**CSC 406: Systems I**

**Homework 1**

**Due by 5:45 pm on Thursday, September 15**

Please write precise and concise answers.

**Reading**

Read the week 2 [assigned readings](https://d2l.depaul.edu/d2l/le/news/896648/966560/view) described in the syllabus 

**Assignment**

First, subscribe to the [Discord server](https://discord.gg/Hj4XtGTsAW) I set up for the class.  
  
Log into cdmcscspprd01.dpu.depaul.edu (as shown in the lecture notes) and copy zipped file hw1.zip from /home/sparsa2/public/ to the directory of your choice. For example:

$ cp /home/sparsa2/public/hw1.zip .

(Note: Do not type the $ symbol; it represents the command line prompt.) The file hw1.zip will be in your current directory as you can verify using the ls command:

$ ls  
hw1.zip

(The output of the command is shown in gray.) Now unzip the compressed file using the unzipcommand:

$ unzip hw1.zip   
Archive:  hw1.zip  
   creating: hw1/  
  inflating: hw1/Makefile              
  inflating: hw1/hw1.c

and move to newly created directory hw1 using the cd command

$ cd hw1/

In directory hw1 you will find C file hw1.c that you will work on and file Makefile which contains the UNIX shell command that will submit your completed homework to the dropbox on the Linux box. You can verify this:

$ ls  
hw1.c  Makefile

Open file hw1.c for editing and do the homework using the instructions below:

$ emacs hw1.c

When done, save your file, exit emacs, and submit the homework as follows:

$ make submit  
cp -i hw1.c /home/sparsa2/dropbox/hw1-`whoami`.c

Notes:

1. To submit your homework, just type make submit and nothing else.
2. DO NOT MODIFY THE NAME OF THE FILE hw1.c. The command make submit will submit to the dropbox a file called hw1.c and no other!
3. If you see the output cp -i hw1.c /home/lperko/dropbox/hw1-`whoami`.c then your submission was successful.
4. You may submit multiple times but each submission will overwrite the previous one; an interactive warning will ask you whether you want to do the overwrite, which you can use to convince yourself that the previous submission was successful.

**Problems**

1.    Implement function ranges that takes no input and prints the ranges (i.e., the minimum and maximum legal values) of integer types char, short, int, long, and long long, both signedand unsigned. You should do this by printing appropriate constants defined in header file <limits.h> located in directory /usr/include. Your output should be similar to:

signed char  
minimum value: -128  
maximum value: 127  
  
unsigned char  
minimum value: 0  
maximum value: 255  
  
signed short   
minimum value: -32768  
maximum value: 32767  
  
... (and so on)

Note: The conversion instructions for signed and unsigned long are li and lu, respectively. The conversion instructions for signed and unsigned long long are lli and llu, respectively. The minimum value for unsigned integers is always 0 and therefore not defined in header file <limits.h>.  
  
  
2. Implement four versions of a function funSum that takes a positive integer n as input and returns the sum of all integers up to and including n that are divisible by 6 or 7:  using a for loop in funSum1, using a while loop in funSum2, using a do-while loop in funSum3, and using recursion in funSum4. Your output for the included test code should be:

funSum1(20) = 57  
funSum2(20) = 57  
funSum3(20) = 57  
funSum4(20) = 57

3.    Implement function types that takes no input, declares 3 variables of type char, 3 of type short, 3 of type int, and 3 of type double---in that order---and prints the addresses of the 12 variables---in the same order---in both hex (use %p conversion instruction) and unsigned long format.

&a1 = 0x7ffd45e3ac0f, 140725776002063  
&a2 = 0x7ffd45e3ac0e, 140725776002062  
&a3 = 0x7ffd45e3ac0d, 140725776002061  
&b1 = 0x7ffd45e3ac0a, 140725776002058  
&b2 = 0x7ffd45e3ac08, 140725776002056  
...