CMPT 130: Week 5 Lab Work

1. Do all the questions in Week 4 lab work using a **for** loop.
2. Do all the questions in Week 4 lab work using a **do-while** loop.
3. Analyze the outputs of the following three programs and determine their outputs.

```
int main()
{
    for (int i = 0; i < 5; i++)
    {
        if (i % 2 == 0)
            cout << i << endl;
        else
            continue;
    }
    system("Pause");
    return 0;
}
```

```
int main()
{
    int i = 0;
    while (i < 5)
    {
        if (i % 2 == 0)
            cout << i << endl;
        else
            continue;
        i++;
    }
    system("Pause");
    return 0;
}
```

```
int main()
{
    int i = 0;
    do
    {
        if (i % 2 == 0)
            cout << i << endl;
        else
            continue;
        i++;
    }while (i < 5);
    system("Pause");
    return 0;
}
```

4. Write a complete C++ program that reads two integers ***a*** and ***b*** and then prints all the integers between a and b (or vice versa) **exclusive;** that is without including ***a*** and ***b*** using a do-while loop. Remember if either the value of ***a*** is equal to ***b*** or that ***a*** and ***b*** are consecutive integers then your program should not print any integer because under such circumstances there is no any integer between ***a*** and ***b***. Under such circumstance your program should instead print the message "There is no any integer".
5. What is the output of the following while loops? Assume these codes are embedded in valid C++ project.

```
int n = 0;
while (n++ < 10)
    cout << n++ << endl;
```

```
int n = 0;
while (++n < 10)
    cout << n++ << endl;
```

```
int n = 0;
while (n+1 < 10)
    cout << n++ << endl;
```

6. Consider the following do-while loop. Assume these codes are embedded in valid C++ project.

```
int n;
do
{
    cout << "Please enter n: ";
    cin >> n;
} while (n<0);
```

Convert this do-while loop to an equivalent

a. for loop.
b. while loop.

7. Write a complete C++ program that reads a positive integer **n**, asserts the value of **n** is a positive integer, and finally prints the first **n** prime numbers greater or equal to **n**. For example if the user input value for **n** is 9, then you must print the prime numbers 11, 13, 17, 19, 23, 29, 31, 37, and 41. As another example, if the user input value for **n** is 2, then you must print the prime numbers 2 and 3.

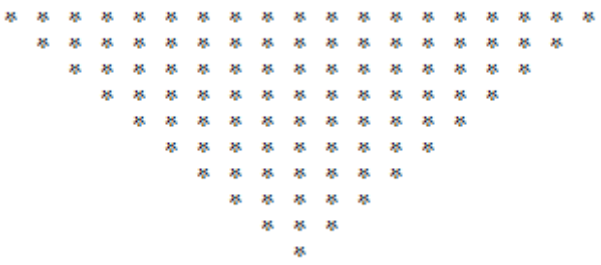
8. Write a C++ program that reads a user input integer value **n** and that prints the following pattern with **n** rows. You program must have only ONE **cout << "*";** statement. Therefore you must use a nested loop. You can choose whichever loop structure (for, while, do-while or combination of them) to use. The following diagram shows the answer for n = 10.

Remark:- When you print patterns made with asterisks, do **NOT** use space or tab in between the asterisks in your output. For example, the following nested loop prints a triangular pattern of asterisks as shown above with no space in between the asterisks.

```
for (int i = 1; i < n; i++)
{
	for (int j = 1; j <= n; j++)
	{
		if (j <= n - i)
			cout << " ";   //One space
		else
			cout << "*";   //No space before or after the asterisk character
	}
	cout << endl;
}
```

9. Write a C++ program that reads a user input integer value **n** and that prints the following pattern with **n** rows. The pattern shown below is for n = 10.

10. Write a C++ program that reads a user input integer value **n** and that prints the following pattern with **n** rows. You program must have only ONE **cout << "*";** statement. Therefore you must use a loop. You can choose whichever loop structure (for, while, do-while or combination of them) to use. The following diagram shows the answer for n = 10.



11. Consider the multiplication table.

1	**2**	**3**	**...**	**n**
2	**4**	**6**	**...**	**2*n**
3	**6**	**9**	**...**	**3*n**
⋮				
n	**2*n**	**3*n**	**...**	**n*n**

Write a C++ program that reads an integer **n** and prints such a multiplication table.

a. Using a for loop inside for loop
b. Using a while loop inside for loop
c. Using a do-while loop inside for loop
d. Using a for loop inside while loop
e. Using a while loop inside while loop
f. Using a do-while loop inside while loop
g. Using a for loop inside do-while loop
h. Using a while loop inside do-while loop
i. Using a do-while loop inside do-while loop

12. Write a C++ program that reads an integer **n** and prints a half multiplication table shown below.

				1
			2	**4**
		3	**6**	**9**
				⋮
n	**2*n**	**3*n**	**...**	**n*n**

13. Write a C++ program that reads a positive integer **n** and then prints the following pattern which has got **2n-1** rows of stars. The pattern shown is for **n = 6**.

```
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * *
* * * *
* * *
* *
*
```

14. Write a C++ program that reads a positive integer **n** and then prints the following pattern which has got **n** rows of numbers. The pattern shown is for **n = 6**.

```
          1
        1 2 1
      1 2 3 2 1
    1 2 3 4 3 2 1
  1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1
```

15. Write a C++ program that reads a positive integer **n** and then prints the following pattern which has got **2n-1** rows of stars. The pattern shown is for **n = 6**.

```
          *
        * *
      * * *
    * * * *
  * * * * *
* * * * * *
  * * * * *
    * * * *
      * * *
        * *
          *
```

16. Write a C++ program that reads a positive integer **n** and then prints the following pattern which has got **2n-1** rows of numbers. The pattern shown is for **n = 6**.

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

17. Write a C++ program that reads a positive integer **n** and then prints the following pattern which has got **2n-1** rows of numbers. The pattern shown is for **n = 6**.

```
     1
    1 2
   1 2 3
  1 2 3 4
 1 2 3 4 5
1 2 3 4 5 6
 1 2 3 4 5
  1 2 3 4
   1 2 3
    1 2
     1
```

18. Write a C++ program that reads a positive integer **n** and then prints the following pattern which has got **2n-1** rows of stars. The pattern shown is for **n = 6**.

```
     *
    * * *
   * * * * *
  * * * * * * *
 * * * * * * * * *
* * * * * * * * * * *
 * * * * * * * * *
  * * * * * * *
   * * * * *
    * * *
     *
```

19. Write a C++ program that reads a positive integer **n** and then prints the following pattern which has got **2n-1** rows of numbers. The pattern shown is for **n = 6**

```
          1
        1 2 1
      1 2 3 2 1
    1 2 3 4 3 2 1
  1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1
  1 2 3 4 5 4 3 2 1
    1 2 3 4 3 2 1
      1 2 3 2 1
        1 2 1
          1
```

20. Write a C++ program that reads a positive integer **n** and then prints the following pattern which has got **2n-1** rows of stars. The following pattern shows the required output for the case **n = 6**.

```
* * * * * * * * * * *
  * * * * * * * * * *
    * * * * * * * *
      * * * * * *
        * * * * *
          * * *
            *
          * * *
        * * * * *
      * * * * * * *
    * * * * * * * * *
* * * * * * * * * * *
```

21. Write a C++ program that reads a positive integer **n** and then prints the following pattern which has got **2n-1** rows of numbers. The following pattern shows the required output for the case **n = 6**.

```
1  2  3  4  5  6  5  4  3  2  1
   1  2  3  4  5  4  3  2  1
      1  2  3  4  3  2  1
         1  2  3  2  1
            1  2  1
               1
            1  2  1
         1  2  3  2  1
      1  2  3  4  3  2  1
   1  2  3  4  5  4  3  2  1
1  2  3  4  5  6  5  4  3  2  1
```

22. Write a C++ program that reads a positive integer **n** and then prints the following pattern which has got **2n-1** rows of stars. The following pattern shows the required output for the case **n = 6**.

```
*                       *
*  *                 *  *
*  *  *           *  *  *
*  *  *  *     *  *  *  *
*  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *
*  *  *  *  *     *  *  *
*  *  *  *           *  *
*  *  *                 *
*  *                    *
*                       *
```

23. Write a C++ program that reads a positive integer **n** and then prints the following pattern which has got **2n-1** rows of stars. The following pattern shows the required output for the case **n = 6**.

```
1                       1
1 2                   2 1
1 2 3               3 2 1
1 2 3 4           4 3 2 1
1 2 3 4 5       5 4 3 2 1
1 2 3 4 5 6   5 4 3 2 1
1 2 3 4 5       5 4 3 2 1
1 2 3 4           4 3 2 1
1 2 3               3 2 1
1 2                   2 1
1                       1
```