

# **CS 200: Computer Organization**

## **Project 7: Random Number Generator**

Shariq M. Jamil

Due: Friday, April 4, 2014

## **Overview**

### **Purpose**

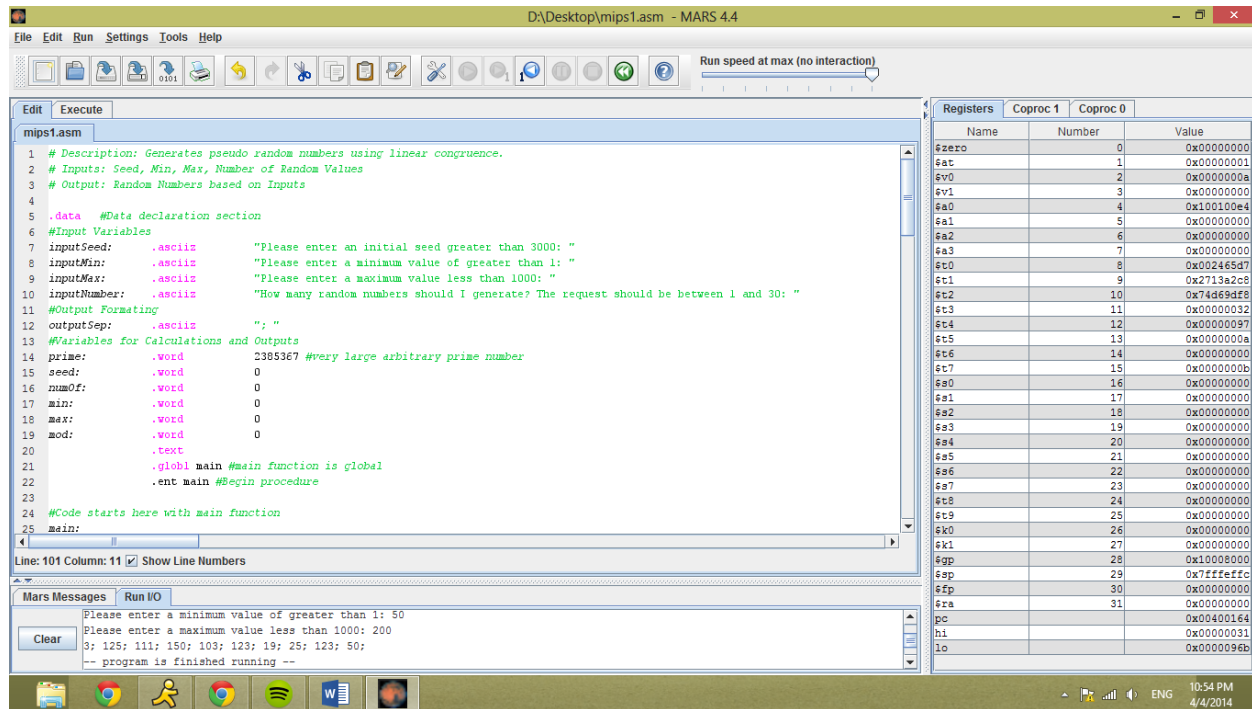
This project required us to write a program in MIPS assembly language that works as a random number generator. The program accepts a seed, a max value, a min value and the number of random numbers to be generated, and then displays the random numbers requested.

### **Approach**

I studied about Linear Congruence and Linear Congruential Generators in depth and broke the algorithm into steps. This helped me develop pseudocode and consequentially assembly code for the program. As I googled functions, I saw that there were numerous articles and books devoted to assembly programming that were very helpful. I had the flu this week and the week before, which lead to me having a hard time focusing on this project. I reached out to my friends for help and we coordinated on this project which made it a bit easier to complete as well. They told me that limits were a good idea for this project and I put them in but I am was not sure how big or small the seed was supposed to be. There were also many articles on Linear Congruence online which helped me understand the concept immensely as well. I started off programming using QtSpim but looked online to see if there were any alternatives because I did not like their interface (I am sure it looks better once the user gets used to it). I did find and use MARS (<http://courses.missouristate.edu/KenVollmar/MARS/index.htm>). It is an IDE developed by engineers at Missouri State University and I really enjoyed using it.

## Solution

## Sample Output



```
1 # Description: Generates pseudo random numbers using linear congruence.
2 # Inputs: Seed, Min, Max, Number of Random Values
3 # Output: Random Numbers based on Inputs
4
5 .data #Data declaration section
6 #Input Variables
7 inputSeed: .asciiz "Please enter an initial seed greater than 3000: "
8 inputMin: .asciiz "Please enter a minimum value of greater than 1: "
9 inputMax: .asciiz "Please enter a maximum value less than 1000: "
10 inputNumber: .asciiz "How many random numbers should I generate? The request should be between 1 and 30: "
11 #Output Formatting
12 outputSep: .asciiz "; "
13 #Variables for Calculations and Outputs
14 prime: .word 2385367 #very large arbitrary prime number
15 seed: .word 0
16 numOf: .word 0
17 min: .word 0
18 max: .word 0
19 mod: .word 0
20 .text
21 .globl main #main function is global
22 .ent main #begin procedure
23
24 #Code starts here with main function
25 main:
26     #Please enter a minimum value of greater than 1: 50
27     #Please enter a maximum value less than 1000: 200
28     #3: 125; 111; 150; 103; 123; 19; 25; 123; 50;
29     -- program is finished running --
```

Registers	Coproc 1	Coproc 0	
Name	Number		Value
\$zero	0		0x00000000
\$at	1		0x00000001
\$v0	2		0x0000000a
\$v1	3		0x00000000
\$a0	4		0x100100e4
\$a1	5		0x00000000
\$a2	6		0x00000000
\$a3	7		0x00000000
\$t0	8		0x002465d7
\$t1	9		0x2713a2c8
\$t2	10		0x74d69df8
\$t3	11		0x00000032
\$t4	12		0x00000097
\$t5	13		0x0000000a
\$t6	14		0x00000000
\$t7	15		0x0000000b
\$s0	16		0x00000000
\$s1	17		0x00000000
\$s2	18		0x00000000
\$s3	19		0x00000000
\$s4	20		0x00000000
\$s5	21		0x00000000
\$s6	22		0x00000000
\$s7	23		0x00000000
\$t8	24		0x00000000
\$t9	25		0x00000000
\$k0	26		0x00000000
\$k1	27		0x00000000
\$gp	28		0x10008000
\$sp	29		0x7fffffc
\$fp	30		0x00000000
\$ra	31		0x00000000
\$pc			0x00400144
\$hi			0x00000031
\$lo			0x0000096b

Mars Messages Run I/O

Please enter a minimum value of greater than 1: 50  
Please enter a maximum value less than 1000: 200  
3: 125; 111; 150; 103; 123; 19; 25; 123; 50;  
-- program is finished running --

Figure 1: Sample Output

## Conclusion

Being sick really caused me issues during this project and I was forced to work with classmates. It was a good exercise in teamwork and we used each other's strengths to get to the solution. One issue that I did not get fixed was that the RNG did not adhere to the minimum value entered by the user but the maximum value part worked great. I did really enjoy the MARS IDE for this assignment. It did not require installation and seems very easy to use. This was a good exercise in working on a problem I had never heard of before, finding the mathematical formula for solving it online, developing pseudocode

based on the formula, converting the pseudocode to code and debugging the program until it works. This process is very fulfilling for me and the skills I am developing now will be useful throughout my career in Computer Science (or even every day life).

## **References**

A Minimalistic Introduction to MIPS Instruction

<http://labs.cs.upt.ro/labs/so2/html/resources/nachos-doc/mipsf.html>

Linear, Congruential Random Number Generators

<http://lamar.colostate.edu/~grad511/lcg.pdf>

MIPS Assembly Language Programmer's Guide

<http://www.tik.ee.ethz.ch/education/lectures/TI1/materials/assemblylanguageprogdoc.pdf>

Wikipedia – MIPS architecture

[http://en.wikipedia.org/wiki/MIPS\\_architecture](http://en.wikipedia.org/wiki/MIPS_architecture)