Handout 13
CSCI 334: Fall 2023

Resubmissions

You have 2 resubmissions at your disposal. You may resubmit the midterm exam and any lab except final project labs. You may not resubmit the final project. A high-quality resubmission recoups up to 50% of the missing points.

Resubmissions must be turned in before the end of the final exam reading period.

I really do read every one of these. Please take your time and do your work carefully and neatly so that I can understand what you mean. I want to give you back the maximum amount of points, so make it easy for me to do so.

Resubmission must include:

1. the original solution,
2. the updated solution, and
3. a document explaining your mistakes.

This document must be typed or it will not be accepted.

Goal

Critically, your explanation document must describe, in plain English, what you did wrong, why you made the mistake, and how your new solution fixes the problem. If you learned something from your mistake, tell me. In short, the goal of this exercise is to give you a second chance to demonstrate that you learned a concept.

How to Resubmit

For code resubmissions:

You should reorganize your code repository so that your resubmission is in a new branch called `resubmission`.

When you are done, commit your files and then visit `https://forms.gle/t4mfbBkY29UtBtyR8`to let me know that your resubmission is ready for review.

For a midterm resubmission:

Your resubmission should be on paper. You should give me:

1. your original exam, and
2. a printed document explaining your mistakes that includes reworked solutions.

I recommend using a paper clip to bind the two documents together so that I can pull them apart and compare them side-by-side. As with code resubmissions, be sure to tell me that you submitted one by filling out the form at `https://forms.gle/t4mfbBkY29UtBtyR8`.

Midterm Resubmission

1. True or False

(c) I said true when the answer is false. BNF actually stands for Backus-Naur form. I forgot to review this part.

2. Troubleshooting?

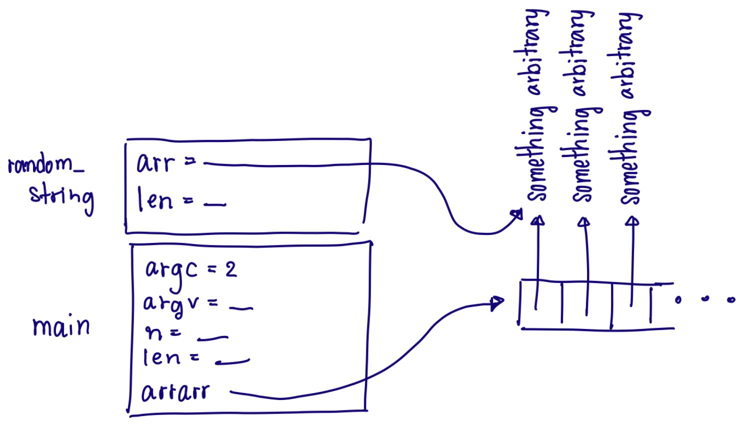
My fix was slightly wrong. Righr before calling $random\_string()$, I added

```
    char * arrarr[i] = malloc(sizeof(char)*MAXLEN);
```

when what I should have added is

```
    arrarr[i] = malloc(sizeof(char)*MAXLEN);
    mcheck(arrarr[i]);
```

There is no need for ”char *” because I am not declaring $arrarr$.
I got my explanation and drawing wrong. In my drawing, I had $arrarr[i]$ pointing back to a call stack because I thought the program would automatically allocate memory on a call stack if we did not $malloc()$. What I should have said is that without allocating sub-array $arrarr[i]$, the address currently living in the sub-array is arbitrary so the value referred to by the sub array is also arbitrary. When we call $memset()$ or manipulating $arrarr[i]$ in $random\_string()$, we are likely to get memory errors. Below is what I should have drawn.



3. Mad Libs for grownups

I said ”The wonderful green child laughed.” is a valid sentence in the language when it is not. I was not careful enough to notice that it was ”The”, not ”the”. Otherwise, ”the wonderful green child laughed.” would be in the language.