

1.0 Program 1 Inputs: Two positive numbers a and b stored in \$s1, and \$s2
Program 1 Outputs: Five positive numbers from loop

Program 2 Inputs: No input
Program 2 Outputs: No Output

Program 3 Inputs: Two positive numbers a and b stored in \$s0, \$s1
Program 3 Outputs: Positive number F

2.0 Program 1 is simple first a message is displayed to the users prompting them to enter 2 positive integers. These two numbers are each entered on a line and stored to \$s1 and \$s2 respectively. A loop is then executed with an increment value stored in \$s0. Each instance of the loop calculates an output F based off the provided formula. This value is displayed, and the loop is incremented. This loop runs five times total and then exits.

Program 2 starts by allocating space for the array. A syscall is made to notify the user the loop has started. The index of the loop is stored in \$s0 also serves in calculating the value of each index in the array. The value is calculated for the array index and stored. The array index and loop counter are then incremented. This continues until the loop reaches termination, the loop jumps to exit and prints the exit line.

Program3 prompts the user for 2 numbers, these numbers are saved in \$s0, and \$s1. I then used the set on less then command to store either a 1 or 0 in \$s4 this command checks the second condition of the if statement, $b < 5$. I then used a similar approach to check the first condition of the if statement and stored the result in \$s3. This gave me more trouble then the first one but I used a clever work around, that $a \geq 5 == 4 < a$. Using the binary values in \$s3 and \$s4 I checked if they were equal with the bne command. If they were I would add a and b, \$s0 and \$s1, and print out the answer, note there is an exit jump here to avoid running the else segment also. If they were not equal I jumped to an else label that would multiply the two variables. NOTE: 1 is true and 0 is false in my code

3.0

Register	Purpose
Program1	
\$v0	Syscall
\$a0	Info to be printed out
\$s1	Store a
\$s2	Store b
\$s0	Store I
\$t0	Store A*I
Program 2	
\$v0	Syscall
\$a0	Info to print out
\$s0	Store I
\$t0	Array address
\$t1	Array index

Program 3

\$v0

\$a0

\$s0

\$s1

\$s2

\$s4

\$s3

Syscall

Info to print out

Store a

Store b

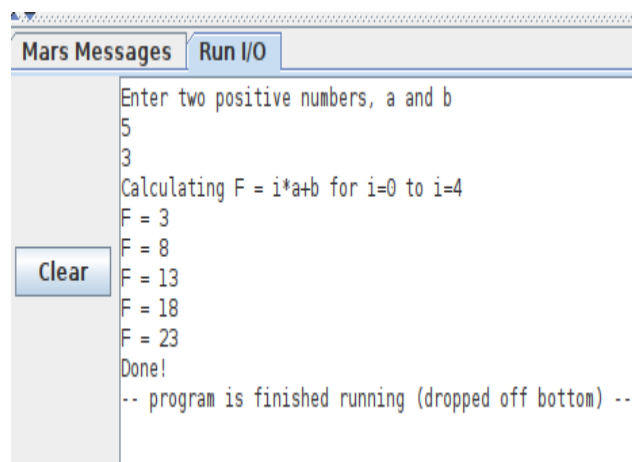
Store f

Store 1 if B<5, else 0

Store 1 if a>=4, else 0

- 4.0 During this assignment I learned
- How to build if statements
 - How to build for loops
 - How to build while loops
 - How to build Arrays and index into specific places
 - How to create and use labels

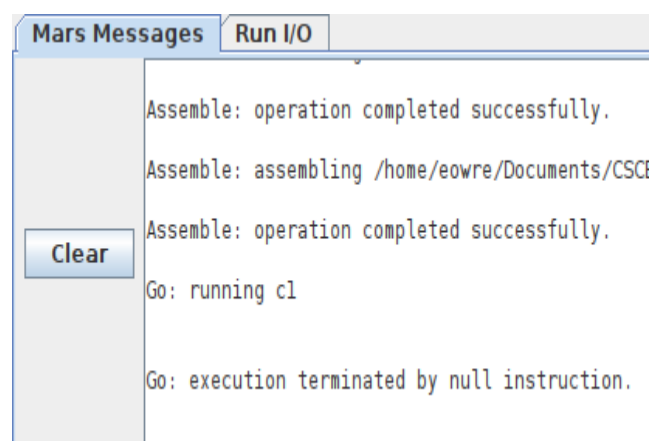
5.0 Program 1



Mars Messages Run I/O

```
Enter two positive numbers, a and b
5
3
Calculating F = i*a+b for i=0 to i=4
F = 3
F = 8
F = 13
F = 18
F = 23
Done!
-- program is finished running (dropped off bottom) --
```

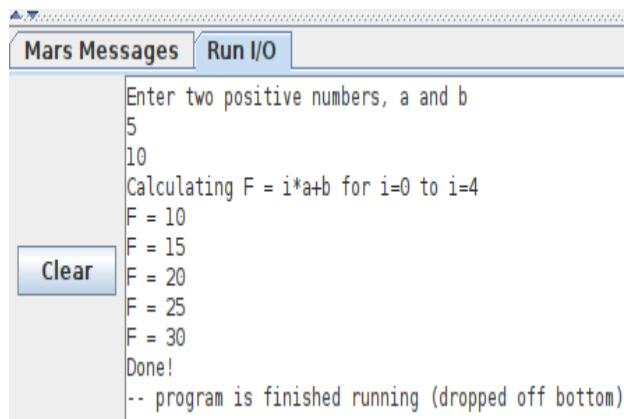
Clear



Mars Messages Run I/O

```
Assemble: operation completed successfully.
Assemble: assembling /home/eowre/Documents/CSCI
Assemble: operation completed successfully.
Go: running c1
Go: execution terminated by null instruction.
```

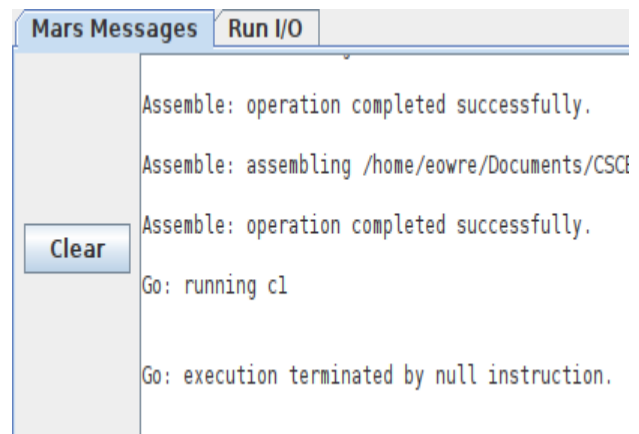
Clear



Mars Messages Run I/O

```
Enter two positive numbers, a and b
5
10
Calculating F = i*a+b for i=0 to i=4
F = 10
F = 15
F = 20
F = 25
F = 30
Done!
-- program is finished running (dropped off bottom)
```

Clear



Mars Messages Run I/O

```
Assemble: operation completed successfully.
Assemble: assembling /home/eowre/Documents/CSCI
Assemble: operation completed successfully.
Go: running c1
Go: execution terminated by null instruction.
```

Clear

Program 2

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	p o o L	a t s	! s t r	o L \0 \n	e p o	i s d n	\0 \n \0 \n	\0 \0 \0 .
0x10010020	\0 \0 \0 \0	\0 \0 \0 .	\0 \0 \0 \0	. . .	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010040	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100e0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010100	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010120	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010140	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010160	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010180	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100101a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100101c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

Program 3

Mars Messages

Run I/O

Enter two positive numbers, a and b

Clear

Enter two positive numbers, a and b
5
3
2
-- program is finished running (dropped off bottom)

Mars Messages

Run I/O

Assemble: assembling /home/eowre/Documents/CSCE212/hw2/c:

Assemble: operation completed successfully.

Go: running c3

Clear

Go: execution terminated by null instruction.

Mars Messages

Run I/O

Reset: reset completed.

Reset: reset completed.

Clear

Enter two positive numbers, a and b
5
10
50
-- program is finished running (dropped off bottom)