NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS
CS-118 Programming Fundamentals - SPRING 2021
ASSIGNMENT- 4

Section (A, B, C, D, E and F)

Due Date: Wednesday 16th May 2021 at 11:59 pm on Google Classroom

Total Marks: 180.

Instructions:

1. Assignments are to be done individually. You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. The code you write must be your own and you must understand each part of your code. You are encouraged to get help from the instructional staff through email, or on google classroom.
2. The AIM of this assignment is to give you practice with Loops (especially in for loop) in C++ (**Chapter 5 of the textbook**).
3. Use appropriate data types, operations, and conditional structures, and loops for each problem. You cannot use **arrays or advanced topics (**for example **Chapter 6** of the textbook and **onward**) for this assignment.
4. No late assignments will be accepted.
5. The output should be properly displayed and well presented. Use appropriate comments and indentation in your source code.
6. Plagiarism: *Plagiarism of any kind (copying from others, copying from the internet etc.,) is **not** allowed. If found plagiarized, you will be awarded zero marks in the assignment. Repeating such an act can lead to strict disciplinary actions and failure in the course.*

Submission Guidelines:

Dear students, we will be using auto-grading tools, so failure to submit according to the below format would result in zero marks in the relevant evaluation instrument.

i) For each question in your assignment if necessary, make a separate .cpp file e.g. for question 1, make q1.cpp and so on. Each file that you submit must contain your name, student-id, and assignment # on the top of the file in the comments.

ii) Combine all your work in one folder. The folder must contain only .cpp files (no binaries, no exe files etc.,).

iii) Run and test your program on a lab machine before submission.

iv) Rename the folder as ROLL-NUM_SECTION (e.g. 19i-0001_B) and compress the folder as a zip file. (e.g. 19i-0001_B.zip).

v) Submit the .zip file on Google Classroom within the deadline.

vi) Submission other than Google classroom (e.g. email etc.) will not be accepted.

vii) The student is solely responsible to check the final zip files for issues like corrupt file, virus in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason it will lead to zero marks in the assignment.

Note: This assignment will be evaluated under two criterions (e.g., general and specific criteria). The marks of each question will be divided into general criteria (weightage: 70%) and specific criteria (weightage: 30%).

General criteria: applies to all questions (70% Marks):

Grading criterion	Description	Percentage
Execution / specification	Program executes correctly (in all cases) with no syntax or runtime errors and meets the specifications.	30%
Correct and well-designed output	Program displays correct output with no errors. Furthermore, the output needs to be well-designed.	20%
Design of logic / code efficiency	Program is logically well designed. Code uses the best and most efficient approach in every case.	10%
Readability	Proper indentation is there, and code is stylistically well-designed (i.e. code is clean, understandable, and well-organized).	5%
Documentation	Program is well-documented.	5%

Question #1: (Total marks: 30)

Write a C++ program for the conversion of a decimal number to a binary, octal and hexadecimal equivalents within a given range of the upper and lower limits. Your program needs to follow the following steps to give the desired output.

(1.a). Your program should ask to choose the upper limit from 256, 128, 32, 16, 8, 4, 2 only. If a user enters a number other than these numbers, the program should display a message of wrong entry and should ask for retry, otherwise continue to step (b).

(1.b). Your program should ask to enter the lower limit. The lower limit should be less than the upper limit (of step 1.a) and greater than 0. If a user enters a number greater than the upper limit or less than 0 (e.g., a negative value), then the program should display a message of wrong entry and should ask for retry, otherwise continue to display the output.

(1.c). If a user enters the upper and lower limits correctly, then the program should print a table of the binary, octal and hexadecimal equivalents of the given decimal numbers in the range of upper and lower limits as given in the example output:

Specific grading criteria (30% Marks): In this program you should use a for loop at least one time. You can use while and do-while loops in the program if further required. The output of your program should be as given in the example output. However, this is just an example output, whereas more clear and well-presented output can improve your marks.
Help and hints: http://www.cplusplus.com/reference/ios/dec/
https://ncalculators.com/number-conversion/binary-to-decimal-hexa-octal-converter.htm

Example output:

```
Please enter one of a number from this list (256, 128, 64, 32, 16, 8, 4 or 2) as upper limit: 45
You should enter the upper limit from this list only: 256, 128, 64, 32, 16, 8, 4, 2
Please enter the upper limit: 32

Enter lower limit:  -12
You should enter the value greater than 0 and less than the upper limit
Enter lower limit:  50

You should enter the value greater than 0 and less than the upper limit
Enter lower limit:  10

Decimal     Binary      Octal      Hexadecimal
.............................................................................
10          001010       12           a
11          001011       13           b
12          001100       14           c
13          001101       15           d
14          001110       16           e
15          001111       17           f
16          010000       20           10
17          010001       21           11
18          010010       22           12
19          010011       23           13
20          010100       24           14
21          010101       25           15
22          010110       26           16
23          010111       27           17
```

24	011000	30	18
25	011001	31	19
26	011010	32	1a
27	011011	33	1b
28	011100	34	1c
29	011101	35	1d
30	011110	36	1e
31	011111	37	1f
32	100000	40	20

Question #2: (20 Marks)

(a) Write a C++ program to print all prime numbers within a given range (e.g., 1 to 100) by using a for loop and print all the possible prime numbers in the given range. Then print the total number of primes. Third sum all the primes numbers within that given range.

Specific grading criteria: Use only for loop and print the output as mentioned below. This is just an example; more clear and well-presented output can improve your marks. (30% Marks).

Example output:

```
Find prime number within a range:
--------------------------------------
 Input number for starting range: 10
 Input number for ending range: 100

 The prime numbers between 10 and 100 are:
11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

 The total number of prime numbers between 10 to 100 is: 21
The sum of all the prime numbers is = 1043
```

Question #3: (Total Marks: 30)
Pascal's Triangle: The example output of this program shows a pascal's triangle. To build the pascals triangle, start with 1 at the top, then continue placing numbers below it in a triangular pattern. Each number is the numbers directly above is added together.
Explore this link for hints of pascal's triangle: https://www.mathsisfun.com/pascals-triangle.html

3.a): Write a C++ program to print the pascal's triangle by using "for loop". The program should take the number of rows as input from the user. The output of the program should be as in the following pattern (e.g., if you enter the number of rows = 8).

Note: specific grading criteria: You need to use only for loop and print a well-designed pattern. The output mentioned below is just an example; more clear and well-presented output will improve your marks. (10 marks).
Example Output:

```
Enter the number of rows: 8
                                   1

                                 1   1

                               1   2   1

                             1   3   3   1

                           1   4   6   4   1

                        1   5   10   10   5   1

                      1   6   15   20   15   6   1

                    1   7   21   35   35   21   7   1
```

(3.b): Use a "for loops" to print the invert of the above pascal triangle. The program should take the number of rows as input from the user. The output of the program should be as in the following pattern e.g., if you enter the number of rows = 8.

Specific grading criteria: Use only for loop and print a well-designed pattern. The output mentioned below is just an example; more clear and well-presented output will improve your marks. (10 Marks).

Example Output:

```
Enter the number of rows: 8
                    1   8   28   56   70   56   28   8   1

                      1   7   21   35   35   21   7   1

                        1   6   15   20   15   6   1

                          1   5   10   10   5   1

                             1   4   6   4   1

                               1   3   3   1

                                 1   2   1

                                   1   1

                                     1
```

(3.c) Use for loops to combine both the output of the part (a) and part (b) as mentioned in the example output below. The program should take the number of rows as input from the user. The output of the program should be as in the following pattern e.g., if you enter the number of rows = 8.

Specific grading criteria: you need to use only for loop and print a well-designed pattern. The output mentioned below is just an example; more clear and well-presented output will improve your marks. (5 Marks).

Example Output:
Enter the number of rows: 8

```
                    1

                  1   1

                1   2   1

              1   3   3   1

            1   4   6   4   1

          1   5   10   10   5   1

        1   6   15   20   15   6   1

      1   7   21   35   35   21   7   1

    1   8   28   56   70   56   28   8   1

      1   7   21   35   35   21   7   1

        1   6   15   20   15   6   1

          1   5   10   10   5   1

            1   4   6   4   1

              1   3   3   1

                1   2   1

                  1   1

                    1
```

(3.d) Use for loop and combine both the output of the part (a) and part (b) as mentioned in the example output below. The program should take the number of rows as input from the user. The output of the program should be as in the following pattern e.g., if you enter the number of rows = 8.

Specific grading criteria: Well-designed pattern and using only for loop. The output mentioned below is just an example; more clear and well-presented output will improve your marks. (5 Marks).

Example Output:

Enter the number of rows: 8

```
           1   8   28   56   70   56   28   8   1
             1   7   21   35   35   21   7   1
                1   6   15   20   15   6   1
                   1   5   10   10   5   1
                      1   4   6   4   1
                        1   3   3   1
                          1   2   1
                            1   1
                              1
                            1   1
                          1   2   1
                        1   3   3   1
                      1   4   6   4   1
                   1   5   10   10   5   1
                1   6   15   20   15   6   1
             1   7   21   35   35   21   7   1
           1   8   28   56   70   56   28   8   1
```

Question #4: (20 Marks)

Write a C++ program that counts the number of even digits (including the number of zeros) and odds from a given list of digits. Your program should ask first for the number of digits of a list. Then ask to enter each digit according to the given list. The program should display the total number of even numbers, zeros, and odd numbers. This program should be general-purpose and print the desired input for any given list. The desired output of the program is the following.

Note: Specific grading criteria: Well-designed output and only for-loop should be used to perform the iterations. The output mentioned below is just an example; more clear and well-presented output will improve your Marks. (30 % marks)

Example output:

```
Please mention the total number of integers to enter (from the list of positive, negative, or zeros) = 10
Please now enter the list of all 10 numbers:
7
0
-4
1
6
-6
0
0
5
-7

There are total 6 evens which includes 3 zeros.
The total odd numbers you have entered is: 4
```

Question #5: (30 Marks)

Write a C++ program using for loop to print all the different arrangements of the letters A, B, C, D and E. Each string printed is a permutation of ABCDE (it means all strings will be different from each other).

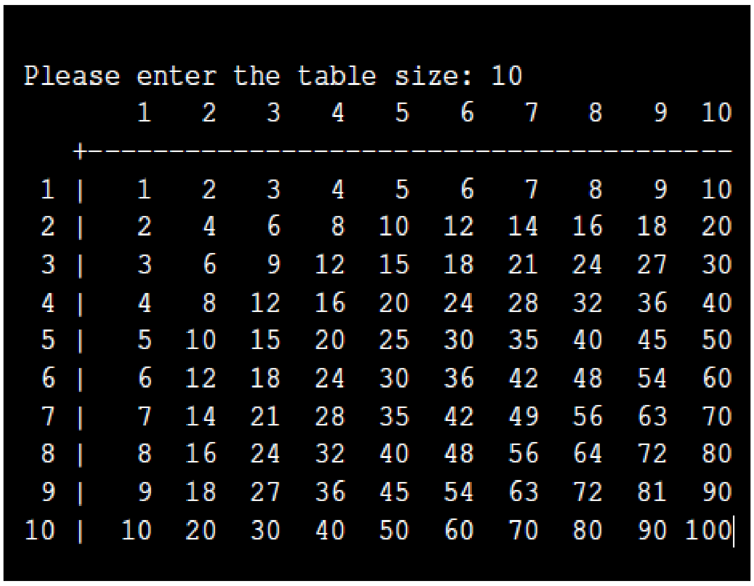
Example output:

```
ABCDE                    BDEAC                    DBAEC
ABCED                    BDECA                    DBCAE
ABDCE                    BEACD                    DBCEA
ABDEC                    BEADC                    DBEAC
ABECD                    BECAD                    DBECA
ABEDC                    BECDA                    DCABE
ACBDE                    BEDAC                    DCAEB
ACBED                    BEDCA                    DCBAE
ACDBE                    CABDE                    DCBEA
ACDEB                    CABED                    DCEAB
ACEBD                    CADBE                    DCEBA
ACEDB                    CADEB                    DEABC
ADBCE                    CAEBD                    DEACB
ADBEC                    CAEDB                    DEBAC
ADCBE                    CBADE                    DEBCA
ADCEB                    CBAED                    DECAB
ADEBC                    CBDAE                    DECBA
ADECB                    CBDEA                    EABCD
AEBCD                    CBEAD                    EABDC
AEBDC                    CBEDA                    EACBD
AECBD                    CDABE                    EACDB
AECDB                    CDAEB                    EADBC
AEDBC                    CDBAE                    EADCB
AEDCB                    CDBEA                    EBACD
BACDE                    CDEAB                    EBADC
BACED                    CDEBA                    EBCAD
BADCE                    CEABD                    EBCDA
BADEC                    CEADB                    EBDAC
BAECD                    CEBAD                    EBDCA
BAEDC                    CEBDA                    ECABD
BCADE                    CEDAB                    ECADB
BCAED                    CEDBA                    ECBAD
BCDAE                    DABCE                    ECBDA
BCDEA                    DABEC                    ECDAB
BCEAD                    DACBE                    ECDBA
BCEDA                    DACEB                    EDABC
BDACE                    DAEBC                    EDACB
BDAEC                    DAECB                    EDBAC
BDCAE                    DBACE                    EDBCA
BDCEA                    EDCBA                    EDCAB

The total permutations of ABCDE are = 120
```

Question#6 (20 Marks): Write a C++ program by using a for loop to print a tabular multiplication as given below. The program should ask the user for the size of the table and then perform the table multiplication accordingly. You can use the **setw** manipulator to adjust the proper setting of the numbers in the table.

Example output:



```
Please enter the table size: 10
          1    2    3    4    5    6    7    8    9   10
    +-----------------------------------------------------
  1 |     1    2    3    4    5    6    7    8    9   10
  2 |     2    4    6    8   10   12   14   16   18   20
  3 |     3    6    9   12   15   18   21   24   27   30
  4 |     4    8   12   16   20   24   28   32   36   40
  5 |     5   10   15   20   25   30   35   40   45   50
  6 |     6   12   18   24   30   36   42   48   54   60
  7 |     7   14   21   28   35   42   49   56   63   70
  8 |     8   16   24   32   40   48   56   64   72   80
  9 |     9   18   27   36   45   54   63   72   81   90
 10 |    10   20   30   40   50   60   70   80   90  100
```

Question #7 (30 Marks): Write a C++ program to find all Pythagorean triples for two sides (e.g., side-a and side-b) and hypotenuse by using a for loop that tries all the possibilities for a given range. Print the integer values of the sides and hypotenuse as mentioned in the desired output below and show the total number of Pythagorean triples in the given range. A Pythagorean Triples is a right-angle triangle can have sides that are all positive integers **a**, **b** and **c** that fit the rule e.g., $\mathbf{a^2 + b^2 = c^2}$. More specifically the set of three integer values for the sides of a right triangle is called a Pythagorean triple. These three sides must satisfy the relationship that the sum of the squares of two of the sides is equal to the square of the hypotenuse.
Source: https://www.mathsisfun.com/pythagorean_triples.html
Note: Specific grading criteria: Use only for loop and print a well-designed pattern. The output mentioned below is just an example; more clear and well-presented output will improve your marks. (30 % Marks)

Example output:

```
Enter a range for all sides: 10
Side-a     side-b     hypotenuse
  .............................................
   3          4           5
   4          3           5
  6          8           10
  8          6           10

A total of 4 Pythagorean triples were found in range 10.
```