Recursion
Due Date : Monday November 4th
No Later than 12:45 pm

Write a **C++** program that does the following :

1. Accepts array size from the keyboard. The size must be positive integer that is >= 5 and <= 15 inclusive .

2. Use the size from step 1 in order to create an integer array. Populate the created array with random integer values between 10 and 200.

3. Display the generated array.

4. Write a recursive function that displays the array in reverse order .

5. Write a recursive function that **returns** a result of a number raised to a power. The function should take two arguments, the number to be raised to the power is the last number in array and the power is 2.

6. Write a recursive function that displays squares of integers in ascending order, starting from 1 to the first number in the array.

7. Write a recursive function that displays the second number in the array vertically .

8. Write a recursive function that **returns** true if the digits of the third number in the array is in increasing order ; otherwise , the function returns false.

9. Write a recursive function that takes the fourth number in the array and **returns** the numbers with the digits reversed.

10. Write a recursive function that determines and **returns** true or false to whether or not the fifth number in the array is a prime number .

NOTES:

- Just one .cpp file with 7 individual recursive functions plus main for testing.
- Do not use global variable , global arrays or vector arrays, linked lists and pointers.
- Not allowed to use build in libraries such as power … etc
- Validation must be conducted on the arrays size. Size must be integer that is >= 5 and <= 15
- Your program's format and messages must match the output provided
- Replace My name (Husain Gholoom) with your first and last name.
- The program runs only once.

Sample Output

```
***  Welcome to My Recursion APP   ***

Enter The array size. ( Must be >= 5 ) ->     -5


Invalid arrays size. Size must >= 5.

Husain Gholoom  -  Programmer
November 2019
```

Sample Output

```
***  Welcome to My Recursion APP   ***

Enter The array size. ( Must be >= 5 ) ->     a


Invalid arrays size. Size must >= 5.

Husain Gholoom  - Programmer
November 2019
```

Sample Output

```
***  Welcome to My Recursion APP   ***

Enter The array size. ( Must be >= 5 ) ->     5


The generated array is   :  27  24  43  59  13

Reversed Array is  :  13   59   43   24   27

13  raised to the 2nd power is: 169

Table of square values  1  -   27

N     N Squared
1     1
2     4
3     9
4     16
5     25
6     36
7     49
8     64
9     81
10    100
11    121
12    144
13    169
14    196
15    225
16    256
17    289
18    324
19    361
20    400
21    441
22    484
23    529
24    576
25    625
26    676
```

```
27    729


24  vertically

2
4


43  is not in increasing order

59  reversed  is  95

13  is  Prime :


Husain Gholoom  - Programmer
November 2019
```

Style Guidelines:

At the beginning of your program (and **before** the #include statement), include the following :

Header comments (file documentation block) should be at the top of each file and should contain: Author / s, Due Date, Assignment Number, Course number and section, Instructor, and a brief description of the purpose of the code in the file. For example :

```
//      Serial Number   :       xxxxxxxxx
//
//      Author  : (Your name  here!!)

//      Due Date  :
//      Programming Assignment Number  5
//
//      Fall   2019   -   CS 3358    -    Your Section  Number
//
//      Instructor:    Husain Gholoom.
//
//       <Brief description of the  purpose of the program>
```

Variable names :

- Must be meaningful.
- The initial letter should be lowercase, following words should be capitalized, no other caps or punctuation (i.e. weightInPounds).
- Each variable must be declared on a separate line with a descriptive comment.

Named constants :

- Use for most numeric literals.
- All capitals with underscores (i.e. TX_STATE_SALES_TAX)
- Should occur at top of function, or global (only if necessary)

Line length of source code should be no longer than 80 characters (no wrapping of lines).

Indentation :

- Use 2-4 spaces (but be consistent throughout your program).
- Indent blocks, within blocks, etc.
- Use blank lines to separate sections.

Comments for variables :

All variable definitions should be commented as follows:

```
int  gender;    // integer value for the gender,
                        // 1 = Male , 2 = Female ,
```

Rules : In order to get a full mark :

1. Your program **must compile** and run using **latest version of codeblocks under windows. You are not allowed to us C++11, C++14 … etc.**
2. Your program must be **documented according the style above . See the website for the sample programming style program.**
3. Must **use functions (prototypes and definitions) with recursive calls .**
4. You must use the appropriate libraries in writing this program.
5. Must properly format the output as it is shown on the sample run above. Replace my name with your name
6. You must name your program as :
   - **PG5_F19_3358_1_LastName_FirstName.cpp**

   Where LastName is your Last Name and FirstName is your First Name.

   For example , the file name should look something like :
   PG5_F19_3358_1_Gholoom_Husain.cpp (not .cbp)

7. Everyone must upload the electronic version of the program no later than 12:45 pm on the due date. **No late assignments will be accepted.** **DO NOT** send your assignment solution via email.

   Use TRACS To upload your program

8. You must **also** turn in hard copy of your source code no later than the starting of class time on the due date . should the hard copy consist of more than one page , then , the hard copy must be **stapled**. if you are unable to turn in a printout during class, you can take the program to the computer science department and hand it to the front desk personal (Comal 211) before the deadline. Make sure that the front office stamps the program. Make sure that include the date and time. Finally ,make sure that they place the program in my mailbox. **Only one copy per group.**

   DO NOT slide your program under my office door – It will **NOT** be accepted

The following points will be deducted if :

- Incorrect file format such as uploading .cbp instead of .cpp , missing electronic copy , missing the hardcopy using .h and .cpp files , compilation errors , Using global variables / global arrays / global vector arrays, using linked lists and pointers , not using recursion … etc **(- 10 points)**

- Other **(at least 1.25 point each)** :

    - Logical Errors
    - Unable to read the source code due to unclear printing
    - Incorrect program file name.
    - Not using at least 7 functions + main .
    - Incorrect Style **such as but not limited to** missing comments or program documentations , missing serial number , missing section number, missing function prototypes , missing signature line , incorrect format … etc