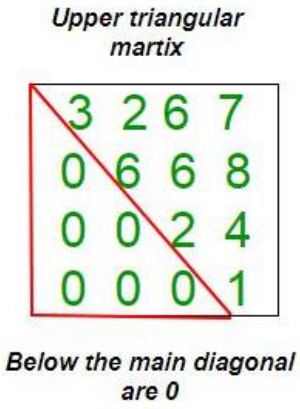
Assignment # 1

Dynamic 2-D Arrays

Submission Dead Line: Thursday 18/4/2021

— PROVIDE PROPER INDENTATION AND COMMENTS WITH YOUR CODE
— YOU MUST DEALLOCATE ALL MEMORY PROPERLY, YOUR CODE SHOULD NOT HAVE ANY **MEMORY LEAKS** OR **DANGLING POINTERS**.

Question 1: Write a C++ program to implement an upper triangular matrix. It is a matrix with non-zero entries on the diagonal and above the main diagonal. The entries below the main diagonal are zero. Since entries below the main diagonal are zero, we do not want to store them. Provide following operations on upper triangular matrices.



1. **Allocate Memory:** Write a function that allocates memory to an upper triangular matrix. It will have n entries in row 1, n-1 entries in row 2, n-3 entries in row 3 and so on. (n should be input from the user).

```
void AllocateTriangularMatrix(int ***arr, int n)
```

2. **Deallocate Memory:** Write a function to deallocate the matrix. All memory must be deallocated properly at end of program.

3. **Input:** write a function to input the contents of the matrix from a user.

4. **Read from File:** Read two upper triangular matrices from file. Sample is attached.

   File format:
   Order (first line specifies the order of the matrix).
   matrix 1 (values of matrix 1 in row major order)
   matrix 2 (values of matrix 2 in row major order)

Note: You have to check whether a matrix is upper triangular or not before storage, and display appropriate message to user.

5. **Print:** Design a function to print the upper triangular matrix row wise and column wise. Printing will include all data including zeros.

6. **Add:** Design a function for Addition of matrices. You cannot convert matrices to normal ones for multiplication. The matrix A has dimensions m x n, Matrix B has dimensions n x r. What will be the dimensions of C???

```
bool MultiplyMatrices(int **A, int **B, int ***C, int m, int n, int r)
```

7. **Multiply:** Find out how matrix multiplication is done you cannot convert matrices to normal ones for multiplication. Implement this function that multiplies two upper triangular matrices A and B and stores the result in C. Make sure that for C a null pointer is passed. If it is not null pointer, then you should return false. Otherwise allocate the matrix C in this function best to call Allocate function. On successful multiplication the function should return true. The matrix A has dimensions m x n, Matrix B has dimensions n x r. What will be the dimensions of C???

```
bool MultiplyMatrices(int **A, int **B, int ***C, int m, int n, int r)
```

8. **Swap:** Write a function to swap two rows of the upper triangular matrix. You need to think about memory allocation and deallocation carefully here.

9. **Delete Row:** Write a function to delete a row from the matrix. After this operation the matrix will no longer be a square matrix. So, make it a square upper triangular matrix by deleting the appropriate column.

10. **Menu:** For all the parts given above make a menu-based program to access and test each function.

Question 2: Write a program that performs following string manipulations: You cannot change prototype of any function.

1. Write a function that takes two strings i.e., str1 and str2 then appends str1 at the end of str2.

```
void StringConcatenate (char *str1, char *str2) ;
```

Note: Do not use any extra string inside the function.

2. Write a function that takes a string and if it finds more than one occurrences of a character in the string, it removes the extra occurrences.

```
Void CompressString(char*);
```

For example:
String: "abadca" String after compression: "abdc"
String: "aaaaaaabbaad" String after compression: "abd"
Note: Do not use any extra string inside the function.

3. Write a function that takes a sentence and returns its inverse.

```
Char* ReverseSentence(char*)
```

String: "I am Pakistani"
After calling Reverse Sentence
String: "Pakistani am I" (Do not change the original string)

4. Write a function that takes a 2d-array of singular words and convert each singular word of array to a ***plural*** word.

```
void pluralWords(char **s, int wordCount);
```

a. Append **'es'** in following cases
   1. If a word ends with letter **'h'** and second last letters are **'c'** or **'s'**
   2. If last letter is **'x' or 'is'**
      Tax – Taxes
      Analysis – Analyses
      Ellipsis – Ellipses
   3. If last letter is **'s'** or **'z'** and second last letter is not vowel a
   4. If the word ends in **'ato'**
      Potato – Potatoes
      Tomato – Tomatoes

b. If last letter is **'z'** and second last letter is vowel **'a'** or **'e'** then append **'zes'**
   Quiz – Quizzes

c. If last letter is **'s'** and second last letter is vowel **'a'** or **'e'** the append **'ses'**
   Gas – Gasses

d. If the noun ends with **'ff'** then append **'s'**
   Staff – Staffs
   Stuff – Staffs
   Lay-off – Lay-offs

e. If the noun ends with **'f'** or **'fe'**, the **f** is changed to **'ve'** before adding the **'s'** to form the plural version.
   Wife – Wives,
   Wolf – Wolves

f. If the singular noun ends in **'on'**, then **''on''** is changed to **'a'**.
   Phenomenon – Phenomena
   Criterion – Criteria

1. If the singular noun ends in **'us'**, then **'us'** is changed to **'i'**.
   Cactus – Cacti
   Focus – Foci

g. If last letter is **'y'** and second last letter is vowel then append **'s'** otherwise **'ies'**
   City – Cities
   Puppy – Puppies
   Ray – Rays
   Roy – Boys

h. For all other options append **'s'**

Input array

Student
Party
Quiz
Wolf

After function call Array should contain

Students
Parties
Quizzes
Wolves

Carefully reallocate memory to store plural words.

- You cannot change the function prototypes given in the questions.
- You cannot use break or goto statements. Breaks are allowed in switch cases.
- Built-in string functions are not allowed.
- Do not use new/extra strings wherever mentioned in the questions.

HAPPY PROGRAMMING