CSc 3320: Systems Programming
Spring 2021
Homework
4: Total points 100
Kevin Gallardo

Full Name: Kevin Gallardo

Campus ID: kgallardowepster1

Panther #: 002-49-7849

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."

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ gcc -o checkPasswd checkPasswd.c
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./checkPasswd
Enter your password please: a
Total Deductions: 45
The password is unsafe! Please reset.
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./checkPasswd
Enter your password please: abcasdfasdf
Total Deductions: 0
The password is safe.
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$
```

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

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ gcc -o checkPasswd2 checkPasswd2.c
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./checkPasswd2
Enter your password please: a1D
Total Deductions: 0
The password is safe.
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./checkPasswd2
Enter your password please: aa
Total Deductions: 60
The password is unsafe! Please reset.
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./checkPasswd2
Enter your password please: 1
Total Deductions: 40
The password is unsafe! Please reset.
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./checkPasswd2
Enter your password please: Fa
Total Deductions: 20
The password is safe.
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$
```

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).

```
[kgallardowepsterl@gsuad.gsu.edu@snowball ~]$ gcc -o isPalindrome isPalindrome.c
[kgallardowepsterl@gsuad.gsu.edu@snowball ~]$ ./isPalindrome
Enter Message: no lemon no melon
Is a Palindrome
[kgallardowepsterl@gsuad.gsu.edu@snowball ~]$ ./isPalindrome
Enter Message: taco cat
Is a Palindrome
[kgallardowepsterl@gsuad.gsu.edu@snowball ~]$ ./isPalindrome
Enter Message: do geese see god
Is a Palindrome
[kgallardowepsterl@gsuad.gsu.edu@snowball ~]$ wow
-bash: wow: command not found
[kgallardowepsterl@gsuad.gsu.edu@snowball ~]$ ./isPalindrome
Enter Message: wow
Is a Palindrome
[kgallardowepsterl@gsuad.gsu.edu@snowball ~]$
```

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.

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ gcc -o swapStrings swapStrings.c
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./swapStrings
Enter string 1: what am I Doing?
Enter string 2: nothing
Strings must be same length
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./swapStrings
Enter string 1: abcdefg
Enter string 2: 1234567
before swap:
String 1: abcdefg
String 2: 1234567
after swap:
String 1: 1234567
String 2: abcdefg
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./swapStrings
Enter string 1: als2d3f4g5
Enter string 2: la2s3d4f5g
before swap:
String 1: als2d3f4g5
String 2: la2s3d4f5g
after swap:
String 1: la2s3d4f5g
String 2: als2d3f4g5
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./swapStrings
Enter string 1: 2ad2dgf3gd
Enter string 2: 2csd3fds4s
before swap:
String 1: 2ad2dgf3gd
String 2: 2csd3fds4s
after swap:
String 1:    d3  s4
String 2:    2d  3g
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ 
```

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)

```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ gcc -o searchCountry searchCountry.c
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./searchCountry
Enter coutnry code: 7
Russia
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./searchCountry
Enter coutnry code: 52
Mexico
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./searchCountry
Enter coutnry code: 91
India
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ ./searchCountry
Enter coutnry code: 52
Mexico
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$
```