

# **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: Faisal Musa

Campus ID: fmusa1

Panther #:002-39-6860

**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."

The screenshot shows a web-based C IDE interface. At the top, there are browser tabs for 'Classwork for CSc 3320: System' and 'Homework4\_csc3320.docx - Google Docs'. Below the tabs is a toolbar with icons for back, forward, search, and refresh. The main workspace is titled 'main.c' and contains the following C code:

```
1 #include<string.h>
2 #include<stdio.h>
3 // checkPasswd.c
4 // main file
5 int main()
6 {
7     char ch[10];
8     int len,pnt,new_len;
9     // will prompt the user to input password
10    printf("Enter password: ");
11    scanf("%s", ch);
12    len=strlen(ch);
13    if(len!=10) // if length doesn't equal 10 then the if statement will run the code in the curly brackets
14 }
```

The status bar at the bottom left indicates 'Ln: 3, Col: 17'. Below the code editor are buttons for 'Run' and 'Share', and a 'Command Line Arguments' field which is empty. The output window below shows the program's execution:

```
Enter password:
00239686011
This password is safe :)
```

At the very bottom, a Windows taskbar shows the system tray with icons for battery, signal, and volume, along with the date and time '8:28 PM 11/14/2021'.

The screenshot shows the Online C IDE interface. The code editor displays the following C program:

```
main.c
13 {
14     new_len=10-len;
15     pnt=5*new_len;
16 {
17     if(pnt>30) // if pnt is greater than 30 then the if statement will run the code in the curly brackets
18     // but if the conditions arent true and doesnt display the values then it will run with the else statement
19     printf("Deductions : %d \nThe password is unsafe, Please reset it \n", pnt);
20 }
21 else {
22     printf("This password is safe :) \n");
23 }
24 }
25 }
26
```

The status bar at the bottom indicates "Ln: 16, Col: 6". Below the code editor is a terminal window showing the execution results:

```
▶ Run Share Command Line Arguments
Enter password:
10
Deductions : 40
The password is unsafe, Please reset it
```

The taskbar at the bottom includes icons for File, Open, Save, and Print. The system tray shows the date and time as "8:25 PM 11/14/2021".

The screenshot shows the Online C IDE interface. The code editor displays the following C program, which is identical to the one in the first screenshot but includes a return statement:

```
main.c
19     printf("Deductions : %d \nThe password is unsafe, Please reset it \n", pnt);
20 }
21 else {
22     printf("This password is safe :) \n");
23 }
24 }
25 }
26 else {
27     printf("The password is safe :) );
28 }
29 return 0;
30 }
```

The status bar at the bottom indicates "Ln: 16, Col: 6". Below the code editor is a terminal window showing the execution results:

```
▶ Run Share Command Line Arguments
Enter password:
10
Deductions : 40
The password is unsafe, Please reset it
```

The taskbar at the bottom includes icons for File, Open, Save, and Print. The system tray shows the date and time as "8:25 PM 11/14/2021".

```
main.c
19     printf("Deductions : %d \nThe password is unsafe, Please reset it \n", pnt);
20 }
21 else {
22     printf("This password is safe :)\n");
23 }
24 }
25 }
26 else {
27     printf("The password is unsafe :))");
28 }
29
30 return 0;
31 }
```

Ln: 16, Col: 6

Run Share Command Line Arguments

Enter password:  
00239686011  
This password is safe :)

\*\* Process exited due to resource limitations \*\*

2. Similar to 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.

- 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

The screenshot shows a web-based IDE interface for C programming. The code editor window displays a file named main.c with the following content:

```
main.c +  
1 #include <stdio.h>  
2 #include <string.h>  
3 int main(){  
4     char password[25]; //declare char array for storing the password  
5     int points = 100,len; //declare initially points equal 100  
6     int lower_count = 0,upper_count=0,num_count=0,consecutive_count = 0;  
7     printf("Enter the password: ");  
8     scanf("%s",password); //Tae password from the user  
9     len = strlen(password); //check the length of the password...  
10    //checks for missing lowercases in password  
11    for(int i=0;i<len;i++) {  
12        if(password[i] >= 'a' && password[i] <= 'z') {  
13            lower_count += 1;  
14        }  
15    }  
16    if(lower_count <= 0) {  
17        points -= 20; //decrease points by 20  
18    }  
19    //check if theres no capital letter  
Ln: 52, Col: 17
```

Below the code editor are three buttons: Run (highlighted in green), Share, and Command Line Arguments.

The output window shows the following terminal-like interaction:

```
Enter the password:  
waterboy  
The points for your password out of 100 is: 60  
Your password is UNSAFE for your confidential data...  
** Process exited - Return Code: 0 **
```

A small blue smiley face icon is located in the bottom right corner of the output window.

The screenshot shows a web-based C++ IDE interface. At the top, there are several tabs open in a browser window, including "Homework 3", "Google Docs", "My Drive - Google Drive", "Homework4\_csc3320.doc", "Online CPP - IDE, Code Edi", and "New Tab". Below the tabs, the address bar shows "online-cpp.com". The main content area is a code editor with the file "main.c" open. The code checks a password for points based on character counts and digit counts. It includes comments and logic for lowercase, uppercase, and numeric characters. The code editor has a toolbar with icons for file operations and a status bar showing "Ln: 52, Col: 17". Below the code editor is a toolbar with "Run" (highlighted in green), "Share", and "Command Line Arguments". The terminal output window shows the following results:

```
Enter the password:  
Water5308.  
The points for your password out of 100 is: 80  
Your password is SAFE for your confidential data...  
** Process exited - Return Code: 0 **
```

A small blue smiley face icon is located on the right side of the terminal window.

```
main.c +  
16- if(lower_count <= 0) {  
17-     points -= 20; //decrease points by 20  
18- }  
19- //check if theres no capital letter  
20- for(int i = 0;i<len;i++) {  
21-     if(password[i] >= 'A' && password[i] <= 'Z') {  
22-         upper_count += 1;  
23-     }  
24- }  
25- if(upper_count < 2) //taking the min uper case we want  
26- {  
27-     points -= 20; //decrease points by 20  
28- }  
29- //check passwrod length  
30- for(int i = 0;i<len;i++) {  
31-     if(password[i] >= '0' && password[i] <= '9') {  
32-         num_count += 1;  
33-     }  
34- }  
Ln: 52, Col: 17  
▶ Run Share Command Line Arguments  
█ Enter the password:  
█ Water5308.  
█ The points for your password out of 100 is: 80  
█ Your password is SAFE for your confidential data...  
█ ** Process exited - Return Code: 0 **
```

The screenshot shows the Online CPP IDE interface. The code editor contains a C program named main.c. The code checks a password for consecutive digits and consecutive letters, deducting points for each occurrence. It also prints the total points and a safety rating based on the points.

```
main.c +  
31- if(password[i] >= '0' && password[i] <= '9') {  
32 num_count += 1;  
33 }  
34 }  
35- if(num_count <= 0) {  
36 points -= 20; //decrease points by 20  
37 }  
38 // checks for consecutive letters in password  
39 for(int i=0;i<len;i++) {  
40- for(int j = i+1;j<len;j++) {  
41 //checks the next character  
42- if(password[j] - password[i] == 1) {  
43 consecutive_count += 1;  
44 }  
45 }  
46 }  
47 if(consecutive_count >= 2) //runs through if statement if more than 2 characters are consecutive  
48- {  
49 points -= 20; //decrease points by 20  
50 }  
Ln: 52, Col: 17
```

Run | Share | Command Line Arguments

Enter the password:  
hi  
The points for your password out of 100 is: 60  
Your password is UNSAFE for your confidential data...  
\*\* Process exited - Return Code: 0 \*\*

The screenshot shows the Online CPP IDE interface. The code editor contains a modified C program named main.c. This version includes printf statements to output the user's password and its safety rating directly to the terminal.

```
main.c +  
43 consecutive_count += 1;  
44 }  
45 }  
46 }  
47 if(consecutive_count >= 2) //runs through if statement if more than 2 characters are consecutive  
48- {  
49 points -= 20; //decrease points by 20...  
50 }  
51 //print details required for user  
52- if(points < 70) {  
53 printf("The points for your password out of 100 is: %d",points);  
54 printf("\nYour password is UNSAFE for your confidential data...");  
55 }  
56- else {  
57 printf("The points for your password out of 100 is: %d",points);  
58 printf("\nYour password is SAFE for your confidential data...");  
59 }  
60 return 0;  
61 }  
Ln: 52, Col: 17
```

Run | Share | Command Line Arguments

Enter the password:  
WATERisgod1234#  
The points for your password out of 100 is: 80  
Your password is SAFE for your confidential data...  
\*\* Process exited - Return Code: 0 \*\*

## 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).

The screenshot shows the online-cpp.com BETA IDE interface. The code editor window displays the following C program:

```
main.c
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdbool.h>
4
5 bool is_palindrome(char string[]);
6
7 int main(void)
8 {
9     // example strings
10    char string1[] = "not a palindrome!";
11    char string2[] = "123";
12    char string3[] = "abcdcba";
13
14    // check if each string is a palindrome and report the results
15    if (is_palindrome(string1)) printf("%s\n- is a palindrome\n", string1);
16    else printf("%s\n- is not a palindrome\n", string1);
17    printf("\n");
18
19    if (is_palindrome(string2)) printf("%s\n- is a palindrome\n", string2);
L: 1, Col: 1
```

The output window below shows the results of running the program:

```
123
- is not a palindrome

abcdcba
- is a palindrome

** Process exited - Return Code: 0 **
```

A feedback pop-up in the bottom right corner says: "Help us improve by sharing your feedback." with a smiley face icon.

online-cpp.com

ONLINE CPP BETA

```
main.c +  
15 if (is_palindrome(string1)) printf("%s\n- is a palindrome\n", string1);  
16 else printf("%s\n- is not a palindrome\n", string1);  
17 printf("\n");  
18  
19 if (is_palindrome(string2)) printf("%s\n- is a palindrome\n", string2);  
20 else printf("%s\n- is not a palindrome\n", string2);  
21 printf("\n");  
22  
23 if (is_palindrome(string3)) printf("%s\n- is a palindrome\n", string3);  
24 else printf("%s\n- is not a palindrome\n", string3);  
25  
26 return 0;  
27 }  
28  
29 // returns true if the string is a palindrome and false otherwise  
30 bool is_palindrome(char string[]){  
31 {  
32 // find the middle of the string as we'll need to check up until here  
33 // ...  
Ln: 1, Col: 1
```

Run Share Command Line Arguments

```
123  
- is not a palindrome  
abcdcba  
- is a palindrome  
  
** Process exited - Return Code: 0 **
```

Help us improve by sharing your feedback.

The screenshot shows the online-cpp.com BETA interface. The code editor window contains the following C code:

```
main.c
27
28 // returns true if the string is a palindrome and false otherwise
29 bool isPalindrome(char string[])
30 {
31     // find the middle of the string as we'll need to check up until here
32     int len = strlen(string);
33     int middle = len / 2;
34
35     // check the corresponding characters on the left and right sides of the
36     // string to see if they match, until we reach the middle, if they ever do
37     // not match then return false as the string is not a palindrome
38     for (int i = 0; i < middle; i++)
39     {
40         if (string[i] != string[len - i - 1]) return false;
41     }
42     // if all corresponding characters on the left and right sides of the string
43     // have matched, the string must be a palindrome and so return true
44     return true;
45 }
```

The output window shows the results of running the program with two inputs:

- Input: 123
- Output: - is not a palindrome
- Input: abcdcba
- Output: - is a palindrome

At the bottom, it says: \*\* Process exited - Return Code: 0 \*\*

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.

online-cpp.com

ONLINE CPP BETA

```
main.c +  
1 #include <stdio.h>  
2 #include <stdlib.h>  
3 #include <string.h>  
4  
5 void swap(char* s1, char* s2);  
6 int main() {  
7     size_t DS = 100; // DS is default size  
8     printf("Enter 1 sentence: \n");  
9     char* sent1= (char*)malloc(DS * sizeof(char)); // sent is sentence  
10    getline(&sent1, &DS, stdin);  
11    printf("Enter another sentence of equal length: \n");  
12    char* sent2= (char*)malloc(DS * sizeof(char));  
13    getline(&sent2, &DS, stdin);  
14    //here we will strip the new line of characters from both sentences  
15    sent1[strlen(sent1) - 1] = 0;  
16    sent2[strlen(sent2) - 1] = 0;  
17    // here is what the 2 messages will be after we swap the sentences  
18    printf("sentence1 after: %s\n", sent1);  
19    printf("sentence2 after: %s\n", sent2);  
Ln: 10, Col: 33
```

Run Share Command Line Arguments

```
Enter 1 sentence:  
the mushroom walks  
Enter another sentence of equal length:  
the mushroom falls  
sentence1 after: the mushroom walks  
sentence2 after: the mushroom falls  
  
** Process exited - Return Code: 0 **
```

Help us improve by sharing your feedback.

online-cpp.com

ONLINE CPP

```
main.c +  
16     sent2[strlen(sent2) - 1] = 0;  
17     // here is what the 2 messages will be after we swap the sentences  
18     printf("sentence1 after: %s\n", sent1);  
19     printf("sentence2 after: %s\n", sent2);  
20     return 0;  
21 }  
22 void swap(char* s1, char* s2) {  
23     //if there is an error in the sentences and not equal then run through for loop  
24     if (strlen(s1) != strlen(s2)) {  
25         printf("Sentences are not equal length. Cant Swap \n");  
26         return;  
27     }  
28     int length = strlen(s1);  
29     for (int i = 0; i < length; ++i) {  
30         s1[i] = s1[i] ^ s2[i];//here we are swapping the variables  
31         s2[i] = s1[i] ^ s2[i];  
32         s1[i] = s1[i] ^ s2[i];  
33     }  
34 }  
Ln: 10, Col: 33
```

Run Share Command Line Arguments

```
Enter 1 sentence:  
the mushroom walks  
Enter another sentence of equal length:  
the mushroom falls  
sentence1 after: the mushroom walks  
sentence2 after: the mushroom falls  
  
** Process exited - Return Code: 0 **
```

Help us improve by sharing your feedback.

### 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)

The screenshot shows a browser window with multiple tabs open. The active tab is 'Online CPP - IDE' at [online-cpp.com](https://online-cpp.com). The code editor contains a C program named 'main.c'. The code defines a struct 'dialing\_code' with variables for a country name and its international code. It then defines a const struct 'country\_codes' containing 20 entries of country names and their codes. The program prompts the user to input an international code, searches for it in the array, and prints the corresponding country name. The terminal window below shows the execution of the program, where the user inputs '243' and the program outputs 'The country is: Congo,Dem.'.

```
main.c +  
1 #include <stdio.h>  
2 struct dialing_code {  
3     // setting the variables  
4     char *country;  
5     int code;  
6 };  
7 int main (int argc, char* argv[]) {  
8     int intco, i;  
9     //countries that you can input the international code for down below  
10    const struct dialing_code country_codes[] = {  
11        {"Antigua and Barbuda", 268}, {"Bangladesh", 880},  
12        {"Burundi", 257}, {"Burma (Myanmar)", 95},  
13        {"China", 86}, {"Cayman Islands", 345},  
14        {"Congo,Dem.", 243}, {"Egypt", 20},  
15        {"Ethiopia", 251}, {"France", 33},  
16        {"Grenada", 473}, {"Greece", 30},  
17        {"Indonesia", 62}, {"Iran", 98},  
18        {"Italy", 39}, {"Japan", 81},  
19        {"Malawi". 263}, {"Nigeria". 234}.  
Ln: 7, Col: 5  
Run Share Command Line Arguments  
Please input the international code(-1 to exit):  
243  
The country is: Congo,Dem.  
Please input the international code(-1 to exit):  
-1  
  
** Process exited - Return Code: 0 **
```

Homework 3 | Google Docs | My Drive - Google | Homework4\_csc | New Tab | Online CPP - IDE | Check If A String | Operators in C | Reading List

online-cpp.com

ONLINE CPP BETA G f + 104

main.c

```
19     {"Congo",      243, {"Congo", "Dem.", 243},  
20     {"Malawi",     263}, {"Nigeria",      234},  
21     {"Pakistan",    92}, {"Philippines",   63},  
22     {"Poland",       48}, {"South Sudan", 211}  
23 };  
24 //setting entry1 as a variable where it is divided by the country codes  
25 int entry1 = sizeof(country_codes) / sizeof(*country_codes);  
26  
27 do {  
28     int found = 0;  
29  
30     printf("Please input the international code(-1 to exit): ");  
31     scanf("%d", &intco);  
32     if (intco == -1)  
33         break;  
34  
35     for (i = 0; i < entry1; i++) {  
36         if (country_codes[i].code == intco) {  
             printf("The country is: %s\n", country_codes[i].country);  
Ln: 7, Col: 5
```

Run Share Command Line Arguments

```
Please input the international code(-1 to exit):  
243  
The country is: Congo,Dem.  
Please input the international code(-1 to exit):  
-1  
  
** Process exited - Return Code: 0 **
```

Smiley face icon

The screenshot shows a browser window with multiple tabs open, including "Homework 3", "Google Docs", "My Drive - Google", "Homework4\_csc", "New Tab", "Online CPP - IDE", "Check If A String", "Operators in C", and "Reading List". The main content is the "ONLINE CPP BETA" IDE interface.

The code in the editor is:

```
main.c +  
29     printf("Please input the international code(-1 to exit): ");  
30     scanf("%d", &intco);  
31     if (intco == -1)  
32         break;  
33  
34     for (i = 0; i < entry1; i++) {  
35         if (country_codes[i].code == intco) {  
36             printf("The country is: %s\n", country_codes[i].country);  
37             found = 1;  
38         }  
39     }  
40     if (!found) //if it is NOT found then the for loop will run through the print statement  
41         printf("Code not found.\n");  
42     } while(1);  
43  
44     return 0;  
45 }  
46  
47
```

The status bar at the bottom left says "Ln: 7, Col: 5".

The output window shows the following interaction:

```
Please input the international code(-1 to exit):  
243  
The country is: Congo,Dem.  
Please input the international code(-1 to exit):  
-1
```

At the bottom right of the output window is a blue smiley face icon.