CS2211a Assignment 4

Issued on: Thursday, November 7, 2013

Due by: Thursday, 11:55 pm, November 14, 2013

- In this assignment, only electronic submission (***attachments***) at owl.uwo.ca is required. Attachments must include:
    - ***ONE pdf*** file that has the three flowcharts, all sample outputs, and any related communications or discussions for all questions.
    - ***text*** soft copy of the programs that you wrote for each question (*one program attachment per question*), i.e., ***THREE*** C program files in total.

  So, in total, you will submit 1 + 3 = 4 files.

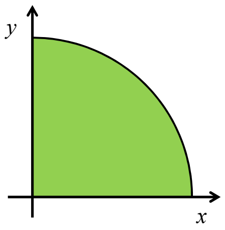
- Late assignments are strongly discouraged
    - 10% will be deducted from a late assignment (up to 24 hours after the due date/time)
    - After 24 hours from the due date/time, late assignments will receive a zero grade.

Program 1 (30 marks)

The area of a circle of radius r is given by

$$Area\ of\ a\ circle = \pi\times r^{2}.$$

Imagine that you divided this circle exactly into 4 quadrants. The area of one quadrant is then $0.25\times\pi\times r^{2}$.



Let us set the radius of this circle to be $r = 1$. The equation hence becomes

$$Area\ of\ one\ quadrant\ of\ a\ circle\ of\ radios\ 1 = 0.25\times\pi.$$

We can simply say:

$$\pi = 4\times the\ area\ of\ one\ quadrant\ of\ a\ circle\ of\ radios\ 1.$$

To calculate the area of the shaded quadrant of the circle, use a random number generator and guess effectively the correct value of the constant π. The technique you should use is to:

- Generate a random number between 0.00 and 1.00 and assign it to x.
- Generate a random number between 0.00 and 1.00 and assign it to y.
- If $sqrt(x^{2}+y^{2}) < 1$, it means that this (x,y) coordinate lies inside the shaded quadrant.
- Repeat these steps N times, where N is *sufficiently large number* and then calculate the ratio of the points located inside the circle to the total number of generated points N. This ratio should approximate the area of one quadrant of this circle.
- Apply this to the equation above to find the mathematical constant π.

Using the above fact, draw a *flowchart* and write a *C program* to calculate the mathematical constant of π.

Add as many inline comments as you can to make your program well documented and easy to be understood by anyone who reads it.

- Test your program using

  $N = 10$, $N = 100$, $N = 1000$, $N = 100000$, $N = 1000000$, $N = 10000000$, *and* $N = 100000000$.

- Re-run your program *10 times* using the above values of N. Calculate the *mean* and *standard deviation* for *results generated from the same value of N*. Report all results (i.e., 70 values, 7 means and 7 standard deviations).
- Discuss your results.
- You should include enough *actual* test cases in your submission to demonstrate the functionality of your program (i.e., covering all cases).

Program 2 (35 marks)

Draw a flowchart **and** write a program that prints an n×n *magic square* (a square arrangement of the numbers $1, 2, \ldots, n^2$ in which the sum of the elements in any row, column, or diagonal is the same). This program should generate a magic square of a specified size n, where the user will specify this size. The size must be an *odd positive* integer number between 1 and 99 (to be validated by your program). You should store the magic square in a two-dimensional array.

Start by placing the number 1 in the middle of row 0. Place each of the remaining numbers 2, 3, …, n^2 by moving up one row and over one column. Any attempt to go outside the bounds of the array should wrap around to the opposite side of the array. For example, instead of storing the next number in row -1, we would store it in row n-1 (the last row). Instead of storing the next number in column n, we would store it in column 0. If a particular array element is already occupied, put the number directly below the previously stored number. Here it is a sample output:

```
Enter size of magic square: 6
Invalid size, try again...
Enter size of magic square: 3
    8    1    6
    3    5    7
    4    9    2
```

```
Enter size of magic square: 5
17        24        1        8        15
23         5        7       14        16
 4         6       13       20        22
10        12       19       21         3
11        18       25        2         9
```

Add as many inline comments in your program as you can to make it well documented and easy to be understood by anyone who reads it.

You should include enough *actual* test cases in your submission to demonstrate the functionality of your program.

Program 3 (35 marks)

Draw a flowchart **and** *write a function* that determines the smallest number of \$20, \$10, \$5, \$2, and \$1 bills/coins necessary to pay a dollar amount. The *function prototype* must be as follow:

```
void pay_amount(int dollars, int *twenties, int *tens, int *fives, int
*toonies, int *lonnie);
```

where the amount is represented by the dollars parameter. The twenties parameter points to a variable in which the function will store the number of \$20 bills required. The tens, fives, toonies and lonnie parameters are similar. You should add as many *inline comments* as you can to make your code well documented and easy to be understood by anyone who reads it.

To test your function, *write a program* that asks the user to enter an integer value for the payment amount, calls pay_amount to get the smallest number of bills/coins necessary to pay the amount, and then display the values returned by the function.

You should include enough *actual* test cases in your submission to demonstrate the functionality of your program (i.e., covering ***all*** cases).