BIL 105E – Introduction to Scientific and Engineering Computing (C)

Spring 2015-2016

Homework 2

Assignment Date: 27.03.2016
Due Date: 11.04.2016, 23:00

IMPORTANT:

- Don't use or get inspired by any lines of code from any other sources (friends, Internet, etc). **Any similarity, which is beyond reasonable, will be accepted as cheating!**
- Name your program as **student_id.c** and don't forget to test it on your ITU account before submission by using ssh client. Any code that can't be compiled will not be evaluated.
- Please just use the subjects **shown in the class**. Don't use any other statements or data structures (e.g., goto, arrays, structs).
- You must submit a report based on the format given on Ninova course page.

The aim of this homework is producing a set of random numbers based on the random number generation functions specified by the user AND printing the histogram of these numbers in base 500. A pseudo-random number generator (PRNG) is used to generate a sequence of numbers that can't be predicted beforehand by its users. You have already used the PRNG supplied by C programming language, by means of srand() and rand() functions.

In this homework your program will implement these five steps:

(1) Firstly you will implement two PRNG function. So your program ask user for the PRNG (**prng**): either MID-PRNG or LSD-PRNG.

(2) Ask user for the **seed**.

(3) Ask user for number of samples (**num_samples**) to generate.

(4) By using the **prng** and starting from the **seed**, your program will produce **num_samples** random numbers. Each random number will be mapped to following ranges: range1=1..100, range2=101..200, range3=201..300, range4=301..400, and range5=401..500.

(5) After finishing generating **num_samples** random numbers and counting their occurences within each range, present the results to the user.

You will write two PRNG functions and test them in a C program:

1) MID-PRNG uses the following method for generating a sequence of pseudo-random numbers:

(a) Take a **number** which have 5 digits. This **number** will be used as the seed.

(b) take the **square** of the **number** (i.e., **square** = **number** * **number**)

(c) Pick out the middle five digits of the **square** as a random **number**. The **square** can be either 9 or 10 digits. In both cases, you will skip first two digits of the **square** and then take the following five digits as the new random **number** sample.

(d) go to (b) to generate a new random number.

Following example shows the random number generation by using MID-RNG algorithm, starting with the **number**=55421:

square = 55421 * 55421 = 30**71487**241 ==> **number** = 71487

square = 71487 * 71487 = 51**10391**169 ==> **number** = 10391

square = 10391 * 10391 = 10**79728**81 ==> **number** = 79728

…….

2) LSD-PRNG uses least significant (i.e., rightmost) digits of a product to generate numbers as follows:

(a)Program takes a 5-digit **number** as an input that is **neither even nor ends in a 5 (your program must control whether the number is odd and not ends in a 5).** It is the seed, i.e., the first number

(b) multiply the **number** with 73 to get a **product**.

(c) Pick out the low-order five digits of the **product** as a random **number**.

(d) go to (b) to generate a new random number. Repeat the steps based on the **num_samples** number specified by the user**.**

Following example shows the random number generation by using LSD-RNG algorithm, starting with the **number**=55421:

product = 55421 x 73 = 40**45733** ==> **number** = 45733

product = 45733 x 73 = 33**38509** ==> **number** = 38509

product = 38509 x 73 = 28**11157** ==> **number** = 11157

…….

Write two functions:

void mid-rng(long *number) and **void lsd-rng(long *number)**

The functions use **number** parameter as the current number, generate a random number by using the steps defined above, and returns back the generated random number within the **number** parameter. Please note that, return type of both functions are void. So, you have to use **call by reference**.

Your program flow is as follows:

(1) Ask user for the PRNG (**prng**): either MID-PRNG or LSD-PRNG?

(2) Ask user for the **seed**.

(3) Ask user for number of samples (**num_samples**) to generate.

(4) By using the **prng** and starting from the **seed**, your program will produce **num_samples** random numbers.

> (4.1) Each **sample** is mapped to a **sample space** which ranges from 1 to 500 by using modular artihmetic (ex: 45733 % 500=233. Thus, 45733 is mapped to 233 + 1 = 234).
>
> (4.2) Divide the **sample space** into 5 ranges: range1=1..100, range2=101..200, range3=201..300, range4=301..400, and range5=401..500.
>
> (4.3) In which range the **sample** is? Increment number of samples within that range.
>
> (4.4) If number of samples generated is less than the **num_samples**, then generate a new sample as described by the **prng** and repeat the steps (4.1) to (4.3).

(5) After finishing generating **num_samples** random numbers and counting their occurences within each range, present the results to the user. For each range, do the following:

> (5.1) Print the minimum and maximum values of the range (e.g., for range1 print 1..100:)
>
> (5.2) Calculate the percentage of occurences within each range.
>
> (5.3) Draw a histogram to show number of occurences. Histogram prints '*' characters to show percentages for each range. The number of '*' characters must be proportional with the range's percentage value (e.g., one or two '*' characters per 1%).
>
> (5.4) print the percentage value within parantheses and end the line to print the information about the next range.
>
> (5.5) If there is any remaining range, repeat the steps (5.1) to (5.4).

Define two functions named as **take_samples()** and **print_histogram()** to generate random numbers as in (4) and print out the histogram as in (5), respectively. You have to decide on the return type and type/number of formal parameters for both functions.

An example program run is given below. **Please use the same output format when you write your programs.** Note that, it is just an example and the percentage vaules in the histogram are dummy numbers. Your program run with the given parameters can mismatch with these numbers.

```
$ Which PRNG Algorithm?

   1. MID-PRNG

   2. LSD-PRNG

$ 1

$ Enter the seed

$ 89452

$ Enter Number of Samples

$ 500

$ 500 random numbers were generated by using MID-PRNG:

    1..100: ***************** (17%)

101..200: *********************** (23%)

201..300: ************************** (26%)

301..400: *************** (15%)

401..500: ******************* (19%)
```