

## Homework #4

**Due: November 18, 2015**

**Motivation:** Gain familiarity with stack frames and how arguments are passed in “C” and x86-64.

**Overview:** We are providing a file called **hw4main.c** which contains functions **main** and **foo**. This code calls a function **recover**, which you will create. You’ll submit your **recover** function in a separate file called **hw4.c**.

When you examine **hw4main.c**, you’ll see functions **main** and **foo**. The function **foo** is called recursively several times after doing some sort of a funny computation to mix up the numbers. The exact number of times that **foo** recurses is data dependent. After recursing several levels down, **foo** will call the **recover** function, which you will write. Your function will do some printing and then return, after which all the calls to **foo** will complete and return.

Function **foo** is passed several values.

Your task is to write the function **recover**, which will print the arguments to **foo**, which are named “a”, “b”, and “c”. Your function should print the arguments to all invocations of **foo** and return.

Unfortunately **recover** is passed no arguments and your code will be tested using our version of **hw4main.c**. How can **recover** know what was passed to **foo**?

**Restriction:** You are free to create and include in **hw4.c** additional helper functions which **recover** may invoke. However, do not write or use any assembly code, even within your **.c** file.

Function **recover** and any other functions you write may only call **putchar**. You may also call **printf** to print strings like “foo called with args\n” and “ a: ”, but don’t use it to print the hex numbers. You may not invoke any other library routines.

Your code must print the arguments in exactly the following format and must print the arguments in hexadecimal.

Here is some sample output for this program. Your code should print the arguments in the order the functions were invoked. In other words, the first set of arguments to be printed is from the initial call that **main** makes.

```
% hw4 11223344 12345 20 5
main called with a = 0xab4130, b = 0x3039, c = 0x14, countDown = 5
foo called with args
  a: 0000000000ab4130
  b: 0000000000003039
  c: 0000000000000014
foo called with args
```

## Homework #4

```
a: 000000000156b259
b: 0000000000006066
c: 0000000000ab4118
foo called with args
a: 0000000002ad04d4
b: 0000000000ab81d4
c: 0000000000003069
foo called with args
a: 0000000005f1887c
b: 00000000015733c1
c: 0000000002ad6406
foo called with args
a: 000000000ab42339
b: 0000000000030384
c: 0000000000ab4070
foo returns 0x30ca49e4
```

The file **hw4main.c** is at `cs.pdx.edu/~harry/cs201/hw4main.c`.

Submit a single file named **hw4.c**. It will be compiled and tested on the PSU Linux machines with the following commands. When tested, we will of course call it with a different set of values than this example, to make sure it recovers the arguments correctly.

```
gcc -Wall -O1 -m64 -S hw4main.c
gcc -Wall -O1 -m64 -S hw4.c
gcc -m64 hw4main.s hw4.s -o hw4
```

*This is an individual project, not a group project. You must work alone to create and write all code you submit. You are not to share code or debug other students' code or look at other people's code. Copying code from the web or anywhere else is totally cheating.*

**Submission:** Send a single email to our grader. Include as a single attachment, the file **hw4.c** with

**Subject:** CS201, HW4, <your name>

**Grading:** It is assumed that your program will produce the correct output. Therefore, your grade will be based on its simplicity, clarity, and efficiency. "Short and sweet" is better than "long and confusing."