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: Aparna Mandapaka

Campus ID: amandapaka2

Panther #: 002553239

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."

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

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).
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

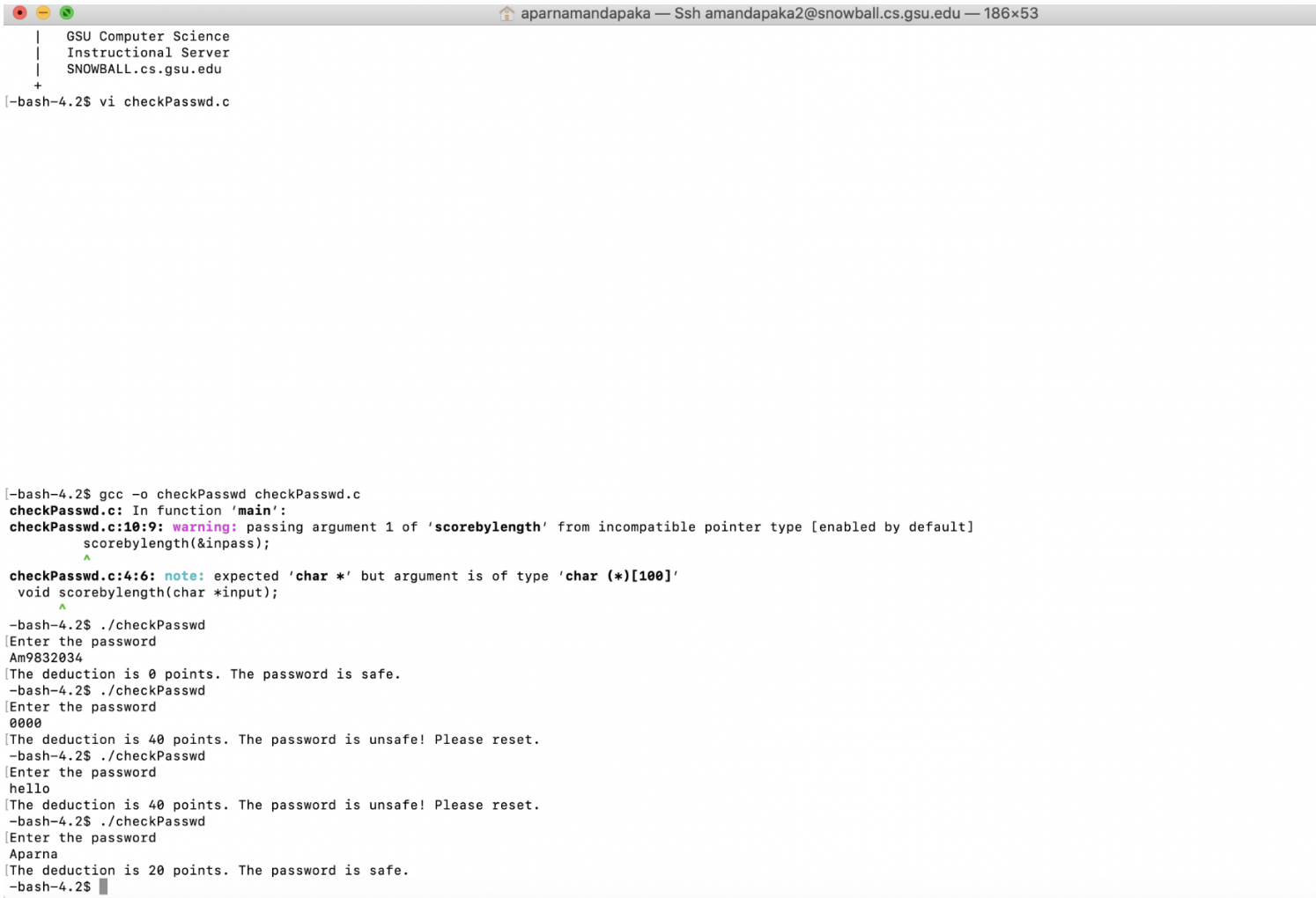
must swap all the numerics in sentence1 with alphabet characters from sentence 2 and vice-versa. Keep the lengths of the sentences as identical.

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)

Program answers

1. checkPasswd.c



aparnamandapaka — Ssh amandapaka2@snowball.cs.gsu.edu —

```c
#include <stdio.h>
#include <string.h>

void scorebylength(char *input);

int main(){
        char ch[100]; //holds the values entered by user
        printf("Enter the password\n");
  //scans the answer or input
         scanf("%s", &ch);//stores the user given input in "ch" variable
        scorebylength(&ch);
        return 0;
   }

void scorebylength(char *input){
       int i, j = 0; //the i and j are initial variables

         int leng = strlen(input);//calculates and checks the length of the password provided

        int points_score = 0;//assigning the intializaing to deduct to 0

         int cap_strike = 0;//cap_strike checks if there are any capital letter in the password

       int num_strike = 0;//this checks if the numbers are in the password

        int low_strike = 0;//this checks for lowercase alphabets

        int comb_strike = 0;//this checks that there are not 2 more consecutive letters and numbers in password

        for(i = 0; i < leng; i++){ //this checks for uppercase, lowercase and numbers in password

          if((input[i]>='a' && input[i]<='z')||(isdigit(input[i]))){
             cap_strike = cap_strike + 1; //checks for lowercase letter or number

      }
         if(!(isdigit(input[i]))){
           num_strike = num_strike + 1;
                }
  if((input[i]>='A' && input[i]<='Z')||(isdigit(input[i]))){//checks if the character is upper case or number


        low_strike = low_strike + 1;
           }
      }

 while(j <= leng-3){
      if(((int)input[j])+1 == (int)input[j+1] && ((int)input[j])+2 == (int)input[j+2]){
                        //Adding 1 concecutive combination score to the strikes counter
         comb_strike = comb_strike + 1;
          }
  //increases jth index to reach other 3 characters
```

```
  //increases jth index to reach other 3 characters

    j++;
 }

      if(comb_strike > 0){
         points_score = points_score + 20;
        }
 if(cap_strike == leng){
                //Adding 20 points to the deducted score
                points_score = points_score + 20;
 }
if(num_strike == leng){
                //Adding 20 points to the deducted score
       points_score = points_score + 20;
   }

if(low_strike == leng){
                //Adding 20 points to the deducted score
   points_score = points_score + 20;
 }
if(points_score > 30){
//if the points are calculated is higher than 30, it gives an error
 printf("The deduction is %d points. The password is unsafe! Please reset.\n", points_score);
        }else{
//if not it prints the password is safe
 printf("The deduction is %d points. The password is safe.\n", points_score);
        }
}
```

checkPasswdpt2.c

aparnamandapaka — Ssh amandapaka2@snowball.cs.gsu.edu — 178×53

```
                        ^
checkPasswdpt2.c:17:27: note: each undeclared identifier is reported only once for each function it appears in
-bash-4.2$ vi checkPasswdpt2.c
-bash-4.2$ gcc -o checkPasswdpt2 checkPasswdpt2.c
checkPasswdpt2.c: In function 'main':
checkPasswdpt2.c:11:2: warning: passing argument 1 of 'scorebylength' from incompatible pointer type [enabled by default]
  scorebylength(&ch);
  ^
checkPasswdpt2.c:4:6: note: expected 'char *' but argument is of type 'char (*)[100]'
 void scorebylength(char *input);
      ^
-bash-4.2$ ./checkPasswdpt2
Enter the password
0000
The deduction is 30 points. The password is safe.
-bash-4.2$ ./checkPasswdpt2
Enter the password
abc
The deduction is 35 points. The password is unsafe! Please reset.
-bash-4.2$ ./checkPasswdpt2
Enter the password
aparna
The deduction is 20 points. The password is safe.
-bash-4.2$ ./checkPasswdpt2
Enter the password
APARNA
The deduction is 20 points. The password is safe.
-bash-4.2$ ./checkPasswdpt2
Enter the password
Aparna2090
The deduction is 0 points. The password is safe.
```

aparnamandapaka — Ssh amandapaka2@snowball.cs.gsu.edu — 202×53

```
#include <stdio.h>
#include <string.h>

void scorebylength(char *input);

int main(){

         char ch[100]; //holds the values entered by user

         printf("Enter the password\n");
          scanf("%s", &ch); //stores the user given input in "ch" variable


 scorebylength(&ch);//checks the length of the password

 return 0;
}

void scorebylength(char *input){
        int leng = strlen(input);//calculates and checks the length of the password provided

        int points_score = 5*(10-leng);//takes off 5 points for every character length that is greater than 10

        if(points_score > 30){ //checks the deducation is greater than 30, if it is then it displays the below message
 printf("The deduction is %d points. The password is unsafe! Please reset.\n", points_score);
        }else{ //if the deducation is not greater than 30, then it displays that the password is safe

if(points_score < 0){ //it resets the deduction points to 0
 points_score = 0;
     }

printf("The deduction is %d points. The password is safe.\n", points_score);
        }
}
~
```

checkPalindrome.c

aparnamandapaka — Ssh amandapaka2@snowball.cs.gsu.edu — 139×53

```c
#include <stdio.h>
#include <string.h>

int main()
{

        char input[20]; //in this the character holds the input message
        printf("Enter Palindrome : "); //tells the user to enter the input
        scanf("%s", input);
        int LefttoRight = 0; //this starts reading from left to right
        int RighttoLeft = strlen(input) - 1; //this starts reading from right to left

        while (RighttoLeft > LefttoRight)
        {
                if(input[LefttoRight++] != input[RighttoLeft--]) //checks from Left to Right, if it doesn't match the palindrome
        {
                printf("%s is not a Palindrome", input);
                break; //if it doesn't match with the Palindrome, then the program breaks
        }
        else {
                printf("%s is a Palindrome", input);
        break; // if it matches with the Palindrome then it breaks too
        }
    }
        return 0; //if the character is different, then 0 is returned, stating that the input is not in the palindrome
}

~
```

aparnamandapaka — Ssh amandapaka2@snowball.cs.gsu.edu — 93×53

```
-bash-4.2$ vi checkPalindrome.c
-bash-4.2$ gcc -o checkPalindrome checkPalindrome.c
-bash-4.2$ ./checkPalindrome
Enter Palindrome : esste
esste is a Palindrome-bash-4.2$ APARNA
-bash: APARNA: command not found
-bash-4.2$ ./checkPalindrome
Enter Palindrome : APARNA
APARNA is a Palindrome-bash-4.2$ ./checkPalindrome
Enter Palindrome : foodisfav
foodisfav is not a Palindrome-bash-4.2$
-bash-4.2$
```

alphaNumeric.c

aparnamandapaka — Ssh amandapaka2@snowball.cs.gsu.edu — 102×53

```
alphaNumeric.c:(.text+0x20): warning: the `gets' function is dangerous and should not be used.
-bash-4.2$ ./alphaNumeric
Enter sentence 1
hello how are you?
Enter sentence 2
I'm doing great, how about you?
Inital string values of 1 and 2: hello how are you?      I'm doing great, how about you?
The messages are not of equal length.
-bash-4.2$ ./alphaNumeric
Enter sentence 1
Hello, how are you?
Enter sentence 2
Hello, how are you?
Inital string values of 1 and 2: Hello, how are you?     Hello, how are you?
Final values of strings 1 and 2: Hello, how are you?     Hello, how are you?
-bash-4.2$ ./alphaNumeric
Enter sentence 1
It's cold outside
Enter sentence 2
It's hot outside
Inital string values of 1 and 2: It's cold outside       It's hot outside
The messages are not of equal length.
-bash-4.2$
```

aparnamandapaka — Ssh amandapaka2@snowball.cs.gsu.edu — 136×53

```c
#include <stdio.h>
#include <string.h>

int swap(char *message1, char *message2);
int main(){
 char input1[200];//stores the characters of array from the first input
 char input2[200];//stores the characters of array from the second input
  printf("Enter sentence 1\n");//asks the user to enter the first sentence
   gets(input1); //reads the input provided by the user
 printf("Enter sentence 2\n"); //asks user type in second sentence
  gets(input2);//reads the second input provided by the user
 swap(&input1, &input2); //this calls the swap method to swap the strings
 return 0;
}

 int swap(char *message1, char *message2){//this is the swap method with two pointers
 int len1 = strlen(message1); //this finds the length of the first sentence provided by the user
int len2 = strlen(message2); //this finds the length of the second sentence provided by the user
 printf("Inital string values of sentence 1 and 2: %s \t %s\n", message1, message2);//displays two strings to the user
if(len1 != len2){
                printf("The messages are not of equal length.\n");
                return 0;
        }
  strcat(message1, message2);//adds both input 1 and 2 and combines into one sentence
 strncpy(message2, message1, len2);//copies the largest input and adds it to the other input
  printf("Final values of strings 1 and 2: %s \t %s\n", message1+len2, message2); //prints the strings after it has swapped

                return 0;
}
```

dailupCount.c

aparnamandapaka — Ssh amandapaka2@snowball.cs.gsu.edu — 111×53

```
-bash-4.2$ vi dailupCount.c
-bash-4.2$ gcc -o dailupCount dailupCount.c
-bash-4.2$ ./dailupCount
Enter the code of the country's phone numbers.
88
The country was not found in this directory!
-bash-4.2$ ./dailupCount
Enter the code of the country's phone numbers.
86
The country is China.
-bash-4.2$ ./dailupCount
Enter the code of the country's phone numbers.
1
The country is United States.
-bash-4.2$
```

aparnamandapaka — Ssh amandapaka2@snowball.cs.gsu.edu — 134×53

```
#include <stdio.h>

struct dialing_code{

 char *country; //holds the name of the country
        int code;//holds the code of the country
};

const struct dialing_code country_code[] = {
                {"Argentina", 54}, {"Brazil", 55}, {"Germany", 49}, {"Bangladesh", 880},
                {"India", 91}, {"Indonesia", 62}, {"Russia", 7}, {"Iran", 98},
                {"China", 86}, {"Ehiopia", 251}, {"France", 33}, {"Colombia", 57},
                {"Italy", 39}, {"South Africa", 27}, {"United Kingdom", 44}, {"South Korea", 82},
                {"United States", 1}, {"Poland", 48}, {"Turkey", 90}, {"Mexico", 52}, {"Philippines", 63}};
int ind_ext(int *code1);
        int main(){
        int con_code; //this holds the input of the country code by user
        printf("Enter the code of the country's phone numbers.\n"); //asks the user to enter the country code
        scanf("%d", &con_code);
         if(ind_ext(&con_code)){
           printf("The country is %s.\n", country_code[ind_ext(&con_code)-1].country);//prints the code and the country
        }else{
         printf("The country was not found in this directory!\n");
        }
        return 0;
        }

int ind_ext(int *code1){
        int i; //keeps track of the index
         int comp = *code1;//this assigns the value of the pointer to the other integer
         for(i = 0; i < 21; i++){
          if(country_code[i].code == comp){ //checks if the country code matches the index
         return(i+1);
                }
        }
        return 0;
}

~
~
~
```