SYST 44288 – Operating Systems & Systems Programming

Assignment 1 – 6.5%

Due Date: See SLATE.

You can work in teams of two.

C Programming Review: Taking in command line arguments

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

argv is an array of pointers of type char (i.e. array of strings). This array holds all of the arguments passed when executing the program. If your program is named 1.1.c, after it has successfully compiled you will execute the program by calling its name and then passing all necessary command line arguments. Ex. ./1.1 3 0 600000

argc is the length of this array.
argv[0] = 1.1
argv[1] = 3
argv[2] = 0
argv[3] = 600000

Complete the following programs in the C programming language.

1. (2 points) Armstrong Numbers
Number n is called Armstrong's number of order k if it is equal to the sum of k-th powers of its digits.

For example: 371 = 3^3 + 7^3 + 1^3 Armstrong's number of order 3. Refer to the Armstrong Numbers research paper provided for details on how Armstrong numbers work.

Write C program which prints all the Armstrong's numbers of a given order k, within the given interval between p and q. You should use functions like pow() provided by the math library.

Input: The user will type the integers k, p, and q on the command line. Taking in one integer at a time, **in this order**.

Output should consist of a single number per line with no extra or blank lines.

Test Input (k, p, q)	Test Output
3 0 600000	153 370 371 407
4 0 600000	1634 8208 9474
5 0 600000	54748 92727 93084
6 0 600000	548834

2. (2 points) Let n be a positive integer. Euler's phi-function φ(n) is defined to be the number of positive integers not exceeding n that are *relatively prime* to n.

Write a program that can be used to calculate and display the value of Euler's phi-function for a series of positive integer, separated by spaces, entered on the command line.

3. (2 points) Using pointers and *avoiding unnecessary local variables*, write a function, *rmchr* that takes a string and a character as arguments. The *rmchr* function should remove **all** occurrences of the character from the string. The *rmchr* function should not leave spaces characters in the string where characters have been removed. The resulting string should be returned using an appropriate data type and printed to the command line.

4. (2 points) Using pointers and *avoiding unnecessary local variables*, write a function, *rmstr* that takes two strings as arguments, removing all occurrences of any character from the second string in the first string. The *rmstr* function should not leave spaces characters in the string where characters have been removed. The resulting string should be returned using an appropriate data type and printed to the command line.

Correctness Rubric

- Does not compile – zero
- Compiles and 100% of output is incorrect – zero
- Marks will be lost for each requirement not completed
- Marks will be lost for each incorrect output

- All requirements stipulated in the assignment are met AND 100% of the output is correct – 100%
- Each question is worth 2 marks.

Submission Requirements Rubric

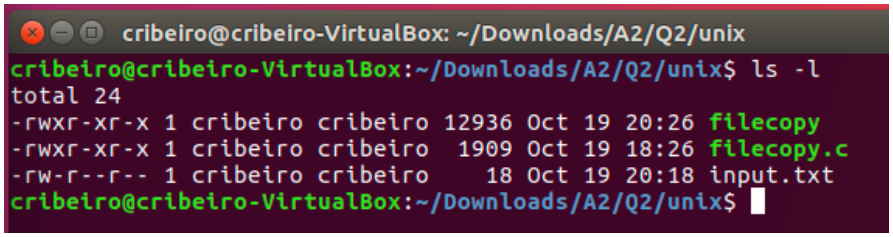
- Commenting where necessary (1 mark)
- Submit the source code not just the compiled code (1 mark)
- Submitted correct file names as outlined (1 mark)
- Submitted all required files as outlined (1 mark)
- Submitted .zip file without subfolders & named correctly as outlined (1 mark)
- Submitted to correct dropbox folder on SLATE (1 mark)

TOTAL: 14 marks

Submission Requirements

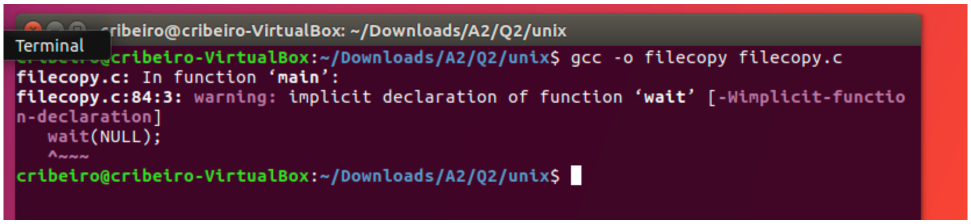
1. **Screenshots for EACH Question:**

A. Must show the command “ls –l” entered on the command line and output of this command.
Name this screenshot Q#_ls.png OR .jpg, where # is the question number.
SAMPLE:



```
cribeiro@cribeiro-VirtualBox: ~/Downloads/A2/Q2/unix
cribeiro@cribeiro-VirtualBox:~/Downloads/A2/Q2/unix$ ls -l
total 24
-rwxr-xr-x 1 cribeiro cribeiro 12936 Oct 19 20:26 filecopy
-rwxr-xr-x 1 cribeiro cribeiro  1909 Oct 19 18:26 filecopy.c
-rw-r--r-- 1 cribeiro cribeiro    18 Oct 19 20:18 input.txt
cribeiro@cribeiro-VirtualBox:~/Downloads/A2/Q2/unix$ 
```

B. Must show compiled without errors, warnings are acceptable.
Name this screenshot Q#_compile.png OR .jpg, where # is the question number.
SAMPLE:



```
cribeiro@cribeiro-VirtualBox: ~/Downloads/A2/Q2/unix
cribeiro@cribeiro-VirtualBox:~/Downloads/A2/Q2/unix$ gcc -o filecopy filecopy.c
filecopy.c: In function ‘main’:
filecopy.c:84:3: warning: implicit declaration of function ‘wait’ [-Wimplicit-functio
n-declaration]
   wait(NULL);
   ^~~~
cribeiro@cribeiro-VirtualBox:~/Downloads/A2/Q2/unix$ 
```

C. Must show execution and correct output
Name this screenshot Q#_execute.png OR .jpg, where # is the question number

2. **Create a file folder named Firstname1_Lastname1_Firstname2_Lastname2_A1:**

   - Zip the file not any other format of archiving files
   - Do **NOT** include any subfolders in the zip file
   - If you’re submitting readme files please make them .txt not .doc