NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABAD

Object Oriented Programming (CS217)

SPRING 2021 ASSIGNMENT # 1

Due Date: Wednesday, April 14 (11:59 pm)

Instructions

Submission: Combine all your work in one .zip file. Use proper naming convention for your submission file. Name the .zip file as **SECTION_ROLL-NUM_01.zip** (**e.g. A_20i0412_01.zip**). Your zip file should not contain any folders or subfolders. It should only contain .cpp files for each question, e.g. Q1.cpp, Q2.cpp, …, Q8.cpp. Submit .zip file on Google Classroom within the deadline. Failure to submit according to the above format would result in **25% marks deduction**. Submissions on the email will not be accepted.

Plagiarism: Plagiarism cases will be dealt with strictly. If found plagiarized, both the involved parties will be awarded zero marks in this assignment, all the remaining assignments, or even an **F grade** in the course. Copying from the internet is the easiest way to get caught!

Deadline: The deadline to submit the assignment is **14th April 2021 at 11:59 PM**. Late submission with marks deduction will be accepted according to the course policy shared earlier. Correct and timely submission of the assignment is the responsibility of every student; hence no relaxation will be given to anyone.

Test cases: Test cases (in gtest) will be shared with you on Google Classroom.

Comments: Comment your code properly. Write your name and roll number (as a block comment) at the beginning of the solution to each problem.

Tip:

- *You must do proper allocation and deallocation of the memory where necessary.*
- All programs must be generic.
- For timely completion of the assignment, start as early as possible.
- Your code should be modular, function prototypes are given with each question.

Note: Follow the given instructions to the letter, failing to do so will result in a zero.

Question # 1 (10 marks)

Write a recursive function for the following problem. You need to change the location of the entries with indices between the two boundaries of the char array.

```
void ChangeLocation(char *Array, int b1, int b2);
```

Example:

Char array [8] == {'P', 'A', 'K', 'I', 'S', 'T', 'A', 'N'}

The bounds are 3 and 7 then after the function execution the value should be like

Char array [8] == {'P', 'A', 'A', 'I', 'S', 'T', 'K', 'N'}

Question # 2 (10 marks)

Write a recursive function to print the following pattern.

```
void PrintPattern(int &n);
```

Example:

Enter any number = 4

4

2 2

1 3

1 1 2

1 1 1 1

Question # 3 (10 marks)

Write a program to take two string inputs from user and call the recursive function to find whether the second string is a subsequent of the first.

```
void CheckSubsequent(char *str1, char *str2, int i, int j);
```

Example:

Enter String 1: pathetic

Enter String 2: the

Output: **True**
Example:
Enter String 1: pack
Enter String 2: cap

Output: **False**

Question # 4 (10 marks)

Write a program to take input in dynamic integer array and show the occurrence of each number in the array in descending order using recursive function.

```
void FindOccurances(int *A, int i);
```

Example:
Enter elements in the array: {2,4,1,5,6,4,2}

Output:

Number	Occurrences
2	2
4	2
1	1
5	1
6	1

Question # 5 (10 marks)

Write a program and take distinct input in the 2D-dynamic array at each index. An element **array[i][j]** of array is termed as an "inversion of Array" if i < j and A[i][j] > A[j][i]. Write a program to count the number of inversions in the array using recursive function.

```
void FindInversion(int **A, int i, int j);
```

Question # 6 (10 marks)

Write a program that takes a 2D pointer array and calculate sum of even and odd using pointer notation of the array.

Question # 7 (25 marks)

Write a program that takes a three 2D pointer array and find the following by implementing the function for each of the following task:

1. Is the sum of any two arrays is equal to the 3rd array?
   void CheckEqualSumArrays(int **A1, int **A2, int **A3);

2. Is the difference of any two arrays is equal to the 3rd array?
   void CheckDifferentArrays(int **A1, int **A2, int **A3);

3. Are there any equal arrays among these?
   void CheckEqualArrays(int **A1, int **A2, int **A3);

4. Find the same rows in each array.
   void FindSameRows(int **A1, int **A2, int **A3);

5. Rotate all three arrays up to 90 degrees in clockwise direction.
   void RotateArrays(int **A1, int **A2, int **A3);

Note: You can only use the pointer notation in manipulation of the array.

Question # 8 (10 marks)

Write a function that receives a string consisting of several lines of text and returns arrays indicating unique words in the text along with their size.

```
void countingUniqueWords (char * string, char *&uwords , int *size);

/*uwords - list of unique words size- size of
words */
```