Tutorial
(Advanced Programming)
Worksheet 5:

Assignment 1: Tic Tac Toe

There shouldn't be any necessity to explain the Tic Tac Toe Game[1]! But you have to "explain" it to the computer by writing the corresponding game:

Define 9 *global variables* a00, a01, a02, a10, a11, ..., a22 of type char. Each variable represents one of the 9 cells. In case that you are already familiar with arrays, you can also use arrays to represent the cell state.
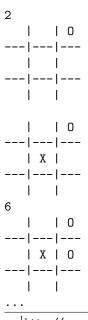
Use 0 for an untouched cell, 1 for a cell ticked by the computer and 2 for a cell marked by the human player.

The input from the user is a number in the range $[0; 8]$, each associated to a cell.

Your task is to write

- a program which is capable of handling the human input values.

- an appropriate handling of invalid input data.

- a computer which is able to act in a optimal way using backtracking[2].

- an output routine for the game field.

Use standard console output to display the game-field after each round, e. g.

```
2
   |   | O
---|---|---
   |   |
---|---|---
   |   |

   |   | O
---|---|---
   | X |
---|---|---
   |   |
6
   |   | O
---|---|---
   | X | O
---|---|---
   |   |
...
```

---

[1]http://en.wikipedia.org/wiki/Tic-tac-toe
[2]http://en.wikipedia.org/wiki/Backtracking

Assignment 2: Ballmer Peak OR don't drink & program

Your next assignment is to develop the core part of a screensaver which prohibits drunken programmers from accessing their computer to avoid damaging their own code.

When humans accidently drink too much alcohol, their reaction time decreases. For our protection screensaver, we do it vice versa: We measure the reaction time of the user and relate this to the blood alcohol concentration.

Now it is up to you to write a program which

1. uses a random delay to wait for user input.

2. measures the current time when requesting some reaction from the user via console output.

3. measures the elapsed time when the user hit the key.

4. outputs the elapsed time to the console.

Use the function *gettimeofday(...)* which is described in the corresponding manpage[3] (Type 'man gettimeofday' in your console):

```
...
int gettimeofday(struct timeval *tv, struct timezone *tz);

struct timeval {
    time_t      tv_sec;      /* seconds */
    suseconds_t tv_usec;     /* microseconds */
};
...
```

Figure out the type of *time_t* and *suseconds_t* and think about possibly necessary type castings before you start programming(!).

Use the function *usleep(...)* to delay the program output with a random time. More information about this function is also available via the man-page.

Homework assignment 2: Ballmer Peak continued

Extend the second assignment also to output a random ASCII character requesting the user to type exactly this character followed by pressing [ENTER] including measuring how long it takes to type this character. Otherwise the user is assumed to be unable to program.

Hints:

- Character 'a' is related to a specific number representation. Character 'b' is also represented by the number of 'a' plus 1. All other characters are similarly represented by an increasing sequence of numbers.

- figure out the number associated to the character 'a'. Since

  ```
  std::cout << 'a' << std::endl;
  ```

---

[3]http://www.kernel.org/doc/man-pages/

is printing the character 'a' to the terminal, you have to cast the character to another type.

- Output a random character by adding a random value within the range $[0; 25]$ to the value of character 'a' and cast it to a character.

Questions:

Answer the following questions:

- Assignment 1: It is indeed possible to write an KI which makes it impossible to let a human player win. If you consider writing such an AI for the game chess or draughts[4]), can you image some problems into which you are running into?

- Assignment 2: Why is the way of measuring the elapsed time not very accurate (This question is not related to the relation between reaction-time and blood alcohol concentration!)? Do you have any ideas to improve this? How is this related to HPC benchmarks?

- Assignment 2: What demands on the hardware and software architecture can you imagine to allow accurate measurements.

---

[4]http://en.wikipedia.org/wiki/Draughts