NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS
OBJECT ORIENTED PROGRAMMING (CS103) - SPRING 2019
ASSIGNMENT-2

Due Date: March 4, 2019 (10:00 am)
Instructions:

1. *Make sure that you read and understand each and every instruction. If you have any questions or comments you are encouraged to discuss with your colleagues and instructors on piazza.*
2. *Plagiarism is strongly forbidden and will be very strongly punished. If we find that you have copied from someone else or someone else has copied from you (with or without your knowledge) both of you will be punished. You will be awarded straight zero in this assignment or all assignments.*
3. *Submit a single '.zip' file for your assignment and each problem solution must be provided in a separate CPP file. For instance, you must name the file containing solution of first file as 'q1.cpp' and second as 'q2.cpp' and so on.*
4. *Note: You have to follow the submission instructions to the letter. Failing to do so can get a zero in assignment.*

I) Write a recursive function find that finds given target value in the array. If value is not found your function must return -1, otherwise it should return the index of array where the value was found. Your function prototype must be as follows:
int find(int array[], int length, int target);

II) Given a number n, write a recursive function whether it's prime number or not. Your function should return true or false. Your function prototype must be as follows:
bool isPrime(int n, int index);

III) Given a string, write a recursive function to find its first uppercase letter. Your function should return the letter. Your function prototype must be as follows:
char findUpperCase(char * str, int index)

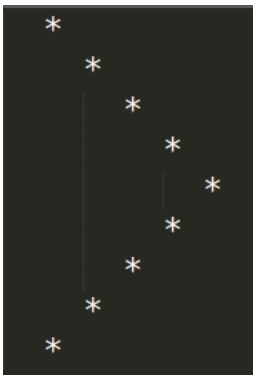
IV) The value of π can be determined by the series equation

$$\pi = 4 * (1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + 1/13 + \ldots)$$

Write a recursive function that takes an odd number n and uses recursion to approximate the value of π using the given formula including term up through nth term. Your function prototype must be as follows:
float PiValue(int n); `//Calling this function "PiValue(1875)" should give the answer 3.14052655581`

V) Write a recursive function that takes two matrix of same size and output whether they are equal or not. If matrices are equal, return 1 else 0. Your function prototype must be as follows:
int equal(int ** matrix1, int** matrix2, int row, int column)

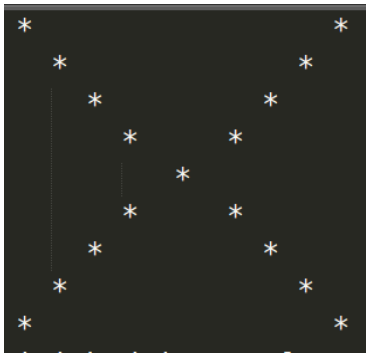
VI) Write a program that takes a 2D pointer array and calculate sum of even and odd using recursive function. Your function prototype must be as follows:
void sum(int **array, int row, int column, int &evenSum, int &oddSum)



VII) Write a C++ recursive function PrintPattern1 to print following pattern using recursion. No loops allowed whatsoever and you can write maximum two functions apart from main function. For example calling your function with these argument PrintPattern1(1,10) should print following pattern. Your function prototype must be as follows:

```
void PrintPattern1(int start, int end, ofstream &output);
```

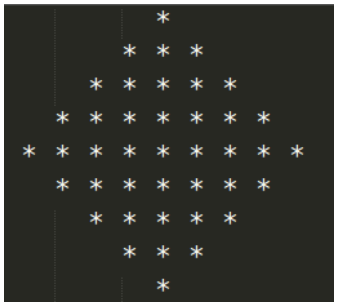
Instead of printing the pattern, it should be written in the file with the help of ofstream object titled as output being sent as function argument



VIII) Write a C++ recursive function PrintPattern2 to print following pattern using recursion. No loops allowed whatsoever and you can write maximum two functions apart from main function. For example calling your function with these argument PrintPattern2(5,1,5) should print following pattern. Your function prototype must be as follows:

```
void PrintPattern2(int, int ,int, ofstream &output);
```

Instead of printing the pattern, it should be written in the file with the help of ofstream object titled as output being sent as function argument



IX) Write a C++ recursive function PrintPattern3 to print following pattern using recursion. No loops allowed whatsoever and you can write maximum two functions apart from main function. For example calling your function with these argument PrintPattern3(1,5) should print following pattern.Your function prototype must be as follows:

```
void PrintPattern3(int, int, ofstream &output);
```

Instead of printing the pattern, it should be written in the file with the help of ofstream object titled as output being sent as function argument

X) Write a recursive method named **permutate** that accepts two integers n and r as parameters and returns the number of unique permutations of r items from a group of n items. For given values of n and r, this value **p(n,r)** can be computed as follows:

$$P(n,r) = \frac{n!}{(n-r)!}$$

For example **permutate(7,4)** should return 840. Your function prototype must be as follows: *long permutate (int n, int r)*

XI) Given an array of integers, print a sum triangle from it such that the first level has all array elements. From then, at each level number of elements is one less than the previous level and elements at the level is be the Sum of consecutive two elements in the previous level. You have to save the sum triangle in another array. Your function prototype must be as follows:
void printTriangle(int arr[]*/*initial array*/****, int size****/*size of array*/****, int **sumTriangle*** */*sum triangle*/****, int row****/*row size of sum triangle*/****, int column****/*column size of sum triangle*/****)***

```
Input : A = {1, 2, 3, 4, 5}
Output : [48]
         [20, 28]
         [8, 12, 16]
         [3, 5, 7, 9]
         [1, 2, 3, 4, 5]

Explanation :
Here,   [48]
        [20, 28] -->(20 + 28 = 48)
        [8, 12, 16] -->(8 + 12 = 20, 12 + 16 = 28)
        [3, 5, 7, 9] -->(3 + 5 = 8, 5 + 7 = 12, 7 + 9 = 16)
        [1, 2, 3, 4, 5] -->(1 + 2 = 3, 2 + 3 = 5, 3 + 4 = 7, 4 + 5 = 9)
```

XII) Given following three values, the task is to find the total number of maximum chocolates you can eat.

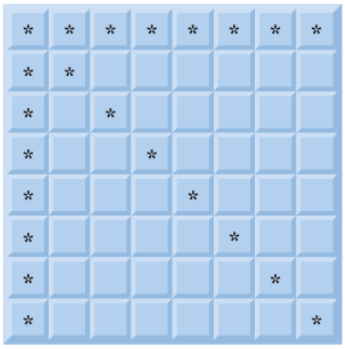
1. *money : Money you have to buy chocolates*
2. *price : Price of a chocolate*
3. *wrap : Number of wrappers to be returned for getting one extra chocolate.*

```
Input :   money = 16, price = 2, wrap = 2
Output :   15
Price of a chocolate is 2. You can buy 8 chocolates from
amount 16. You can return 8 wrappers back and get 4 more
chocolates. Then you can return 4 wrappers and get 2 more
chocolates. Finally you can return 2 wrappers to get 1
more chocolate.

Input :   money = 15, price = 1, wrap = 3
Output :   22
We buy and eat 15 chocolates
We return 15 wrappers and get 5 more chocolates.
We return 3 wrappers, get 1 chocolate and eat it
(keep 2 wrappers). Now we have 3 wrappers. Return
3 and get 1 more chocolate.
So total chocolates = 15 + 5 + 1 + 1

Input :  money = 20, price = 3, wrap = 5
Output :   7
```

It may be assumed that all given values are positive integers and greater than 1. Your function prototype must be as follows:

int countMaxChoco(int money, int price, int wrap)

XIII) Solve the **five Queens** problem recursively. The objective is to place 5 queens on an empty chessboard (5x5 matrix) so that no queen is "attacking" any other, i.e., no two queens are in the same row, the same column, or along the same diagonal?



Hint: See the Book Deitel & Deitel's How to Program Problem 7.26

Your function prototype must be as follows:

bool solveNQUtil(int **board, int N, int col)
//board is the 5 x 5 matrix that has to be uploaded, N = 5, while col is the current column that needs to be checked for placing the queen.