Simulation Example

Random Number Generator

I will assume that Random is a function that returns a random number between 0 and 1. As discussed in class a method for creating such a function is:

```pascal
const C = 25173;
D = 13849;
M = 32768;

var Seed : integer;

function Random : real
begin
  Seed := (C * Seed + D) mod M;
  random := Seed / M;
end;
```

the variable Seed must be assigned a randomly selected value between 0 and M-1 at the beginning of the program.

Example

PROBLEM: We want to run a simulation to approximate the probability that in four rolls of a die we observe at least one 6. To do this we need to simulate rolling a die. The statement

```pascal
x := Random;
```

"randomly" assigns a number between 0 and 1 to the variable x. Note that if 0 < x < 1 then 0 < 6x < 6. Also since x is equally likely to be any number between 0 and 1, 6x is equally likely to be any number between 0 and 6. (This will be proven later in the course.) This means that the integer part of 6x is equally likely to be any one of the integers 0, 1, 2, 3, 4, 5. So the statements:

```pascal
x := Random;
y := Trunc(6 * x);
```

would randomly assign one of the integers 0, 1, 2, 3, 4, 5 to the variable y, with each number being equally likely. If we modify this to:

```pascal
x := Random;
y := Trunc(6 * x) + 1;
```

then y would be assigned one of the integers 1, 2, ..., 6 with each number being equally likely. This is exactly what we want for simulating the roll of a die. We could combine these two statements into one statement:

```pascal
y := Trunc(6 * Random) + 1;
```
The following code simulates rolling a die 4 times. After each roll it checks to see if a six was rolled. If a six is rolled it changes the value of the variable Yes to a 1:

```pascal
Yes := 0;
for i := 1 to 4 do
  if ( Trunc(6 * Random) + 1 = 6 ) then Yes := 1;
```

This performs the experiment once. Now we would like to perform the experiment many times and count how many of those times Yes turns out to be 1. Say we repeat the experiment 1000 times. We can set up a loop to repeat the above simulation 1000 times and keep count of the number of times Yes comes out to be 1 in a variable called Count. Then we can approximate the probability of observing at least one six in four rolls by Count/1000:

```pascal
program dice;
var i, j, Yes, Count : integer;
  p : real;
begin
  Randomize (* This procedure places a random seed in the random number generator in Turbo Pascal *)
  Count := 0;
  for j := 1 to 1000 do
    begin
      Yes := 0;
      for i := 1 to 4 do
        if ( Trunc(6 * Random) + 1 = 6 ) then Yes := 1;
      Count := Count + Yes;
    end;
  p := count / 1000;
  Writeln('The approximate probability is ', p:4:3);
end;
```

In 10 separate runs of this program I get the following results:

0.510, 0.516, 0.532, 0.537, 0.507, 0.519, 0.513, 0.502, 0.515, 0.542