Laboratory 05 – loops

09.11.2021

1. [3points] Write a C program that will calculate the number of digits of the number given by the user (`scanf`). Use a `do-while` loop. To check the result, use the logarithm and the ternary operator (? :) to count the number of digits in a given integer.

a)What arithmetic operation should be used to count the digits of an integer.?
b)What happens to the value of a user-supplied variable after counting the digits?
c)How many variables are needed?

Test data:

```
Enter any number: 123456789
Total digits: 9
Total digits (log10): 9

Enter any number: 0
Total digits: 1
Total digits (log10): 1
```

2. [5points] Write a C program to check if user-specified integer is a palindrome.

A palindrome is a word, number, phrase, or other sequence of characters which reads the same backward as forward, such as *madam*, *racecar*.
There are also numeric palindromes: *121*, *12321*, ...

The algorithm:
a) Use the `scanf` function to input an integer (`n`).
b) Make a copy of `n`.
c) Find the reverse of `n` and assign it to `rev`.
Example: Reverse of 1234 is 4321.
The `rev` calculation should be performed in the `while` loop.
What will be the loop termination condition?
The body of the `while` loop consists of two lines.
- The first line is the calculation of `rev`.
- The second line is necessary for the loop to end.
To calculate the `rev` values use: `*, +, %, rev, n, 10`.
d) Check if `rev` is equal to copy of `n`. If so, the number is a palindrome, if not, it is not.

Test data:

```
Enter any number to check palindrome: 1234
1234 is not palindrome.

Enter any number to check palindrome: 121
121 is palindrome.

Enter any number to check palindrome: 12321
12321 is palindrome.
```

3. [5points] In mathematics, the Fibonacci numbers (Fn), form a sequence, called the Fibonacci sequence, such that each number is the sum of the two preceding ones, starting from 0 and 1.

$$F_n := \begin{cases} 0 & \text{dla } n = 0, \\ 1 & \text{dla } n = 1, \\ F_{n-1} + F_{n-2} & \text{dla } n > 1. \end{cases}$$

Write a C program that computes the n terms of the Fibonacci sequence.
a) Use a `for` loop.
b) How many variables are needed in calculations inside the loop?
c) What are their starting values?

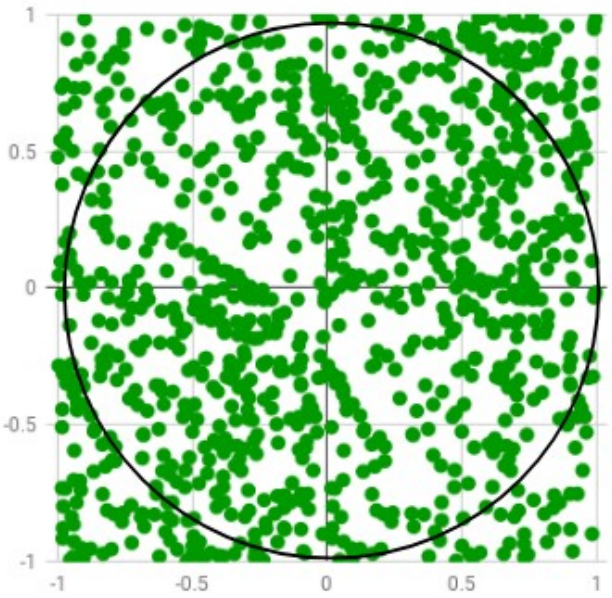
Test data:

```
Enter number of terms: 15
Fibonacci sequence:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

4. [5points] Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. One of the basic examples of getting started with the Monte Carlo algorithm is the estimation of Pi.

Estimation of Pi

The idea is to simulate random `(x, y)` points in a `2D` plane with domain as a square of side `1` unit. Imagine a circle inside the same domain with same diameter and inscribed into the square. We then calculate the ratio of number points that lied inside the circle and total number of generated points.



We know that area of the square is `1` unit sq while that of circle is $\pi \cdot \left(\frac{1}{2}\right)^2 = \frac{\pi}{4}$. Now for a very large number of generated points, $\frac{\text{area of the circle}}{\text{area of the square}} = \frac{\text{number of points inside the circle}}{\text{number of points inside the square}}$ that is, $\pi = 4 * \frac{\text{number of points inside the circle}}{\text{number of points inside the square}}$.

The algorithm:
a) Initialize a variable that counts the number of points inside the circle.
b) In the loop:
    c) Generate a pseudo-random `x` coordinate between `-1` and `1`.
    d) Generate a pseudo-random `y` coordinate between `-1` and `1`.
    e) Check if the drawn point lies inside the circle with a radius of `1`.
    f) If so, increase the appropriate variable.
g) Repeat the calculation in loop `7000`.
h) Calculate and print out your estimate for `pi`.

Generate pseudo-random number
Function `int rand (void)` returns a pseudo-random integral number in the range between `0` and `RAND_MAX`. `RAND_MAX` is a constant defined in `<stdlib.h>`.

Test data:

```
Estimation of Pi = 3.171200
```

5. [2points] Let's see how the estimate value depends on the number of pseudo-random number generations. Add a loop to the previous program that will control the number of pseudo-random number generations.

Test data:

```
Estimation of Pi (1.0e+01) = 3.200000
Estimation of Pi (1.0e+02) = 2.800000
Estimation of Pi (1.0e+03) = 3.132000
Estimation of Pi (1.0e+04) = 3.120000
Estimation of Pi (1.0e+05) = 3.147400
Estimation of Pi (1.0e+06) = 3.141496
Estimation of Pi (1.0e+07) = 3.141903
Estimation of Pi (1.0e+08) = 3.141712
Estimation of Pi (1.0e+09) = 3.141567
```

Next time:
Laboratory 06 – Arrays