the range [5,25]. Print the elements of array on screen.

**Task 3:**

**You are asked to make some functions on Simple array.**

Write a C++ program in which write a function to print menu.

In menu following option should be given:

1. Insert
2. Delete
3. Search
4. Print
5. Exit

**The main function calls the function menu and asks the user to enter the choice from 1-5.**

1. If user enters 1. To **Insert**, User must be asked to enter a value to be inserted in array.
2. If user enters 2. To **Delete**, User must be asked to enter a value to be deleted from array.
3. If user enters 3. To **Search**, User must be asked to enter a value to be Search in array. If the Search function returns the valid index, the message should be printed that Value exists on the valid index.
4. If user enters 4. To **Print**, the content of whole array should be printed on console.
5. If user enters 5. To **Exit**, the content of whole array should be printed on console and program will terminate.

**You will repeat the above process until user enters 5.**

**Important Note:** Create an array of size/capacity 10. Make a variable named as capacity, initialize it with 10. And make another variable named as currentIndex and initialize it with zero.

Do not confuse **capacity** of **array** with **currentIndex**. Array **capacity** is fixed, whereas currentIndex may be any value from 0 to capacity-1 at a given time.

**Functions are given below:**

1. **bool Insert (int arr[], int capacity, int & currentIndex, int element);**

Insert the element at currentIndex. If currentIndex is valid.

For example: Array capacity is 10 and has already 4 elements then currentIndex will have value 4 and the next element will be inserted here. currentIndex will change with each successful insert operation.

If the array is already full, output an appropriate message “Sorry no room for insertion!”.

1. **bool Delete (int arr[], int & currentIndex, int key);**

First your program has to search this key/element in the array. Use the given below function to search element. It returns -1 if the element/key is not present in array. Otherwise it returns the index where the element/key exists.

1. **int Search (int arr[], int currentIndex, int key);**

**Delete** function calls the **Search** function if it returns the valid index, then delete it and currentIndex will change with each successful delete operation.

For example: Array is {2, 4, 5, 1, 8, 7} and currentIndex is 6.

User enters 1 to be deleted.

After delete then array becomes {2, 4, 5, 8, 7}, and currentIndex becomes 5.

**Hint:** Copy all elements one index back.

Another example: Array is {2, 4, 5, 8, 7} and currentIndex is 5.

User enters 3 to be deleted.

Now output will be “Specified element not found in Array!”.

In this case currentIndex will remain same.

1. **void print (int arr[], int currentIndex);**