Informatik für Mathematiker und Physiker HS12

Exercise Sheet 10

Submission deadline: 3.15pm - Tuesday 27th November, 2012

Course URL: http://www.ti.inf.ethz.ch/ew/Lehre/Info1_12/

Assignment 1 - (8 points)

The game Tic Tac Toe (also know as *Noughts and Crosses*) is a simple 2-player game that is usually played on a 3×3 grid. The players take turns to place their marks (the first player places **O**, the second **X**) and who gets first 3 of his marks in a row on a horizontal, vertical or diagonal line is the winner.

We can generalize the game to be played on a grid of a size $r \times c$ (r rows by c columns) and the goal is to have n marks in a row on a horizontal, vertical or diagonal line. Let us refer to the squares of a grid via coordinates were the first number refers to a row and the second to a column. That is $(0, 0)$ refers to the top-left corner and $(r - 1, c - 1)$ refer to the bottom-right corner squares. We also place the following restrictions: $r \geq 3$, $c \geq 3$ and $3 \leq n \leq min(r, c)$.

Your task is to write a program ticTacToe.cpp that allows two players to play Tic Tac Toe. The program should read from the standard input r, c and n. Then the game should start, players take turns and input their moves (coordinates of squares where they want to place their marks). It is illegal to put a mark on a non-empty square or outside of the grid. Your program should be able to determine when the game is over (there is a winner or the game is a draw) and also, if there is a winner, who the winner is. The game is a draw if neither of the players is a winner and there is no empty square on the grid. See Figure 1 and Figure 2 for the example input and output of the program.

You can use the provided program skeleton ticTacToeSkeleton.cpp that can help you to structure your program. You can also use sample inputs tttO.in, tttX.in and tttDraw.in to help you to test your program.

```
How many rows? 3
How many columns? 3
How many in line to win? 3
1 1
0 2
0 0
2 2
1 2
1 0
0 1
2 1
2 0

The game is a draw.
```

Figure 1: Example input and output of the program.

O	O	X
X	O	O
O	X	X

Figure 2: The sequence of moves on from Figure 1 results to the following board.

Assignment 2 - (4 points)

Consider the generator `ansic` used in choosing_numbers.cpp. Since the modulus is $m = 2^{31}$, the internal computations of the generator will certainly overflow if 32 bits are used to represent `unsigned int` values. Despite this, the sequence of pseudorandom numbers computed by the generator is correct and coincides with its mathematical definition. Explain this!

Assignment 3 - (4 points)

Find a loaded dice that beats the fair dice in the game of choosing numbers. (This is a theory exercise.)

To recall, here are the rules for one round of choosing numbers: Each of two players independently writes down an integer between 1 and 6. Then the numbers are compared. If the are equal, the game is a draw. If the numbers differ by one, the player with the smaller number gets CHF 2 from the one with the larger number. If the two numbers differ by two, or more, the player with the larger number gets CHF 1 from the one with the smaller number.

The exercise asks for a loaded dice which wins against a fair dice on average. The fair dice produces every number between 1 and 6 with the same probability $\frac{1}{6}$.

Challenge - (8 points)

Exercise 151 from the lecture notes.