country_codesCSC 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.

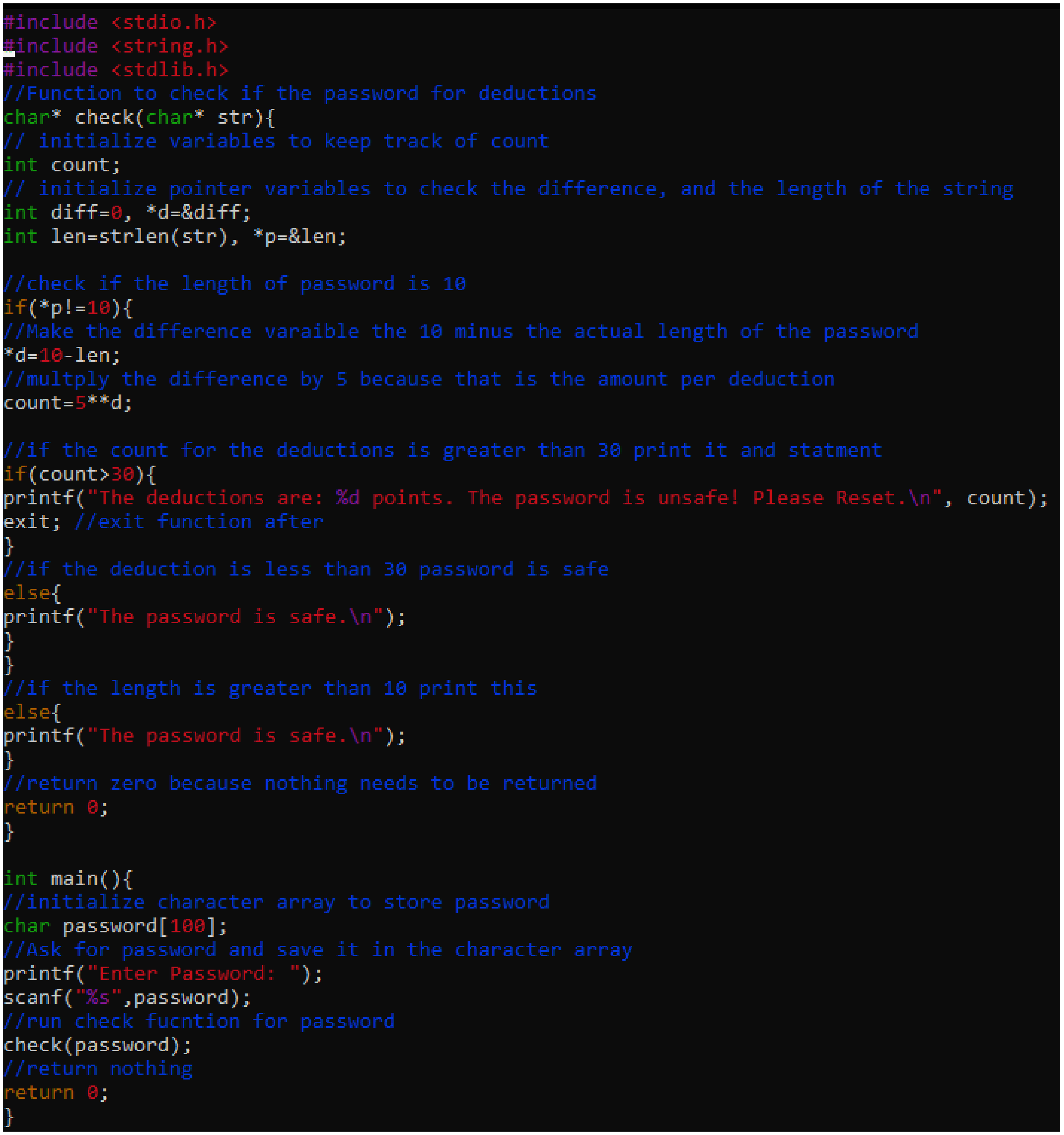
Full Name: Talaal Tahir

Campus ID: Ttahir1

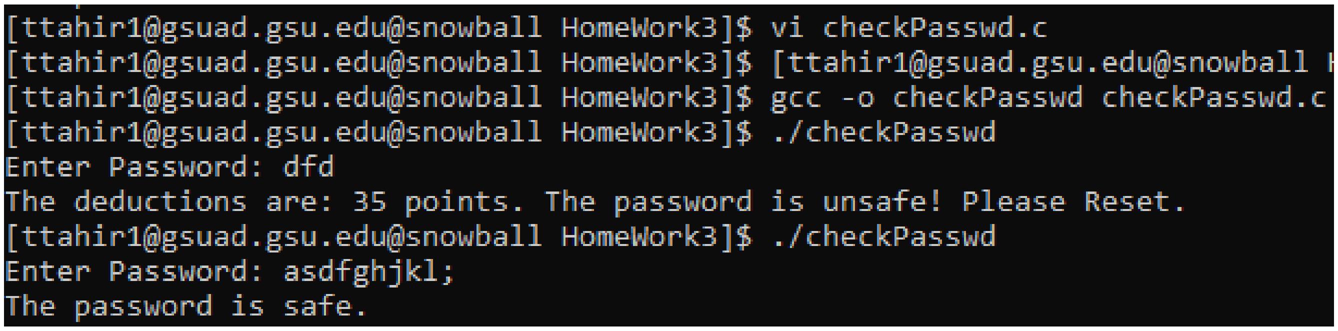
Panther #: 002504147

Part 1

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."

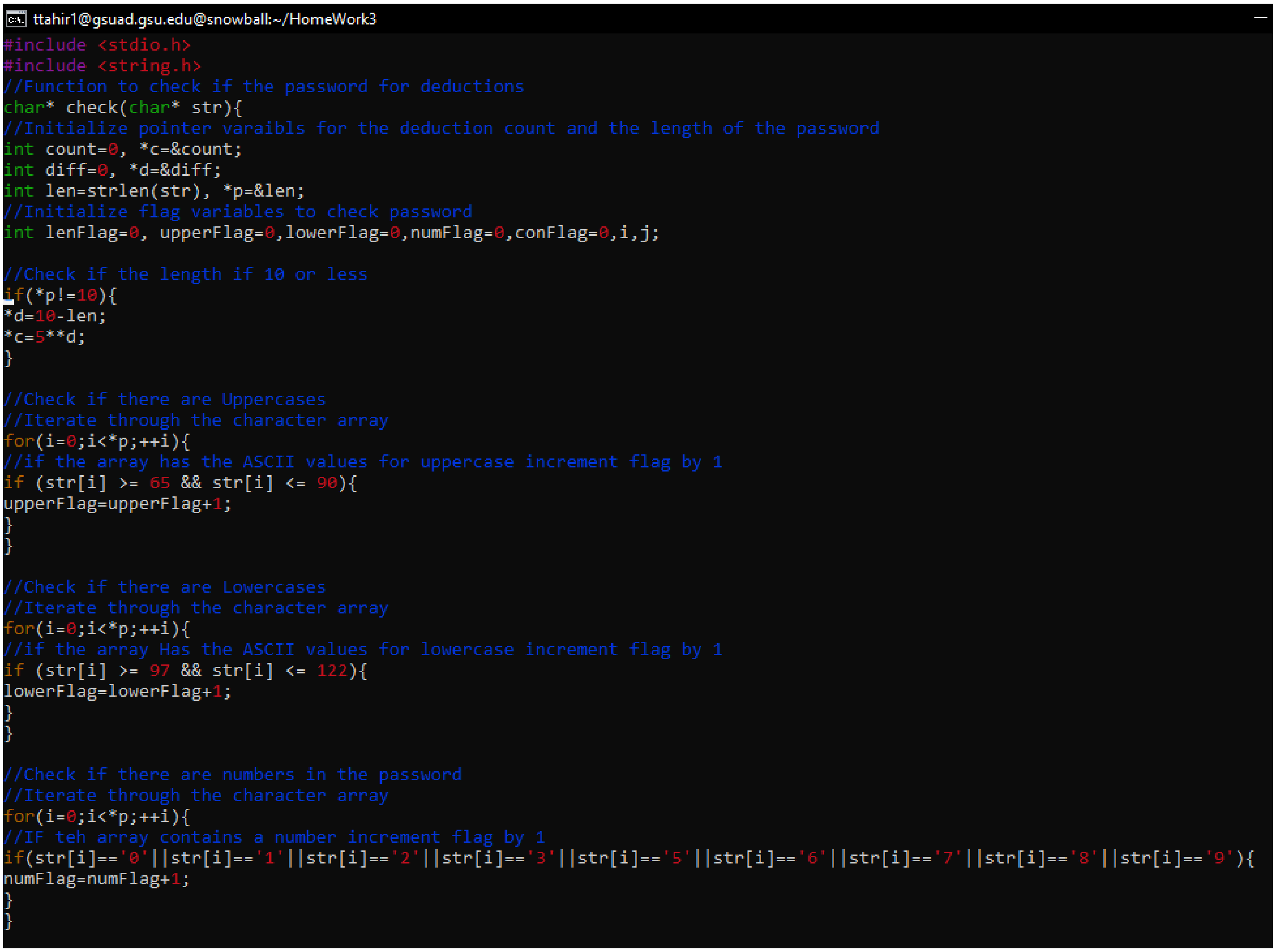


```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
//Function to check if the password for deductions
char* check(char* str){
// initialize variables to keep track of count
int count;
// initialize pointer variables to check the difference, and the length of the string
int diff=0, *d=&diff;
int len=strlen(str), *p=&len;

//check if the length of password is 10
if(*p!=10){
//Make the difference varaible the 10 minus the actual length of the password
*d=10-len;
//multply the difference by 5 because that is the amount per deduction
count=5**d;

//if the count for the deductions is greater than 30 print it and statment
if(count>30){
printf("The deductions are: %d points. The password is unsafe! Please Reset.\n", count);
exit; //exit function after
}
//if the deduction is less than 30 password is safe
else{
printf("The password is safe.\n");
}
}
//if the length is greater than 10 print this
else{
printf("The password is safe.\n");
}
//return zero because nothing needs to be returned
return 0;
}

int main(){
//initialize character array to store password
char password[100];
//Ask for password and save it in the character array
printf("Enter Password: ");
scanf("%s",password);
//run check fucntion for password
check(password);
//return nothing
return 0;
}
```



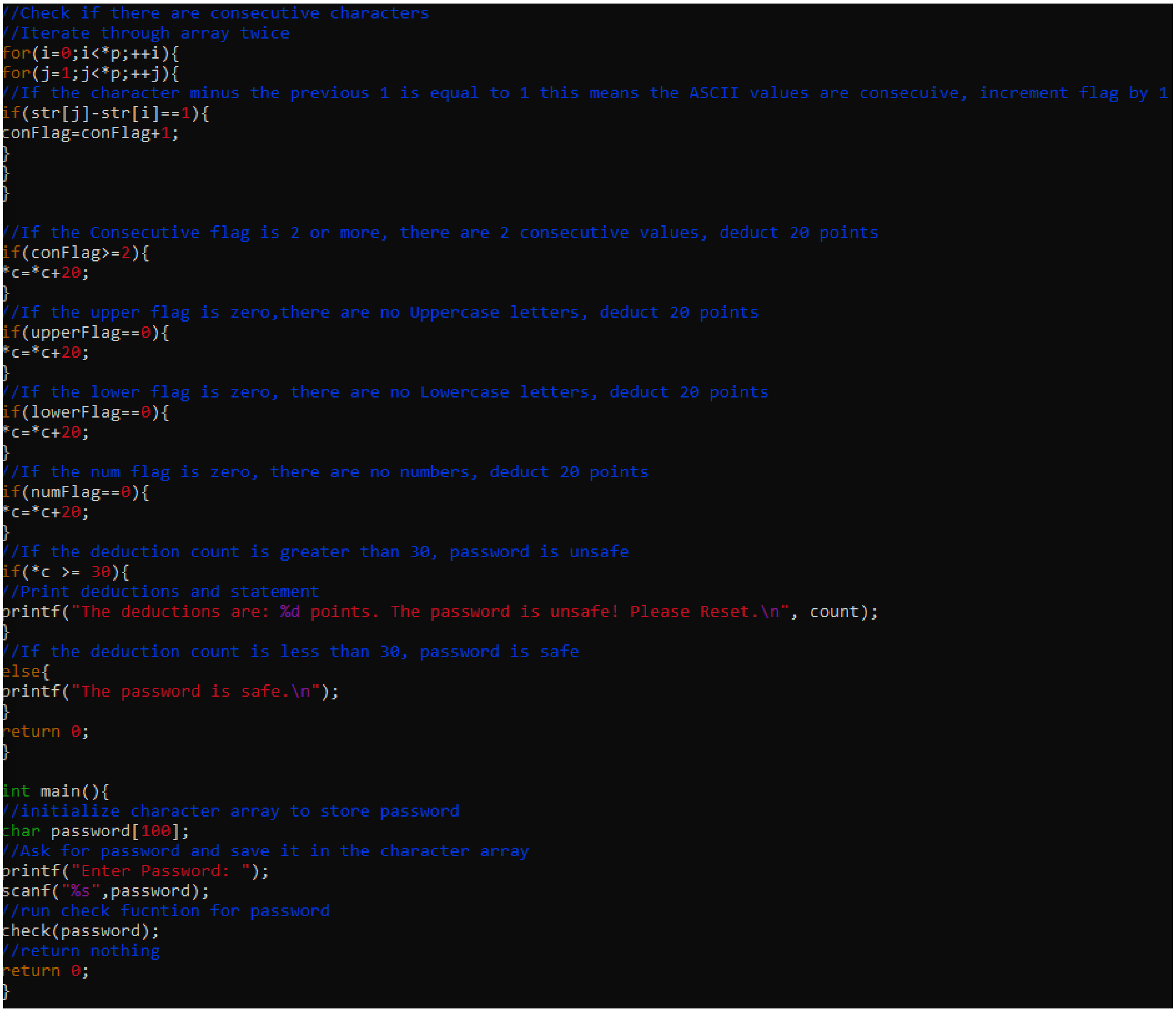
```
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ vi checkPasswd.c
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ [ttahir1@gsuad.gsu.edu@snowball H
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ gcc -o checkPasswd checkPasswd.c
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ ./checkPasswd
Enter Password: dfd
The deductions are: 35 points. The password is unsafe! Please Reset.
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ ./checkPasswd
Enter Password: asdfghjkl;
The password is safe.
```

2. Similar to the above question, update the C program checkPasswd.c to check if a password is safe or 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.

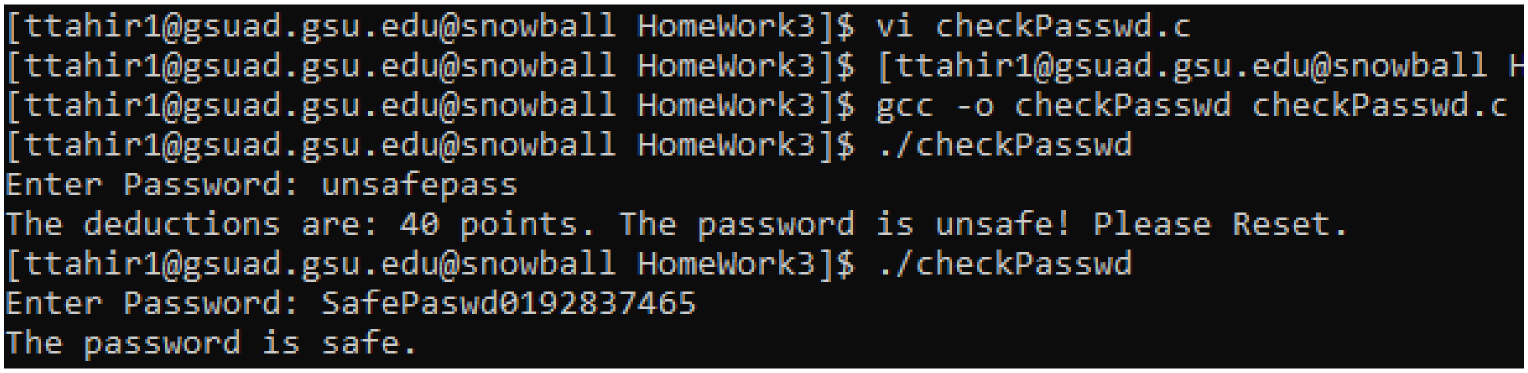


ttahir1@gsuad.gsu.edu@snowball:~/HomeWork3

```c
#include <stdio.h>
#include <string.h>
//Function to check if the password for deductions
char* check(char* str){
//Initialize pointer varaibls for the deduction count and the length of the password
int count=0, *c=&count;
int diff=0, *d=&diff;
int len=strlen(str), *p=&len;
//Initialize flag variables to check password
int lenFlag=0, upperFlag=0,lowerFlag=0,numFlag=0,conFlag=0,i,j;

//Check if the length if 10 or less
if(*p!=10){
*d=10-len;
*c=5**d;
}

//Check if there are Uppercases
//Iterate through the character array
for(i=0;i<*p;++i){
//if the array has the ASCII values for uppercase increment flag by 1
if (str[i] >= 65 && str[i] <= 90){
upperFlag=upperFlag+1;
}
}

//Check if there are Lowercases
//Iterate through the character array
for(i=0;i<*p;++i){
//if the array Has the ASCII values for lowercase increment flag by 1
if (str[i] >= 97 && str[i] <= 122){
lowerFlag=lowerFlag+1;
}
}

//Check if there are numbers in the password
//Iterate through the character array
for(i=0;i<*p;++i){
//IF teh array contains a number increment flag by 1
if(str[i]=='0'||str[i]=='1'||str[i]=='2'||str[i]=='3'||str[i]=='5'||str[i]=='6'||str[i]=='7'||str[i]=='8'||str[i]=='9'){
numFlag=numFlag+1;
}
}
```



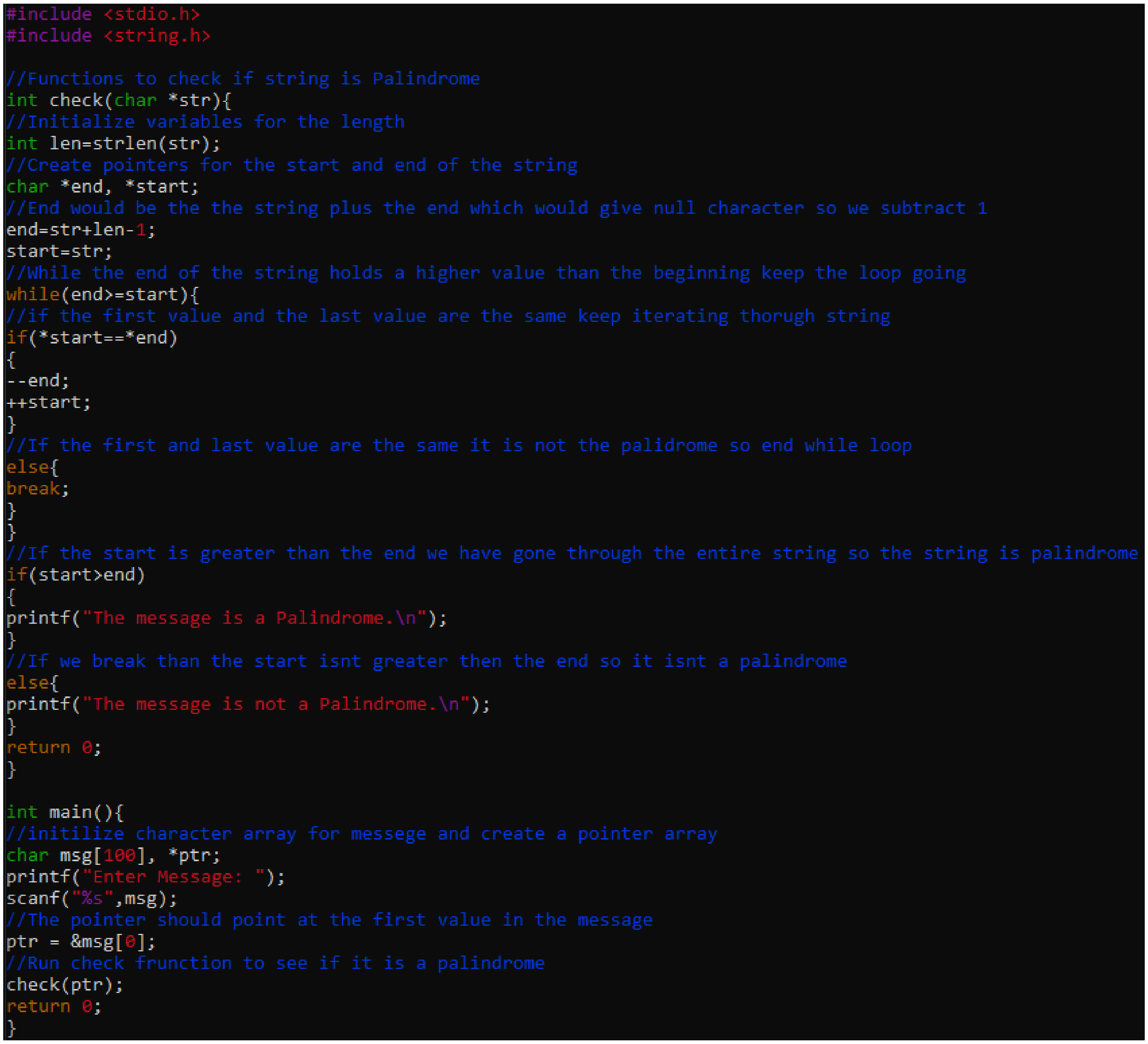
```c
//Check if there are consecutive characters
//Iterate through array twice
for(i=0;i<*p;++i){
for(j=1;j<*p;++j){
//If the character minus the previous 1 is equal to 1 this means the ASCII values are consecuive, increment flag by 1
if(str[j]-str[i]==1){
conFlag=conFlag+1;
}
}
}

//If the Consecutive flag is 2 or more, there are 2 consecutive values, deduct 20 points
if(conFlag>=2){
*c=*c+20;
}
//If the upper flag is zero,there are no Uppercase letters, deduct 20 points
if(upperFlag==0){
*c=*c+20;
}
//If the lower flag is zero, there are no Lowercase letters, deduct 20 points
if(lowerFlag==0){
*c=*c+20;
}
//If the num flag is zero, there are no numbers, deduct 20 points
if(numFlag==0){
*c=*c+20;
}
//If the deduction count is greater than 30, password is unsafe
if(*c >= 30){
//Print deductions and statement
printf("The deductions are: %d points. The password is unsafe! Please Reset.\n", count);
}
//If the deduction count is less than 30, password is safe
else{
printf("The password is safe.\n");
}
return 0;
}

int main(){
//initialize character array to store password
char password[100];
//Ask for password and save it in the character array
printf("Enter Password: ");
scanf("%s",password);
//run check fucntion for password
check(password);
//return nothing
return 0;
}
```



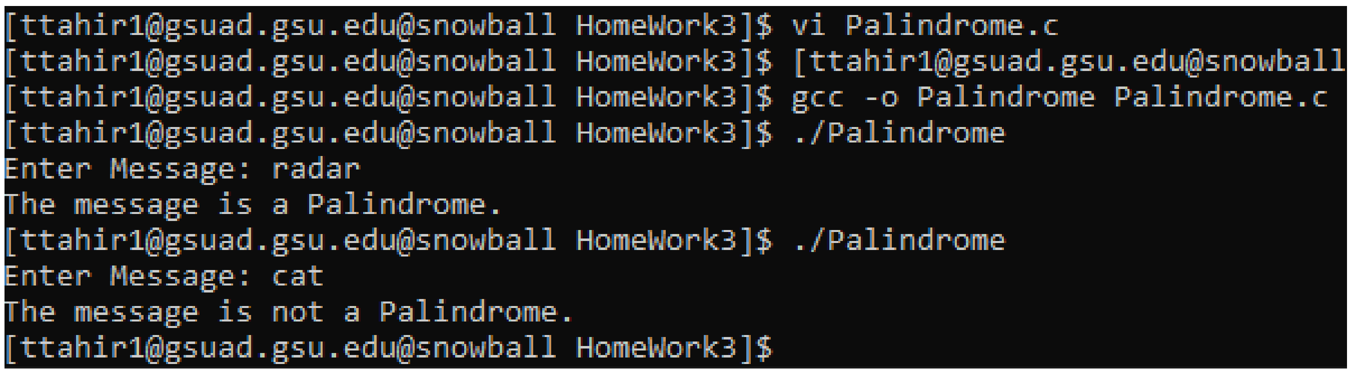
```
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ vi checkPasswd.c
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ [ttahir1@gsuad.gsu.edu@snowball H
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ gcc -o checkPasswd checkPasswd.c
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ ./checkPasswd
Enter Password: unsafepass
The deductions are: 40 points. The password is unsafe! Please Reset.
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ ./checkPasswd
Enter Password: SafePaswd0192837465
The password is safe.
```

Part 2

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).

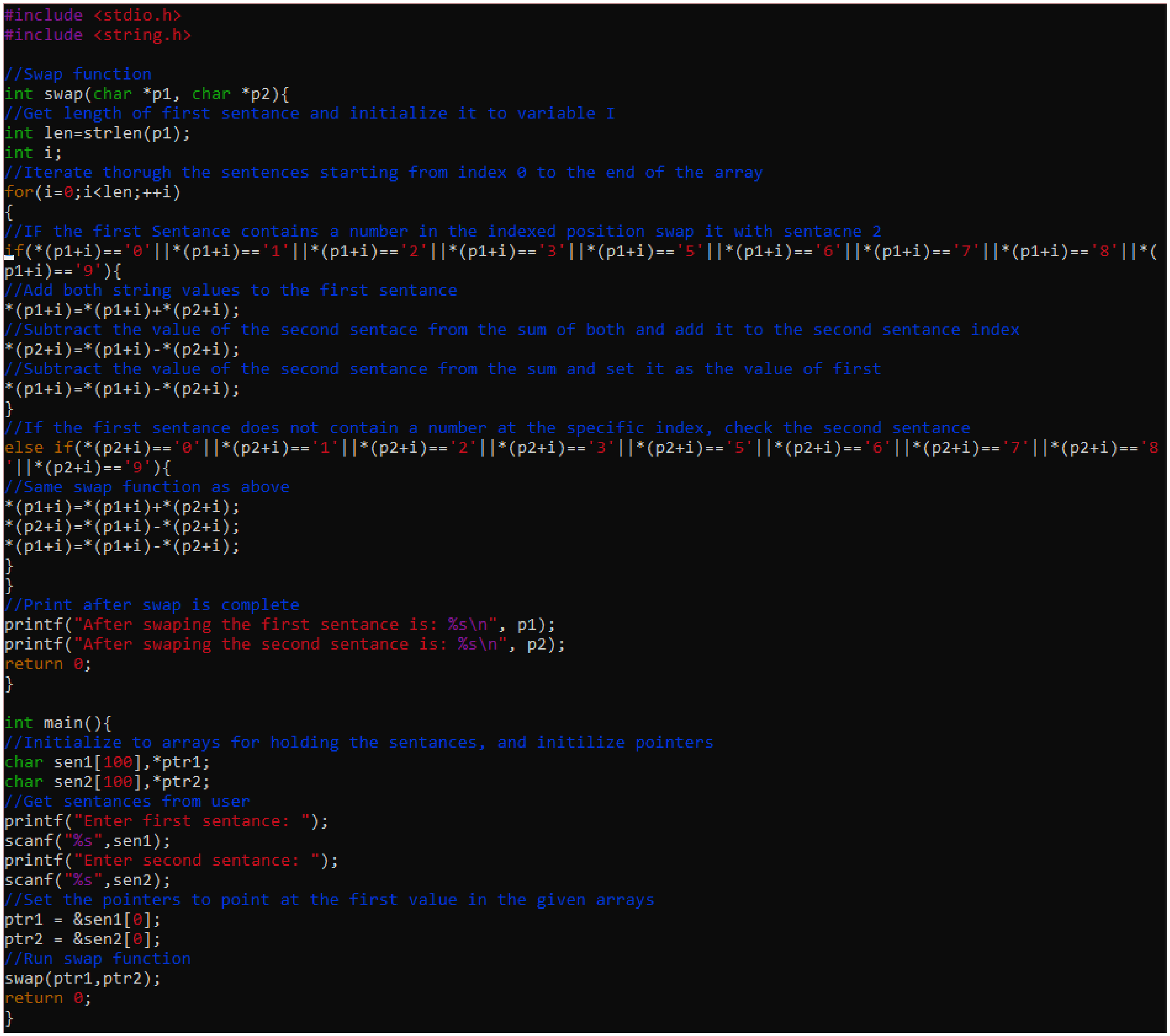


```c
#include <stdio.h>
#include <string.h>

//Functions to check if string is Palindrome
int check(char *str){
//Initialize variables for the length
int len=strlen(str);
//Create pointers for the start and end of the string
char *end, *start;
//End would be the the string plus the end which would give null character so we subtract 1
end=str+len-1;
start=str;
//While the end of the string holds a higher value than the beginning keep the loop going
while(end>=start){
//if the first value and the last value are the same keep iterating thorugh string
if(*start==*end)
{
--end;
++start;
}
//If the first and last value are the same it is not the palidrome so end while loop
else{
break;
}
}
//If the start is greater than the end we have gone through the entire string so the string is palindrome
if(start>end)
{
printf("The message is a Palindrome.\n");
}
//If we break than the start isnt greater then the end so it isnt a palindrome
else{
printf("The message is not a Palindrome.\n");
}
return 0;
}

int main(){
//initilize character array for messege and create a pointer array
char msg[100], *ptr;
printf("Enter Message: ");
scanf("%s",msg);
//The pointer should point at the first value in the message
ptr = &msg[0];
//Run check frunction to see if it is a palindrome
check(ptr);
return 0;
}
```

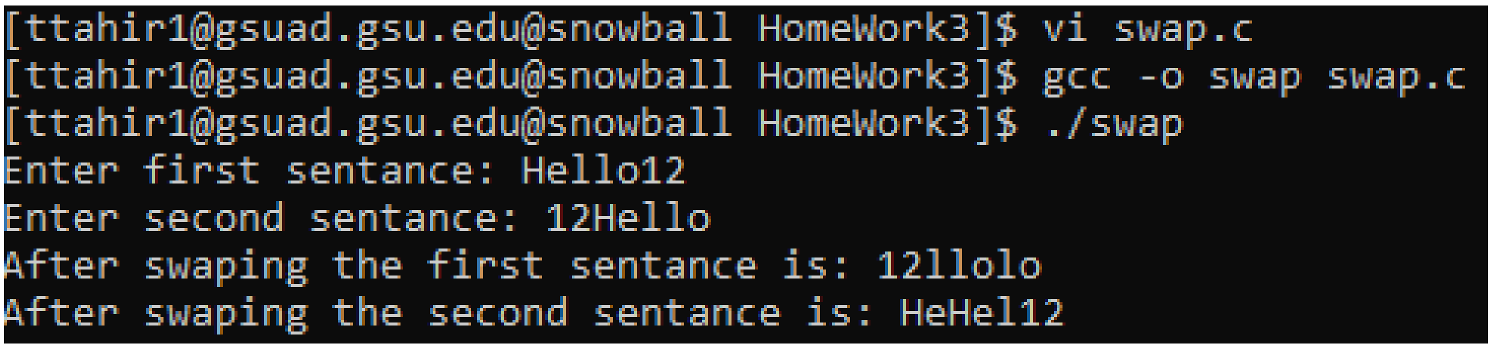


```
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ vi Palindrome.c
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ [ttahir1@gsuad.gsu.edu@snowball
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ gcc -o Palindrome Palindrome.c
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ ./Palindrome
Enter Message: radar
The message is a Palindrome.
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ ./Palindrome
Enter Message: cat
The message is not a Palindrome.
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$
```

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.



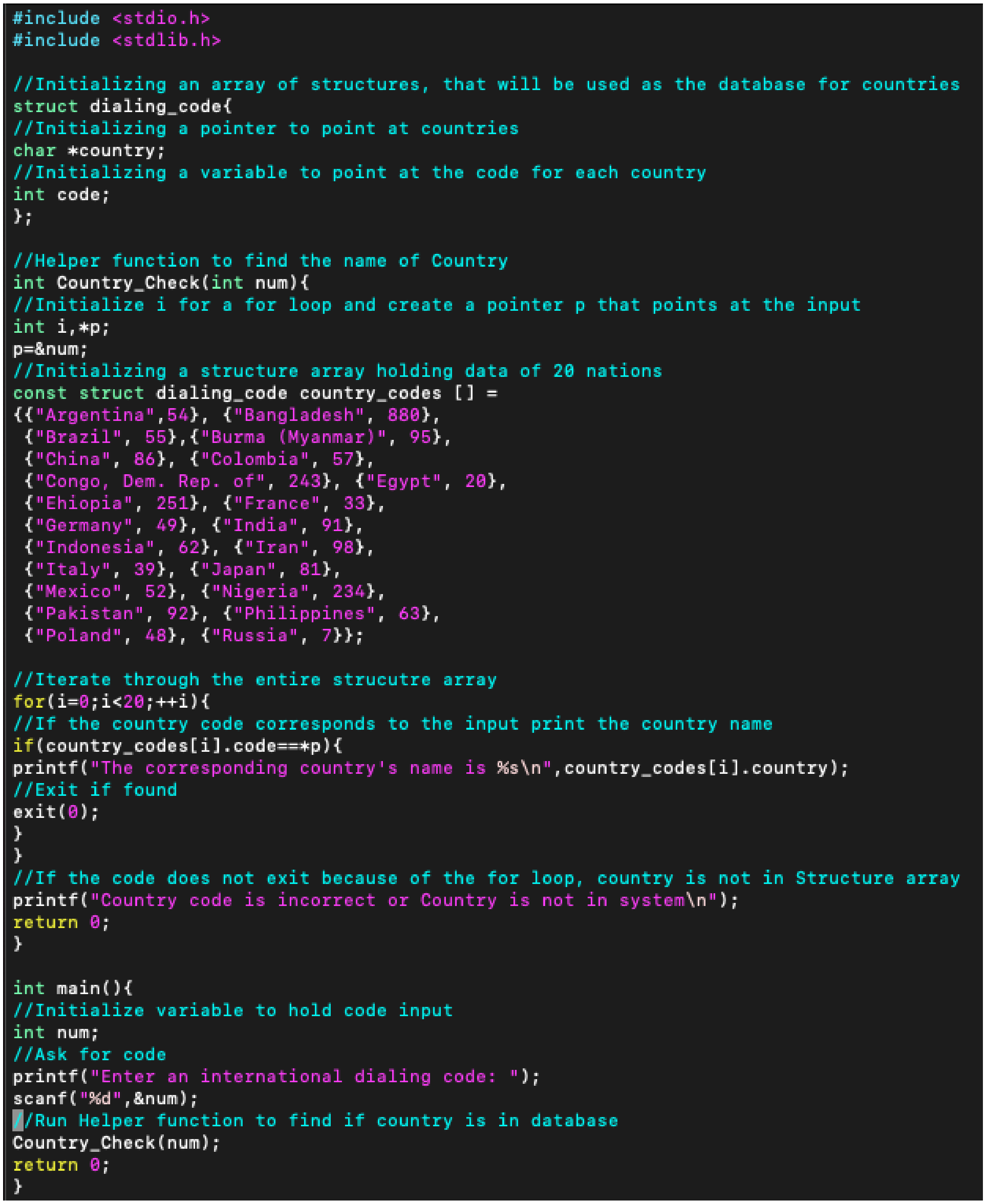
```c
#include <stdio.h>
#include <string.h>

//Swap function
int swap(char *p1, char *p2){
//Get length of first sentance and initialize it to variable I
int len=strlen(p1);
int i;
//Iterate thorugh the sentences starting from index 0 to the end of the array
for(i=0;i<len;++i)
{
//IF the first Sentance contains a number in the indexed position swap it with sentacne 2
if(*(p1+i)=='0'||*(p1+i)=='1'||*(p1+i)=='2'||*(p1+i)=='3'||*(p1+i)=='5'||*(p1+i)=='6'||*(p1+i)=='7'||*(p1+i)=='8'||*(
p1+i)=='9'){
//Add both string values to the first sentance
*(p1+i)=*(p1+i)+*(p2+i);
//Subtract the value of the second sentace from the sum of both and add it to the second sentance index
*(p2+i)=*(p1+i)-*(p2+i);
//Subtract the value of the second sentance from the sum and set it as the value of first
*(p1+i)=*(p1+i)-*(p2+i);
}
//If the first sentance does not contain a number at the specific index, check the second sentance
else if(*(p2+i)=='0'||*(p2+i)=='1'||*(p2+i)=='2'||*(p2+i)=='3'||*(p2+i)=='5'||*(p2+i)=='6'||*(p2+i)=='7'||*(p2+i)=='8
'||*(p2+i)=='9'){
//Same swap function as above
*(p1+i)=*(p1+i)+*(p2+i);
*(p2+i)=*(p1+i)-*(p2+i);
*(p1+i)=*(p1+i)-*(p2+i);
}
}
//Print after swap is complete
printf("After swaping the first sentance is: %s\n", p1);
printf("After swaping the second sentance is: %s\n", p2);
return 0;
}

int main(){
//Initialize to arrays for holding the sentances, and initilize pointers
char sen1[100],*ptr1;
char sen2[100],*ptr2;
//Get sentances from user
printf("Enter first sentance: ");
scanf("%s",sen1);
printf("Enter second sentance: ");
scanf("%s",sen2);
//Set the pointers to point at the first value in the given arrays
ptr1 = &sen1[0];
ptr2 = &sen2[0];
//Run swap function
swap(ptr1,ptr2);
return 0;
}
```



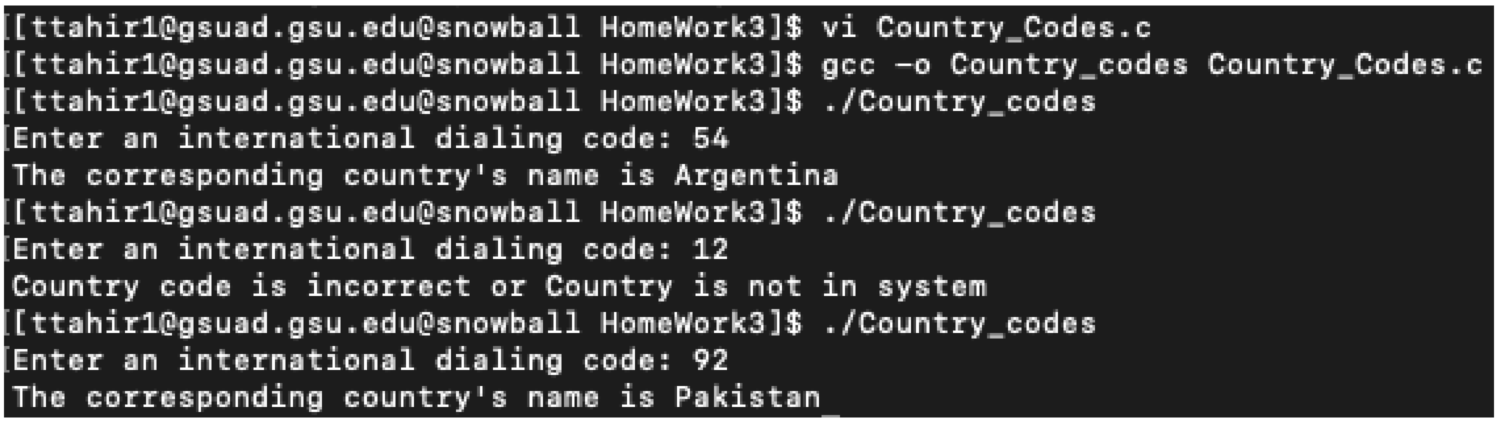
```
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ vi swap.c
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ gcc -o swap swap.c
[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ ./swap
Enter first sentance: Hello12
Enter second sentance: 12Hello
After swaping the first sentance is: 12llolo
After swaping the second sentance is: HeHel12
```

Part 3

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)



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

//Initializing an array of structures, that will be used as the database for countries
struct dialing_code{
//Initializing a pointer to point at countries
char *country;
//Initializing a variable to point at the code for each country
int code;
};

//Helper function to find the name of Country
int Country_Check(int num){
//Initialize i for a for loop and create a pointer p that points at the input
int i,*p;
p=&num;
//Initializing a structure array holding data of 20 nations
const struct dialing_code country_codes [] =
{{"Argentina",54}, {"Bangladesh", 880},
 {"Brazil", 55},{"Burma (Myanmar)", 95},
 {"China", 86}, {"Colombia", 57},
 {"Congo, Dem. Rep. of", 243}, {"Egypt", 20},
 {"Ehiopia", 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}};

//Iterate through the entire strucutre array
for(i=0;i<20;++i){
//If the country code corresponds to the input print the country name
if(country_codes[i].code==*p){
printf("The corresponding country's name is %s\n",country_codes[i].country);
//Exit if found
exit(0);
}
}
//If the code does not exit because of the for loop, country is not in Structure array
printf("Country code is incorrect or Country is not in system\n");
return 0;
}

int main(){
//Initialize variable to hold code input
int num;
//Ask for code
printf("Enter an international dialing code: ");
scanf("%d",&num);
//Run Helper function to find if country is in database
Country_Check(num);
return 0;
}
```



```
[[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ vi Country_Codes.c
[[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ gcc -o Country_codes Country_Codes.c
[[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ ./Country_codes
[Enter an international dialing code: 54
 The corresponding country's name is Argentina
[[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ ./Country_codes
[Enter an international dialing code: 12
 Country code is incorrect or Country is not in system
[[ttahir1@gsuad.gsu.edu@snowball HomeWork3]$ ./Country_codes
[Enter an international dialing code: 92
 The corresponding country's name is Pakistan_
```