Programming Assignment - 2

Due Date : Monday September 25th , 2019 - No Later than 1:00 pm

---

A perfect matrix is an arrangement of **distinct** numbers (i.e. each number is used once), usually integers, in a square grid. The **property** of the perfect matrix is that the numbers in each row, and in each column, and the numbers in the main and secondary diagonals, all add up to the same perfect number. A **perfect matrix** has the same number of rows as it has columns, Thus, a **perfect matrix** always contains n^2 numbers.

Example of Perfect Matrix (perfect Number is 15)

6	**7**	**2**
1	**5**	**9**
8	**3**	**4**

Example of None Perfect Matrix (perfect number is 16)

1	**3**	**5**
2	**6**	**9**
4	**7**	**12**

Write a C++ program that :

1. Allow the user to enter the size of the matrix such as N. N must be an **integer** that is greater than or equal to 2.
2. Create an vector of size N x N.
3. Call a function to do the following : Populate the vector with N^2 **distinct** random numbers. Display the created array.
4. Call a function to determines whether the numbers in n x n vector satisfy the perfect matrix property. Look at the sample run for the exact output

5. Repeat steps 1 - 4 until the user terminates the program.

Note : Use the following method to calculate the perfect matrix property number :

Add all the numbers in the vector array , then divide the sum by 3.

Note : The program must be running until the user decides to terminated it by entering n or N

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 :

```
//      Author / s : (Your names and serial numbers   here!!) -
//      Serila Number / s :
//
//      Due Date  :
//      Programming Assignment Number  2
//      Fall 2019   -   CS 3358   -   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 :

1. **This program must be done as a group. Individual work will not be accepted and a grade of zero will be assigned.**
2. Your program **must compile** and run. The program will be tested using the latest version of Codeblocks for windows.
3. Your program must be properly **documented according the style above . See the website for the sample programming style program.**
4. Must **use function prototypes & definitions . You can also use repetitions , control structures , and switch statements. You re not allowed to use 1 dimensional arrays / vector arrays, or global variables / arrays. You are only allowed to use 2 dimensional vector arrays.**
5. You must use the appropriate libraries in writing this program.
6. 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.**
7. You must name your program as :
   - **3358_4_LastName_FirstName_PG2.cpp**

   Where LastName is your Last Name and FirstName is your First Name. For example , the file name should look something like :

   3358_4_ Gholoom_Husain_PG2.cpp (not .cbp)

8. You must upload your programs no later than the starting of class time on the due date. **No late assignments will be accepted.**

   Use TRACS To upload your program. Everybody must upload the electronic version of the assignment.

9. You must **also** turn in hard copy of your source code no later than the due date / time **. One hard copy / group** . 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.

   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 , assignment done as an individual work and not part of a group **(- 10 points)**
- Compilation Errors **(- 10 points)**
- Logical Errors **(at least - 1 point)**
- Other **(at least 1 point)** if **any** of the following takes a place :
  - Unable to read the source code due to unclear printing
  - **Missing function prototypes & definitions, incorrect number of functions , missing switch statements .. etc**
  - 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 group number , missing section number … etc

Sample Run

```
Welcome to my perfect matrix program. The function of the program is
to:

   1. Allow the user to enter the size of the perfect matrix ,such as
      N. N>=2.
   2. Create a 2 D vector array of size N x N.
   3. Populate the 2 D vector array with distinct random numbers.
   4. Display the perfect number , sum for each row, column, and
       diagonals then determine whether the numbers in N x N vector
       array are perfect matrix numbers.

Enter the size of the matrix : 2


The perfect matrix that is created for size 2 :


2        3


4        1



The perfect number is : 3


Sum of numbers in Row      #   1      =    5
Sum of numbers in Row      #   2      =    5

Sum of numbers in Column   #   1      =    6
Sum of numbers in Column   #   2      =    4

Sum of numbers in first diagonal      =    3
Sum of numbers in second diagonal     =    7

The above is not a perfect matrix
```

```
Would you like to find another perfect matrix  - Enter y or Y for yes
or n or N for no: y


Enter the size of the matrix : 3


The perfect matrix that is created for size 3 :


0        7        10


12       4        1


8        14       9



The perfect number is : 21


Sum of numbers in Row      #   1      =   17
Sum of numbers in Row      #   2      =   17
Sum of numbers in Row      #   3      =   31

Sum of numbers in Column   #   1      =   20
Sum of numbers in Column   #   2      =   25
Sum of numbers in Column   #   3      =   20

Sum of numbers in first diagonal      =   13
Sum of numbers in second diagonal     =   22


The above is not a perfect matrix
```

```
Would you like to find another perfect matrix  - Enter y or Y for yes
or n or N for no: x

Error *** Invalid choice - Must enter y|Y or n|N


Would you like to find another perfect matrix  - Enter y or Y for yes
or n or N for no: Y


Enter the size of the matrix : -3

Error *** Invalid choice – Size must be > 1


Would you like to find another perfect matrix  - Enter y or Y for yes
or n or N for no: n


This algorithm is implemented by Husain Gholoom
September  25 - 2019
```