# EECS 10: Assignment 4

October 14, 2005

Due Monday 10/24/2005 12:00pm

## 1 Guess the Number [20 points + 5 points (extra credit)]

Write a program that plays the game of "guess the number". In this game, the computer "thinks" of a random number between 0 and a user-specified upper limit and the player has to guess this number. The computer will help the player by giving hints on whether the guessed number is less than or greater than the chosen number.

The format of the program should be as follows:

First, the computer asks the user for the upper bound for generating the random number. Your first line of output should display something like:

```
Enter the upper bound for the random number:
```

Once the user has entered the upper bound (say **`n=1000`**), your program chooses the number to be guessed by selecting an integer at random in the range **`0`** to **`(n-1)`**. The program then displays the following:

```
"******Guessing Game******"
I have selected a number in the range of 0 to 999.
Can you guess the selected number?
Please enter your guess number 1:
```

The player then types a first guess. The program responds with one of the following according to the guess made:

- `Great!! You guessed it right! You have made XXX guesses.`
- `Your number was too low. Please try again!`
- `Your number was too high. Please try again!`

If the player's guess is incorrect, your program should help the player to zero in on the correct answer by repeating the hints until the player finally gets the number right. Your program should also count the number of guesses the player makes, and display the number of guesses at the end (in the successful text above, replace the **`XXX`** with the proper number of guesses).

To show that your program works correctly, play it once with the upper bound 1000 (i.e. range from 0 to 999) and submit this run as your script file (**`guess.script`**).

**HINT**
For generating the initial random number, you have to use a random number generator which is provided by the C standard function **`rand()`**. This function generates a random number of type **`int`** in the range of **`0`** to **`32767`**. This function is provided in the header file **`stdlib.h`**.

In practice, no computer functions can produce truly random data -- they only produce pseudo-random numbers. These are computed from a formula and the number sequences they produce are repeatable. A seed value is usually used by the random number generator to generate a number. Therefore, if you use the same seed value all the time, the same sequence of "random" numbers will be generated (i.e. your program will always produce the same "random" number in every program run). To avoid this, we can use the current time of the day to set the random seed, as this will always be changing with every program run. With this trick, your program will produce different guesses every time you run it.

To set the seed value, you have to use the function **srand()**, which is also provided by header file **stdlib.h** . For the current time of the day, you can use the function **time()**, which is defined in header file **time.h** (**stdlib.h** and **time.h** are header files just like the **stdio.h** file that we have been using so far).

In summary, use the following code fragments to generate the random number for the game:

1. Include the **stdlib.h** and **time.h** header files at the beginning of your program:

```
#include <stdlib.h>
#include <time.h>
```

2. Include the following lines at the beginning of your main function:

```
/* initialize the random number generator with the current time */
srand(time(0));

/* generate the random number in the range 0 to (n-1) */
randomNumber = rand() % n;
```

Here, **n** specifies the upper bound of the range in which the random number will be generated, and **randomNumber** is the integer variable which is assigned the generated random number.

### For 5 Extra Credits:

How many steps at most does it take for a "Good Guesser" to guess any number between 0 and 999? Explain why!

Files to submit for this problem are: **guess.c**, **guess.script**, and **guess.txt.**
**Note:** File **guess.txt** should briefly (!) explain your program, as well as the extra credit part.

## 2 Exercise 4.26 from the textbook, Page 140 [20 points]

**Additional instructions:** Your program should ask the user for the number of terms **(say n)** to be used in the series and should print the table in the following format, starting with number of terms to be **1** in the infinte series to **n** :

```
No Terms  |  approx. PI  |  actual PI  |  difference  | accuracy
   1      |  4.00000000  |  3.14159265 |  0.85840735  | 27.3240%
   2      |  2.66666667  |  3.14159265 |  0.47492598  | 15.1174%
   3      |  3.46666667  |  3.14159265 |  0.32507402  | 10.3474%
   .      |  ...         |  ...        |  ...         | ...
   .      |  ...         |  ...        |  ...         | ...
   .      |  ...         |  ...        |  ...         | ...
   n      |  ...         |  ...        |  ...         | ...
```

For "actual PI", you may take the constant value shown above. For "accuracy", simply compute the difference as a percentage of the actual PI value.

**Note:** Also, for your script file, please submit only a table with the first 15 lines! **(n=15)**

Files to submit for this problem are: **4.26.c**, **4.26.script**, **4.26.txt.**

## 3 Submission

Submission for these files will be similar to last week's assignment. The only difference is that you need to create a directory called **hw4/**. Put all the files listed above in that directory and run the **/ecelib/bin/turnin** command to submit your homework.