Ontario Tech UNIVERSITY

Faculty of Science

Course:	CSCI 2050U: Computer Architecture I
Date:	April 19th, 2021
Component:	Final Lab Exam

Overview

In this lab exam, you will create a program that loops through an array of 20 integers, copies all of the elements from that array into another array if they are divisible by some user-entered number, and returns the count of all items copied into the destination array.

***Note**: You will add all of your code to the file called `exam_tosubmit.asm`, which will mean that the provided `Makefile` will correctly build your project with debug information.*

Instructions

This lab exam involves the creation of a function, `getDivisible`, and a main function which includes code to test that function.

Copying Numbers

The `getDivisible` function will take four arguments, which will be passed using registers according to the convention used in the lectures (System V ABI):

1. `rdi` – the (destination) array to which the values are to be copied
2. `rsi` – the (source) array from which the values are to be copied
3. `rdx` – the number of elements in the source array
4. `rcx` – the divisor

Only values that are evenly divisible by the divisor will be copied over to the destination array. You will also return the number of values copied, which could be anywhere between 0 and `numValues`.

***Note**: Be sure to go to the next number in the array by adding the correct size (in bytes) to each of your memory registers (e.g. `rsi` and `rdi`).*

***Hint**: Using the IDIV instruction is a good way to test divisibility. The in-class example `evenodd` does something similar.*

Test Code

In the `main` function, you will ask the user to enter a single integer, `divisor`. First, display the input prompt (`inputPrmpt`), then read in an integer using `scanf` and the input format string (`inputFmt`).

You will then pass the source array (`startValues`), the destination (empty) array (`endValues`), the `numValues` (provided in the data section), and `divisor`, according to the ordering provided, above.

Finally, you will print the numbers in the destination array one at a time, by looping over the destination array (`endValues`), using the output format string (`outputFmt`) and `printf`.

***Note**: Do not print all of the numbers in the array according to its capacity, only print the count of numbers that was copied (i.e. the return value of the `getDivisible` function).*

***Note**: Make sure that you use the variables (and their values) declared in the provided code.*

Sample Output

Below, you will find some example inputs and outputs:

```
$ ./main.out
Enter a number: 2
70 is divisible by 2
-16 is divisible by 2
0 is divisible by 2
-72 is divisible by 2
30 is divisible by 2
-28 is divisible by 2
6 is divisible by 2
-2 is divisible by 2
4 is divisible by 2

$ ./main.out
Enter a number: 3
-1 is divisible by 3
-16 is divisible by 3
0 is divisible by 3
-7 is divisible by 3
15 is divisible by 3
30 is divisible by 3
-28 is divisible by 3
6 is divisible by 3
21 is divisible by 3
33 is divisible by 3

$ ./main.out
Enter a number: 5
70 is divisible by 5
-1 is divisible by 5
-16 is divisible by 5
0 is divisible by 5
15 is divisible by 5
30 is divisible by 5
```

How to Submit

Ensure that your assembly language file is called `exam_tosubmit.asm`, and submit only this file as your response. Do not use any unusual libraries (aside from libc) or external code. All of the code for this lab exam must be in your (single) .asm file.

You are not creating a zip of the entire folder, but rather are only submitting the `exam_tosubmit.asm` file. This is the only file type that Canvas will accept.

Note: If you submit the wrong file, it will not be accepted once the exam period has ended. Take extra care and name your files appropriately.

Note: The instructor will not be able to verify which file you have submitted. There are simply too many students for that to happen. Canvas shows your file before you submit your exam. Verify that this is the correct file before you click Submit.