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: Dang Truong

Campus ID: dtruong17

Panther #: 002420309

Part1:

1. #include<stdio.h>

int main()

{

char ch; // initiate char

printf("Enter a password : "); // display the prompt to ask for user input

int count = 0;

do{

ch = getchar(); // get user input

count++;

}

while(ch != '\n'); // while loop to check for characters inserted by user

int score = -(10 - count + 1) \* 5; // set score

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

if(score < -30) // if statement when score is less than 30

printf("The password is unsafe! Please reset."); // display when score is less than 30

else

printf("The password is safe."); // display output when score is higher than 30

return 0;

}

Text

Description automatically generated

2. #include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main() {

int score=0, uppercase\_count=0, lowercase\_count=0, numbers\_count=0; // initialize int

char ch; // initiate char

printf("Enter a password :"); // display promt asking user

while ((ch = getchar()) != '\n') { // while loop to check for user input

if(ch>='a' && ch<='z') { // if statement for lowercase letters

lowercase\_count++;

}

if(ch>='A' && ch<='Z') { // if statement for uppercase letters

uppercase\_count++;

}

if(ch>='0' && ch<='9') { // if statement for numbers

numbers\_count++;

}

}

if(lowercase\_count==0) { // add 20 if input has lowercase

score+=20;

}

if(uppercase\_count==0) { // add 20 if input has uppercase

score+=20;

}

if(numbers\_count==0) { //add 20 if input has numbers

score+=20;

}

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

if(score>30) { // if statement to check if score is lower than 30

printf("The password is unsafe! Please reset.\n"); // display result with score lower than 30

}

else {

printf("The password is safe.\n"); // display this when score is high

}

return 0;

}

Text

Description automatically generated

Part 2:

3. #include <string.h>

#include <stdio.h>

int isPalindrome(char str[]) { // initialize int parlindrome

int len = strlen(str); // string length

int left = 0;

int right = len - 1;

char ch;

while(left < right) { // while loop for left side is less than right side

ch = str[left]; // read from the left side

if(!((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))) { // checking letters from left side right

left++;

continue;

}

ch = str[right]; // read from right side

if(!((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))) { // checking letters from right side to left

right--;

continue;

}

if(str[left] != str[right]) { // if read for left and right are the same

return 0; // return palindrome

}

left++;

right--;

}

return 1;

}

int main() {

char str[1000]; // char string up to 1000 letters

printf("Enter a string: "); // display prompt asking user to enter a string

gets(str); // get string

if(isPalindrome(str)) { // check if this string is a palindrome

printf("%s is a palindrome\n", str); // display if the string is palindrome

} else {

printf("%s is not a palindrome\n", str); // display when the string is not a palindrome

}

return 0;

}Text

Description automatically generated

4.

Part3:

#include <stdio.h>

struct dialing\_code {

char \*country; // initiate char

int code; // initiate int

};

int

main (int argc, char\* argv[]) {

int intl\_code, i;

const struct dialing\_code country\_codes[] = { // construct country’s code and name

{"Argentina", 54}, {"Bangladesh", 880},

{"Brazil", 55}, {"Burma (Myanmar)", 95},

{"China", 86}, {"Colombia", 57},

{"Congo, Dem.",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},

};

int n\_entries = sizeof(country\_codes) / sizeof(\*country\_codes); // user input for country’s code

do {

int found = 0; // if no code was found

printf("Please input the international code(-1 to exit): "); // display this when no code was found and enter -1 to exit

scanf("%d", &intl\_code); // search for code input

if (intl\_code == -1) // stop the search when no code is found using -1

break; // stop execution

for (i = 0; i < n\_entries; i++) { // loop through codes and countries entered

if (country\_codes[i].code == intl\_code) { // if code was found

printf("The country is: %s\n", country\_codes[i].country); // display the country’s name

found = 1;

}

}

if (!found)

printf("Code not found.\n"); // display when code not found

} while(1);

return 0;

}Text

Description automatically generated