CSc 3320: Systems Programming

Spring 2021

Homework

4: Total points 100

Submission instructions:

1. Create a Google doc for each homework assignment submission.
2. Start your responses from page 2 of the document and copy these instructions on page 1.
3. Fill in your name, campus ID and panther # in the fields provided. If this information is missing in your document TWO POINTS WILL BE DEDUCTED per submission.
4. Keep this page 1 intact on all your submissions. If this *submissions instructions* page is missing in your submission TWO POINTS WILL BE DEDUCTED per submission.
5. Each homework will typically have 2-3 PARTS, where each PART focuses on specific topic(s).
6. Start your responses to each PART on a new page.
7. If you are being asked to write code copy the code into a separate txt file and submit that as well.
8. If you are being asked to test code or run specific commands or scripts, provide the evidence of your outputs through a screenshot and copy the same into the document.
9. Upon completion, download a .PDF version of the document and submit the same.

Full Name: Hawa Sylla

Campus ID: hsylla2

Panther #: 002-31-2868

ALL PROGRAMS MUST BE COMMENTED. YOUR SOLUTION WILL NOT BE ACCEPTED IF THERE ARE NO COMMENTS IN YOUR SCRIPT. Also note that the comments MUST be useful and not be random.

PART 1: 40pts
Must incorporate use of Functions and Pointers

1. Write a C program `checkPasswd.c` to check if the length of a given password string is 10 characters or not. If not, deduct 5 points per missing character. If the total deduction is greater than 30 points, print out the deduction and message "The password is unsafe! Please reset."; otherwise, print out "The password is safe."

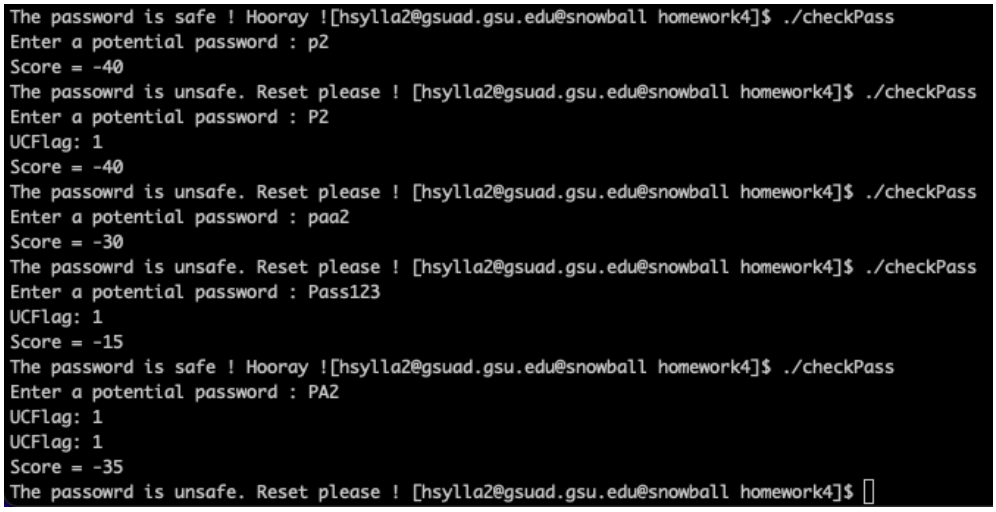
```
[hsylla2@gsuad.gsu.edu@snowball homework4]$ ./checkPass
Enter a potential password : Pass
Score = -30
The passowrd is unsafe. Reset please ! [hsylla2@gsuad.gsu.edu@snowball homework4]$ ./checkPass
Enter a potential password : PasswordBayBee
Score = 20
The password is safe ! Hooray ![hsylla2@gsuad.gsu.edu@snowball homework4]$
```

```
#include<stdio.h>

int main()
{

        char ch;

        printf("Enter a potential password : "); //this is a prompt for the user to enter the pass

        int count = 0; // this is to count the length of the password


        do{
                ch = getchar(); //reading each character that is entered
```

```
            count ++; // with each character entered, we increment the count of the user's pass
        }

        while(ch != '\n'); //while the user has not pressed enter;

        int score = -(10 - count + 1) * 5; // we are keeping track of the count where the minimum has to be 10. when the count is less than ten, it will end up having a negative score.

        printf("Score = %d\n",score);


        if(score <= -30)

            printf("The password is unsafe. Reset please ! "); //per the instructions, when the decrement is less than 30, the User will be prompted to start over and make a new password.

        else
            printf("The password is safe ! Hooray !");


        return 0;


}
```

2. Similar to above question, update the C program `checkPasswd.c` to check if a password is safe or by not by checking only the evaluation criteria below. It will still print out the final score, and "safe" or "unsafe" when deduction is more than 30 points.

- Missing lower case -20 points
- Lack of capital letters -20 points
- Missing numbers -20 points
- More than 2 consecutive characters (e.g. 123 or abc) -20 points




```
#include<stdio.h>
#include<ctype.h>
#define TRUE 1
#define FALSE 0


int main()
{
        int uCFlag = FALSE;
        int lCFlag = FALSE;
        int numFlag = FALSE;
        char ch;

        printf("Enter a potential password : "); //this is a prompt for the user to enter the pass
```

```
int count = 0; // this is to count the length of the password


do{
ch = getchar(); //reading each character that is entered

count ++; // with each character entered, we increment the count of the user's pass

// condition for checking for lowercase and capital letter

if(ch>64 && ch<91)
        {
        uCFlag = TRUE;
        printf("UCFlag: %d\n",uCFlag); //debugger to check if True flag was flagged
        }
        else
        {
        lCFlag = TRUE;
        }
        numFlag = isdigit(ch);

}

while(ch != '\n'); //while the user has not pressed enter;

int score = -(10 - count + 1) * 5 ; // we are keeping track of the count where the minimum has to be 10. when the count is less than ten, it will end up having a negative score.

// The following is the additional conditions as requested by part 2

if(uCFlag = FALSE)
{
score - 20;
}
else if(lCFlag = FALSE)
{
score - 20;
}
else if(numFlag = 0)
{
score - 20;
}
```

```
// ^condition for missing number

printf("Score = %d\n",score);


if(score <= -30)

printf("The password is unsafe. Reset please ! "); //per the instructions, when the decrement is less than 30, the User will be prompted to start over and make a new password.

else
printf("The password is safe ! Hooray !");
```

Part II : 40pts
Must incorporate the use of Functions and Pointer arrays

3. Write a program that reads a message (can be characters, numeric or alphanumeric) and checks whether it is a palindrome (the characters in the message are the same when read from left-to-right or right-to-left).

```
#include<stdio.h>

#include<string.h>

#define MAX_LEN 100
//I'm just setting the Maximum length of the string to be 100 characters.
//

int is_palindrome(char *seq, int seq_length){

        int isPal = 1; //setting the sequence to be palindrome by default

        char rev_seq[MAX_LEN]; // keeping track of the reverse of the sequence/string

        int j=0; //index of reverse

        for(int i = seq_length-1; i >= 0; i--) //for the length of the sequence
        {
                rev_seq[j]=seq[i]; //putting the reverse of the sequence into the reverse sequence holder
                j++; //increment j so it can keep going through the sequence
        }

        rev_seq[j]='\0'; //ending sequence with null for string

        for(int i = 0; i< seq_length; i++)
        {
```

```
                if(seq[i]!= rev_seq[i]) // if the sequence character in the index does NOT match the reverse character in the index,
                { isPal = 0;
                break;
                }
        }

return isPal;

}

int main()
{

char seq[MAX_LEN];

printf("Enter the Sequence please ! : ");
//^prompt to ask user for possible palindrome
scanf("%s",seq);
//^user input
if(is_palindrome(seq,strlen(seq))) //taking the input information, so the user inputted sequence, and the program determined sequence length.
{
 printf("%s is a palindrome ! Yay !\n",seq); // if isPalindrome is true, then return a sentence signifying that to the user.
}

else

{

printf("%s is not palindrome, sorry !\n",seq); // if isPalindrome is false, then return a sentence signifying that it is not a palindrome.

}

return 0;
```

```
}
```

```
[hsylla2@gsuad.gsu.edu@snowball homework4]$ ls
codeLookUp  codeLookUp.c  palin  palin.c  palindrome.c  swap  swap.c
[hsylla2@gsuad.gsu.edu@snowball homework4]$ ./palin
Enter the Sequence please ! : racecar
racecar is a palindrome ! Yay !
[hsylla2@gsuad.gsu.edu@snowball homework4]$ ./palin
Enter the Sequence please ! : snowball
snowball is not palindrome, sorry !
[hsylla2@gsuad.gsu.edu@snowball homework4]$
```

4. Write a program that will swap two variables without the use of any third variable. Utilize this program to write a program that reads two sentences that contain alphanumeric characters and the program must swap all the numerics in sentence1 with alphabet characters from sentence 2 and vice-versa. Keep the lengths of the sentences as identical.

Part III : 20pts
Must incorporate Functions, Pointers or PointerArrays, and Structures or Unions

5. Write a program that asks the user to enter an international dialing code and then looks it up in the country_codes array (see Sec 16.3 in C textbook). If it finds the code, the program should display the name of the corresponding country; if not, the program should print an error message. For demonstration purposes have at least 20 countries in your list.
   (Programming Project 1 on pg412 in C textbook)

```
#include<stdio.h>

#define COUNTRY_COUNT \
        ((int) (sizeof(country_codes) / sizeof(country_codes[32])))
//#define COUNTRY_COUNT 32
//above, we are defining the number of countries by taking the amount of country codes in the
//following struct

struct dialing_code {
        char *country;
        int cCode;
};
//^ this struct makes up the 'dialing code', consisting of the country and the country's code.

const struct dialing_code country_codes[32] =
{{"Argentina",          54}, {"Bangladesh",    880},
  {"Brazil",            55}, {"Burma (Myanmar)",  95},
  {"China",             86}, {"Colombia",       57},
  {"Congo, Dem. Rep. of", 243}, {"Egypt",              20},
  {"Ethiopia",          251}, {"France",        33},
  {"Germany",           49}, {"India",          91},
  {"Indonesia",         62}, {"Iran",           98},
```

```
  {"Italy",          39}, {"Japan",          81},
  {"Mexico",         52}, {"Nigeria",        234},
  {"Pakistan",       92}, {"Philippines",    63},
  {"Poland",         48}, {"Russia",         7},
  {"South Africa",   27}, {"South Korea",    82},
  {"Spain",          34}, {"Sudan",          249},
  {"Thailand",       66}, {"Turkey",         90},
  {"Ukraine",        380}, {"United Kingdom",  44},
  {"United States",  1}, {"Vietnam",         84}};
//^ the array of countrycode structs, where the given countries and phone codes lie
// for the example, I just used the ones from the textbook.
int main(void)
{
        int code, i;
        printf("Enter Dial code ! : ");
        scanf("%d", &code);
        for (i = 0; i < COUNTRY_COUNT; i++)  // this loop will iterate and search through the length of the Country Count struct
        {
        if (code == country_codes[i].cCode) // if the code that was entered matches the code in the struct...
        {
                printf("The country with dialing code %d is %s\n", code, country_codes[i].country);
                return 0; // when it it found, the country along with its code is returned to user.
        }
```

```
[hsylla2@gsuad.gsu.edu@snowball homework4]$ vi codeLookUp.c
[hsylla2@gsuad.gsu.edu@snowball homework4]$ gcc codeLookUp.c -o codeLookUp
[hsylla2@gsuad.gsu.edu@snowball homework4]$ vi codeLookUp.c
[hsylla2@gsuad.gsu.edu@snowball homework4]$ gcc codeLookUp.c -o codeLookUp
[hsylla2@gsuad.gsu.edu@snowball homework4]$ ./codeLookUp
Enter Dial code ! : 54
The country with dialing code 54 is Argentina
[hsylla2@gsuad.gsu.edu@snowball homework4]$
```