Ean Dodge

Lab1

Coding Project 1

Report

Code:

```
.text
add $s3, $s3, $zero
loop:
#getting F
li $v0, 4 #print Question for F
la $a0, fprint # arguments
syscall

li $v0, 5 #get f
la $a0, finput #arguments
li $a1, 6
syscall
move $s0, $v0

#getting G
li $v0, 4 #print Question for G
la $a0, gprint #argument
syscall

li $v0, 5 #get g
la $a0, ginput #arguments
li $a1, 6
syscall
```

```
add $s1, $v0, $zero


                        #f = g - (f + 5)

addi $t0, $s0, 5 #f+5

sub $t1, $s1, $t0 #g-(f+5)

li $v0, 4 #print result message

la $a0, result #arguments

syscall


add $a0, $t1, $zero #move result value into argument register

li $v0, 1 #print result value

syscall

addi $s3, $s3, 1 #ran one more time

bne $s3, 3, loop #if it has not run 3 times, start over


exit: #does nothing

.data #where to put data(messages and inputs)


fprint: .asciiz "\nWhat is F?\n"

gprint: .asciiz "What is G?\n"

finput: .space 10

ginput: .space 10

result: .asciiz "Your result to f= g-(f+5): "
```

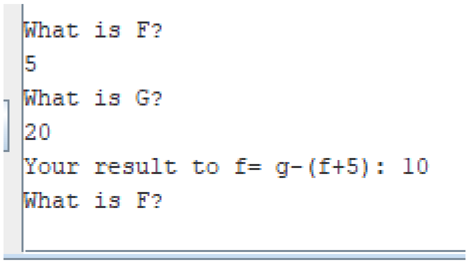
Summary:

I started with my data, I made my messages for F, G, and my result. Then I made my inputs for the user to put in the numbers they want. Then I would start with my text. I started by getting an F from the user. I used li command with $v0 to indicate what function I am doing. With the number 4, I will be printing a string. It takes the arguments through $a0, which took fprint as a string to output. Then I called a system call. I then do the same thing with $v0, but this time the number is 5 to indicate that I am going to input an integer. I use finput as my first argument, then the second argument is how much space for the user to input. I put 6 to be on the safe side, but this number can be smaller, or bigger. Next, I call a system call., Then I move that number into a register so that I can use it later in my equation. I do the same exact thing after this to get a number for g.

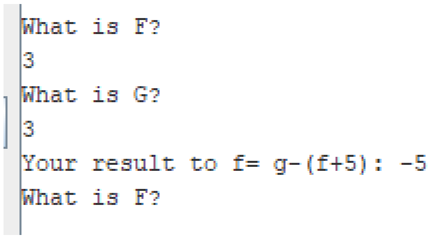
To do the actual equation, I first start by adding the user input f and adding it to a constant 5. I put this result into a temporary register. I then subtracted the user input g by the value in the temporary register. That would give me my result. I printed the result message, and called a system call for that. Then I moved the result number into an argument register to use it in a print function. I called a system call to run this print.

My loop starts at the beginning of the program where it says loop. This puts all my code into the loop. At the end, I use a number to index the number of times the code has been run. Once it has run three times, it will jump to exit and end normally.

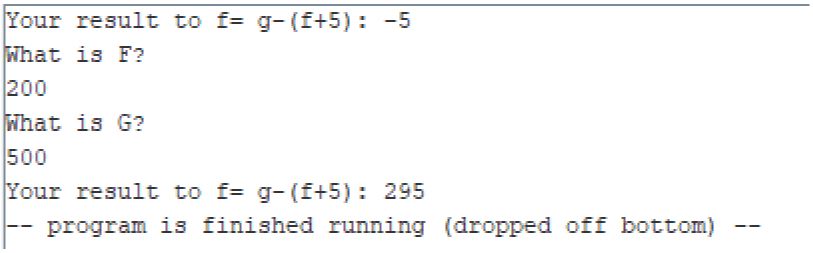
Screen shots:



```
What is F?
5
What is G?
20
Your result to f= g-(f+5): 10
What is F?
```



```
What is F?
3
What is G?
3
Your result to f= g-(f+5): -5
What is F?
```



```
Your result to f= g-(f+5): -5
What is F?
200
What is G?
500
Your result to f= g-(f+5): 295
-- program is finished running (dropped off bottom) --
```

Conclusion:

I have learned the basics of an assembler language. I can now read and write in this language. I have also learned that a computer reads an equation the same way that we do. The parenthesis first then moves on. My problems faced were figuring out the loop part, but I used my experience to figure it out quickly.