Computer Science 3B

Practical Assignment 04

2017-08-24

Deadline - 2017-08-24 17h00 Marks: 75

This practical assignment must be uploaded to eve.uj.ac.za **before** 2017-08-24 17h00. Late or incorrect submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student.

Good coding practices include a proper coding convention and a good use of commenting. Marks will be deducted if these are not present. See the reminder page for more details.

There are many ways to generate random numbers. One method that is often used is to create a ***Linear Congruential Generator***. The generator is defined as follows:

$$X_{n+1} \equiv (aX_n + c)$$

where

a - *multiplier* c - *increment* X_n - *seed*

In order to generate a sequence of random numbers you initialise the seed to a known value, perform the above calculation and return the result (X_{n+1}) to the user as a random numbers. X_{n+1} then forms the seed value for the next iteration.

IMPORTANT STEP - Once you have calculated the seed value for the next iteration, and **BEFORE** you record it for use in the next iteration you **MUST** truncate its value. The following code can be used to do this:

```
.DATA
  bits    DWORD    7FFFFFFFh
.CODE
;the value that must be truncated is in eax
  AND eax, bits
;the truncated value is stored in eax
```

Write an 80x86 assembly program that uses a *stack-allocated* ***local one-dimensional array***[1]. The user must specify the size of the array and initial seed. Each element in the array must contain a random number that has been generated using the ***Linear Congruential Generator*** shown above. The **only** items that must be global in your application are the ***string constants*** and the ***constant integer values*** specified below:

a	1103515245	c	12345

[1] local arrays are allocated in the same way as local variables, however space on the stack is allocated for each element in the array

Mark sheet

1. Design [10]
2. Capture array size and seed [05]
3. Generate random number generation using formula [10]
4. Loop for generating random numbers [10]
5. Display array using loop [05]
6. Structure and layout (no extra globals, correct data types) [05]
7. Commenting [05]
8. Correct execution. [25]

NB

Submissions which **do not assemble** will be capped at 40%!

Execution marks are awarded for a correctly functioning application and not for having some related code.

Reminder

Your submission must follow the naming convention as set out in the general learning guide.
Example

Surname	Berners-Lee	**Initials**	TJ
Student number	209912345	**Module Code**	CSC3B10
Current Year	2017	**Practical number**	P04

`Berners-Lee_TJ_209912345_ CSC3B10_2017_P04`

Your submission must include the following in a single zip (compressed) file:

- *Source file* (`asm` file) - File containing your solution. Your details must be included at the top of the source code.
- *Program design* (`pdf` file) - File containing your design. Your details must be included at the top of the design.