

Lab 7: More Functions (extra credit up to +10pts on Exam 1)

This lab is extra credit. It is designed to give you more thorough experience with functions. It is optional, but can get you as much as an extra letter grade on exam 1 if you complete it entirely. You may also opt to complete the portions you are comfortable with or have the time to do for partial extra credit.

Here goes :)

Write a program that writes to several string objects. The program should have at least two string variables defined in main. The strings should be initialized with the value “000000000” (that’s nine 0’s).

You should write the following functions:

`void clearStr(string&)` - takes a reference to a string object as input and changes all 9 of it’s characters to 0.

`void printStr(string)` – takes a string object as input and prints the string to standard output. The string should be printed so that there is a newline character after every third 0. An example output for the 0 string would be:

```
000
000
000
```

`void fillStr(string&, char)` - takes a reference to a string object as input and a character and fills all 9 positions with the character passed into the function as a parameter.

`bool hasChar(string, char)` – checks if a string has a character and returns true if so and false if not

`bool hasCharAt(string, char, int)` – checks if a string has a character at a particular position and returns true if so and false if not

`bool hasCharAt(string, char, int, int)` – checks if a string has a character at a particular position and returns true if so and false if not. The position is based on row and column coordinates with the first parameter being the row and the second being the column, counting from top to bottom and left to right.

`void plotChar(string&, char, int)` – plots a character at a particular position in the string, the integer represents what element position in the string is to be filled

`void plotChar(string&, char, int, int)` – plots a character onto a string at a particular position using row and column coordinated (represented by the two integers passed in). The row should be the first integer

and the column should be the second integer. Thus, the function call `plotChar(str, '#', 2, 3)` where `str = "000000000"` should edit the `str` variable so that it becomes `00000#000` or when printed using `printStr()`:

```
000
00#
000
```

`void plotSquare(string&, char)` – this function should use one of the `plotChar()` functions to draw a square onto a string so that the zero string `"000000000"` should output to `stdout` like this when passed into `printStr()`:

```
###
#0#
###
```

`void plotX(string&, char)` - this function should use one of the `plotChar()` functions to draw an X onto a string so that the zero string `"000000000"` should output to `stdout` like this when passed into `printStr()`:

```
#0#
0#0
#0#
```

`void plotPlus(string&, char)` – this function should use one of the `plotChar()` functions to draw a + onto a string so that the zero string `"000000000"` should output to `stdout` like this when passed into `printStr()`:

```
0#0
###
0#0
```

Note that the `#` symbol in the three previous examples could be any character passed as input to the two functions.

`string createOrPlot(string, string, char)` – this function should take two strings of length nine as input and compare their values. The function should return a new string. The new string should start as all zeros inside the function. The two strings passed in as a parameter should be compared to each other. If both strings are not 0 at a particular position or one or the other are not zero, the new string should be written to at that same position with the character passed in as a parameter. Once all positions have been evaluated and tentatively written to, the new string should be returned.

`string createAndPlot(string, string, char)` - this function should take two strings of length nine as input and compare their values. The function should return a new string. The new string should start as all zeros inside the function. The two strings passed in as a parameter should be compared to each other. If both strings are not 0 at a particular position, the new string should be written to at that same position

with the character passed in as a parameter. Once all positions have been evaluated and tentatively written to, the new string should be returned.

String `createNotEqualPlot(string, string, char)` - this function should take two strings of length nine as input and compare their values. The function should return a new string. The new string should start as all zeros inside the function. The two strings passed in as a parameter should be compared to each other. If both strings are 0 or both strings are not 0, the new string should remain a 0 at that same position, otherwise, that position in the new string should be written to with the character passed in as a parameter. Once all positions have been evaluated and tentatively written to, the new string should be returned.

string `createNotPlot(string, char)` - this function should take a single string of length nine as input. The function should return a new string. The new string should start as all zeros inside the function. If the original string passed in as a parameter is zero at a particular position, the new string should be filled with the character parameter at that position. Once all positions have been evaluated and tentatively written to, the new string should be returned.

For the functions starting with the word 'create' you should make use of calls to `hasCharAt()` and `plotChar()` from within the function when appropriate to make less work and less code repetition.

Your main function should call each function and demonstrate that the functions you have written work as described in this lab. Just like any lab, you can receive partial credit for this lab and still have that credit applied. The points for this lab will be awarded to exam one. You can earn up to 10 additional points on the exam if you complete this lab in full.