Programming Assignment 5
Recursion

Due Date : Section 0 - Wednesday April 9th , 2018 - No Later than 2:15 pm

Section 1 - Wednesday April 9h , 2018 - No Later than 3:45 pm.

Section 2 - Wednesday April 9th , 2018 - No Later than 5:15 pm.

Write a C++ program that does the following :

1. **Accepts array size from the keyboard. Size must be a positive integer that is >= 10 and <= 1000.**
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 1000 inclusive**
3. **Display the first 10 elements of the generated array.**
4. **Write a function that uses recursion in order to display squares of integers in ascending order, starting from 1 to last number in the array.**
5. **Write a function that uses recursion to raise a number to a power. The function should take two arguments, the number to be raised to the power is the first number in array and the power is 2.**
6. **Write a recursive function that returns the maximum elements in the array**
7. **Using quick sort , write a function that uses recursion to sort the array in descending order. Calculate and print the CPU time before each step starts and after each completed step then calculate actual CPU time for the completion of each step. Use the last value as a pivot value.**
8. **Write a recursive function that that takes an integer (first number in the sorted arrays) as a parameter and returns the sum of digits of that integer.**

9. **Write a recursive function that determines whether or not every number in the sorted array is a prime number.**

The program displays a menu on the screen allowing the user to enter the a choice to enter the size of an array or to terminate the program.

NOTES:

- Just **one** .cpp file with at least 6 individual recursive functions plus main for testing.
- Do not use global variable , arrays ... etc.
- Validation on the menu selection and the array size.
- **`Replace My name (Husain Gholoom) with your first and last name.`**

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 :

```
//      Roster Number / s :      xxxxxxxxx
//
//      Author / s : (Your name here!!)
//      Due Date :
//      Programming Assignment Number 5
//
//      Spring 2018 - 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).

I**ndentation :**

- 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 :

1. Your program **must compile** and run. The program will be tested using the **latest** version of Codeblocks for windows.
2. Your program must be properly **documented according the style above . See the website for the sample programming style program.**
3. Must properly format the output by use the appropriate library. See the output below . Also , **Replace my first / last name with your own first / last name.**
4. You must name your program as : You must name your program as :

   - **PG5_3358_0_LastName_FirstName.cpp (Section 0)**
   - **PG5_3358_1_LastName_FirstName.cpp (Section 1)**
   - **PG5_3358_2_LastName_FirstName.cpp (Section 2)**

   Where LastName is your Last Name and FirstName is your First Name. For example , the file name should look something like : PG5_3358_0_Gholoom_Husain.cpp (not .cbp)

5. Every one must upload the electronic version of the program no later than the starting of class time on the due date. **No late assignments will be accepted. DO NOT** send your assignment solution via email. **Group members must upload identical copy of the assignment.**

   Use TRACS To upload electronic version of your program

6. 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 :

- Compilation errors , Incorrect file format such as uploading .cbp instead of .cpp , missing electronic copy , missing the hardcopy , different copies of the assignment per group , not using 6 or more different recursion functions , not using the indicated sort algorithm, more than one file such as using .h , .cpp file **(- 10 points)**

- Each logical error : **(- 1.25 points)**

- Other : **(1.5 points)** if **any** of the following takes a place :

    - Unable to read the source code due to unclear printing
    - Incorrect Output format.
    - Incorrect program file name.
    - Hard copy is not stapled.
    - Incorrect Style **such as but not limited to** Missing Header , footer , comments or program documentations , missing roster number , missing section number … etc

Sample Output

```
Thinking Recursively

The function of this program is to
use recursion in order to perform the following operations  :-

     1.  Display squares of integers in ascending  order from 1 to the
         last element in the array
     2.  Raise the first number to a power  2
     3.  Find the array's max value.
     4.  Sort the array in descending order
     5.  Calculating sum of digits
     6.  Determine if a number is prime (processes all array values)
```

```
Select from the following menu
A.    Enter Array Size that is > 4.
X.    Terminate The Program.     B

***  Invalid Option  ***


Select from the following menu
A.    Enter Array Size that is > 4.
X.    Terminate The Program.     3

***  Invalid Option  ***



Select from the following menu
A.    Enter Array Size that is > 4.
X.    Terminate The Program.     A


Enter Array Size: P

***  Invalid Array Size Value  ***
Enter an integer not a char: 3

***  Invalid Array Size Value  ***
***  Enter Array Size That is >= 10 and <=  1000 : 9

The generated array is:

433  411  430  370  295  20  143  461  441  64

Table of square values 1 - 64

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
28    784
29    841
30    900
31    961
32    1024
33    1089
34    1156
35    1225
36    1296
37    1369
38    1444
39    1521
40    1600
41    1681
42    1764
43    1849
44    1936
45    2025
46    2116
47    2209
48    2304
49    2401
50    2500
51    2601
52    2704
53    2809
54    2916
55    3025
56    3136
57    3249
58    3364
59    3481
60    3600
61    3721
62    3844
63    3969
64    4096
```

```
Power Function:

433 raised to the 2nd power is: 187489


Max Number of ( 433 , 411 , 430 , 370 , 295 , 20 , 143 , 461 , 441 ,
64 ) is : 461


Sorted array

461  441  433  430  411  370  295  143  64  20

Start Time      :
End Time   :
Actual CPU Clock time  :

Sum of digits for  the number   461     is    11

Is it prime:

461 is Prime
441 is Not Prime
433 is Prime
430 is Not Prime
411 is Not Prime
370 is Not Prime
295 is Not Prime
143 is Not Prime
64 is Not Prime
20 is Not Prime


Select from the following menu
A.    Enter Array Size that is > 4.
X.    Terminate The Program.     a

Enter Array Size: 10

The generated array is:

120  432  120  290  260  307  56  295  488  374


Table of square values 1 - 374

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
28    784
29    841
30    900
31    961
32    1024
33    1089
34    1156
35    1225
36    1296
37    1369
38    1444
39    1521
40    1600
41    1681
42    1764
43    1849
44    1936
45    2025
46    2116
47    2209
48    2304
49    2401
50    2500
51    2601
52    2704
53    2809
54    2916
55    3025
```

```
56    3136
57    3249
58    3364
59    3481
60    3600
61    3721
62    3844
63    3969
64    4096
65    4225
66    4356
67    4489
68    4624
69    4761
70    4900
71    5041
72    5184
73    5329
74    5476
75    5625
76    5776
77    5929
78    6084
79    6241
80    6400
81    6561
82    6724
83    6889
84    7056
85    7225
86    7396
87    7569
88    7744
89    7921
90    8100
91    8281
92    8464
93    8649
94    8836
95    9025
96    9216
97    9409
98    9604
99    9801
100   10000
101   10201
102   10404
103   10609
104   10816
105   11025
106   11236
107   11449
```

```
108   11664
109   11881
110   12100
111   12321
112   12544
113   12769
114   12996
115   13225
116   13456
117   13689
118   13924
119   14161
120   14400
121   14641
122   14884
123   15129
124   15376
125   15625
126   15876
127   16129
128   16384
129   16641
130   16900
131   17161
132   17424
133   17689
134   17956
135   18225
136   18496
137   18769
138   19044
139   19321
140   19600
141   19881
142   20164
143   20449
144   20736
145   21025
146   21316
147   21609
148   21904
149   22201
150   22500
151   22801
152   23104
153   23409
154   23716
155   24025
156   24336
157   24649
158   24964
159   25281
```

```
160   25600
161   25921
162   26244
163   26569
164   26896
165   27225
166   27556
167   27889
168   28224
169   28561
170   28900
171   29241
172   29584
173   29929
174   30276
175   30625
176   30976
177   31329
178   31684
179   32041
180   32400
181   32761
182   33124
183   33489
184   33856
185   34225
186   34596
187   34969
188   35344
189   35721
190   36100
191   36481
192   36864
193   37249
194   37636
195   38025
196   38416
197   38809
198   39204
199   39601
200   40000
201   40401
202   40804
203   41209
204   41616
205   42025
206   42436
207   42849
208   43264
209   43681
210   44100
211   44521
```

```
212   44944
213   45369
214   45796
215   46225
216   46656
217   47089
218   47524
219   47961
220   48400
221   48841
222   49284
223   49729
224   50176
225   50625
226   51076
227   51529
228   51984
229   52441
230   52900
231   53361
232   53824
233   54289
234   54756
235   55225
236   55696
237   56169
238   56644
239   57121
240   57600
241   58081
242   58564
243   59049
244   59536
245   60025
246   60516
247   61009
248   61504
249   62001
250   62500
251   63001
252   63504
253   64009
254   64516
255   65025
256   65536
257   66049
258   66564
259   67081
260   67600
261   68121
262   68644
263   69169
```

```
264   69696
265   70225
266   70756
267   71289
268   71824
269   72361
270   72900
271   73441
272   73984
273   74529
274   75076
275   75625
276   76176
277   76729
278   77284
279   77841
280   78400
281   78961
282   79524
283   80089
284   80656
285   81225
286   81796
287   82369
288   82944
289   83521
290   84100
291   84681
292   85264
293   85849
294   86436
295   87025
296   87616
297   88209
298   88804
299   89401
300   90000
301   90601
302   91204
303   91809
304   92416
305   93025
306   93636
307   94249
308   94864
309   95481
310   96100
311   96721
312   97344
313   97969
314   98596
315   99225
```

```
316   99856
317   100489
318   101124
319   101761
320   102400
321   103041
322   103684
323   104329
324   104976
325   105625
326   106276
327   106929
328   107584
329   108241
330   108900
331   109561
332   110224
333   110889
334   111556
335   112225
336   112896
337   113569
338   114244
339   114921
340   115600
341   116281
342   116964
343   117649
344   118336
345   119025
346   119716
347   120409
348   121104
349   121801
350   122500
351   123201
352   123904
353   124609
354   125316
355   126025
356   126736
357   127449
358   128164
359   128881
360   129600
361   130321
362   131044
363   131769
364   132496
365   133225
366   133956
367   134689
```

```
368   135424
369   136161
370   136900
371   137641
372   138384
373   139129
374   139876


Power Function:
120 raised to the 2nd power is: 14400


Max Number of ( 120 , 432 , 120 , 290 , 260 , 307 , 56 , 295 , 488 ,
374 ) is : 488


Sorted array

488  432  374  307  295  290  260  120  120  56

Start Time      :
End Time   :
Actual CPU Clock time  :


Sum of digits for  the number   488     is    20


Is it prime:

488 is Not Prime
432 is Not Prime
374 is Not Prime
307 is Prime
295 is Not Prime
290 is Not Prime
260 is Not Prime
120 is Not Prime
120 is Not Prime
56 is Not Prime


Select from the following menu
A.    Enter Array Size that is > 4.
X.    Terminate The Program.     x


Husain Gholoom - Tweak Programming Developer
April 2018
```