Lab 4 - Registers

Dr. Donald Davendra
CS311 - Computer Architecture 1

October 18, 2016

The third laboratory exercise requires you to assign the contents of an array in **yasm** and calculate the **sum** and **difference** of the elements in the array.

Create a file named `array.asm` in ebe.

Question 1 - `.data` section.

You are required to assign an array in the `.data` segment as the following:

- label - `a`
- size - `dw` - 2 bytes
- contents - 112, 67, 121

The `segment .data` is given as:

```
      segment .data
a     dw   112, 67, 121       ; array of 3 values
sum   dq   0                  ; memory to store the sum
diff  dq   0                  ; memory to store the difference
```

You can declare other variables as you deem necessary to solve this lab.

Question 2 - `.text` section.

The text segment is empty.

```
      segment .text
      global  main
main:
```

Question 3 - `global main` section.

The task in the main section is to iteratively add and subtract the values in array `a` and store the results in:

- `sum` memory field for the sum of all numbers
- `diff` memory field for the difference of all numbers

You are allowed to use a maximum of **three general purpose registers** in this lab. Some of the commands of use in this lab are:

- `mov` - moving data from register-register, register-variable etc
- `lea` - loading effective address of a variable to a register.
- `add` - adding two values in registers or in variables
- `sub` - subtracting two values in registers or in variables

Upon completion of the task, zero out the `rax` register and return. This is given as:

```
...                     ; your code
xor     rax, rax        ; zero out rax
ret
```

Submission

The file must be submitted through Canvas by 5pm October 28, 2016. The penalty for late submission is 10% for 1 day, 20% for 2 day, after which it will be zero. The grading rubric is given in Table 1.

Table 1: Grading rubric

File	Aspects	Points
`array.asm`	Compiles	5
	Correct result	30
	Correct use of registers	25
	Correct use of memory offsets/addressing	25
	Documentation	15