Program 4 – madlibs game

## http://4.bp.blogspot.com/-7lkrxgMK4ys/Ul8DL_lzYnI/AAAAAAAAAFs/D4CaYNdNmbc/s1600/MadLibs-Logo.png

# ASSIGNMENT & DUE DATES:

* **Assignment Date:** Monday, October 10,2019
* **Due Date:** Friday, November 15, 2019 (by 11:59pm)

# DESCRIPTION:

You will write a menu-based program that will allow the user to play Mad Libs! There should be a minimum of three different Mad Libs that the user may play.

## RULES:

* You must have a minimum of five functions
  + **main function** –
    - Call **showMenuGetChoice** to display the menu & get the user’s choice.
    - Open appropriate file according to choice.
    - The first line in the file contains how many questions or prompts there are for the particular mad libs game. You need to read this number from the file in order to know your array sizes.
    - Call **createStringArray** function to **dynamically allocate** an array of strings that will hold all the questions (or prompts). Then call the **createStringArray** function again to dynamically allocate an array of strings that will hold all the answers that the user enters during the game.
    - Read the questions/prompts from the appropriate file, close the file.
    - Call the **getInfoFromUser** function to get the answers from the user.
    - Call the **printStory** function to print the completed MADLIBS story.
    - Make sure to deallocate any arrays that were allocated during program.
    - The menu should continue to display until the user decides to end the game.
  + **showMenuGetChoice function** – show the menu, get the user’s choice, **validate** the user’s choice, return the users choice from this function
    - The menu is a list of MAD LIBS games that the user can choose from including 1) The Power of the Force (Star Wars Mad Libs), 2) Dog Days (Dog Ate My Mad Libs), and 3) Talk Like a Pirate (Pirates Mad Libs). Option 4 of the menu should be to end the game.
  + **createStringArray function** – accepts an integer indicating the size of the array to be created, dynamically allocates a new string array of this size & returns a pointer to this array.
  + **getInfoFromUser function** – this function accepts three parameters – pointer to question array, pointer to answer array, and the array size. The function supplies the user with each question or prompt (from the question array) and then allows the user to enter in an answer (put this in the answer array).
  + **printStory function** – this function accepts three parameters – the user’s choice (so you know what file to open), the answer array, and the array size. Open the appropriate file (for example starWarsText.txt), make sure the file exists and if it does, parse the file and print out answers from the answer array as necessary. Basically you will do the following repeatedly: read in a line from a file, print the line, print an answer from the array. Make sure to close the file.
* When passing an array to a function, use pointer notation instead of array notation.
* When accessing an array element, use pointer notation instead of array notation.
  + Example of array notation: myArray[x]
  + Example of pointer notation: \*(myArray+x)
* **You may not have any global variables – even if they are constant.** This is so you must practice sending data to/from functions!
* The user must be able to run the program as many times as they wish.
* You must have a comment block at the top of your program as well as a comment above each function telling the function name & purpose.
* You must indent your code properly.

## GIVEN:

* You will be given all necessary files for three Mad Libs games. For example, for the Star Wars mad libs game you are given two files:
  + **starWars.txt** – contains # of questions/prompts as well as the prompts
  + **starWarsText.txt** – contains the text needed to create the story. Between each line an answer will be inserted.

# WHAT TO TURN IN:

Zip all the following files in a zip file and upload the file to the Program 4 ilearn submission folder.

* program4.cpp
* starWars.txt
* starWarsText.txt
* pirate.txt
* pirateText.txt
* dog.txt
* dogText.txt

# SAMPLE OUTPUT:

A text version of the sample output is available in ilearn in a file called **program4\_sample\_output.txt**.





