CS 354 - FALL 1994
Assignment 7

Grader: Vaishnavi Anjur
Due before noon on 10/28/94

1 Purpose

The purpose of this assignment is to write a *SAL* program that deals with arrays.

2 The Problem

You are to implement a *random walk* , forward or backward through an array of 8 elements. The user enters a *3 digit* positive seed for random number generation. Using the seed, a random number n is generated scaled to

```
[-2 -1 0 +1 +2 ]
```

Walk forward or backward n steps depending on n being positive or negative.

Keep a running total within each element, each time adding in the random number that takes you to that element. If the walk takes you out of the array then you get to the last element of the array (ie.,*a[7]*). Add the integer that attempted to take you out of the array to the last element and continue.

At the end of 20 iterations print out the contents of each of the array elements and stop.

The array elements are to be initialized with zero and you are at the zeroth element to start with.

You are given a procedure that generates one random number each time it is called, in the required scale.

3 Sample Run

Several sample runs of the program might look :

```
Enter the seed for random number generation : 383

Sequence of random numbers generated

2 -1 2 1 -1 0 2 1 1 2 1 -1 2 -1 0 0 -1 -2 -1 1


Contents of the array :
0  0
1 -1
2  1
3  0
4  1
5  1
6 -1
7  6
```

4 The random number generator

A procedure *getrand* that generates a random number in the required scale, each time it is called ,will be provided to you. Copy the procedure from the file *rand.sal* in

```
~cs354-1/public.
```

In order to use the procedure *getrand* in your program, you must have the following declared in your program:

- A variable

  ```
  seed: .word
  ```

  that should be initialized to contain the random number seed entered by the user.

- A variable

  ```
  rand: .word
  ```

  that will contain the random number in the required range after each call to *getrand*

- A variable

  ```
  loop_label: .word
  ```

  that should contain the address of the instruction you want *getrand* to return to.

5 Requirements

The following requirements must be followed to receive full credit for the assignment.

- You may assume that the integer entered by the user will be small enough that it will not cause an overflow.
- If your program encounters illegal input, it should print an error message followed by a linefeed and halt.
- The program must contain procedures. You might want to use separate procedures to :
  - Read in the random number seed initially.
  - Check if a move takes you out of the array.
  - Update the contents of the appropriate array element after every move.
  - Print out the contents of the array elements.

6 How to turn in your assignments

You are to turn in the source code for your program by using the *handin7* program. When you have completed the assignment, hand in your source code by typing:

```
~cs354-1/bin/handin7 filename
```

where filename is the file containing the source code. You may receive feedback from the grader of your assignment by handing in a printout of your source code.