Willis T. Allstead

CS 135L

23 February 2016

Your Title Goes Here

FEEDBACK:

Project 1:
Note: 1. Rather using “\n” for printing new line, you should use “endl”. (cout << endl). This is just not to print new line, rather it flushes the buffer, which will be explained by Nancy in future.
2. for stringCompare(), you should use “bool” as return type, rather using int (0/1/-1). You are basically checking if a char is greater then the other, for which you're expecting a true/false value.
3. You can avoid, “// increment index to go to next char in both strings” or, “// create index” or, “// file close” comments.

Project 2:
Nice work!
Note:
There is no point of creating a function that only takes the input and pass is by reference. You can make is more useful by moving all your display options into there from the main().

Project 3:
Excellent! Keep up the good work.
+5: Bonus. for comparing the agency name
Missing points:
-2: You should return true/ false rather 0/ 1 from stringComp()
Note:
1. Since, you're not doing anything with your “aptr” inside the “do..while” loop in main(), rather just passing the pointer in functions, which will be shallow copy rather hard copy.
So, you don't need to assign it to the “agencies” each time.

Project 4:
Missing points:
-10: SegFault!
File opened. Decoding...
Segmentation fault (core dumped)
You're assuming the key and jump for individual will sum up less then the pieces count, and so you can compare them both of them at once.
You should run a "for" loop for key. and use a wrap inside that.
Then you check if the jump is 0, until it is, do jump and also do wrap if necessary.
You're now trying to access NULL address! Using gdb gives:
Program received signal SIGSEGV, Segmentation fault.
0x00000000004013b3 in strConcat (str1=0x605449 "", str2=0x0) at project4.cpp:147
147 while(*str2 != '') {
(gdb) print str2
$2 = 0x0
-2: filename should be of exact size.

QUESTIONS:

None. The only problem I've had so far is in project 4 and it was explained very well in the feedback.

TOP 3 THINGS I WANT TA TO DISCUSS:

1) GDB more in-depth. I feel like we rushed through it and I feel like it is important.
2) VALGRIND more in-depth. Also how to use valgrind with gdb.
3) Overall more information on how to find exactly where each memory leak happens would be nice.