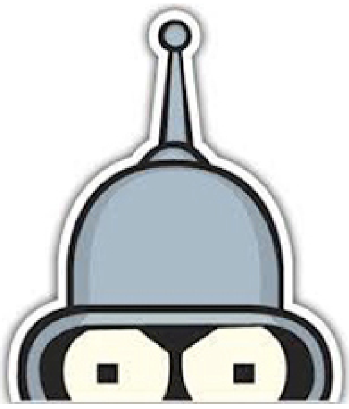
B1 - Elementary Programming in C

B-CPE-111

Matchstick

Leave the last matchstick!



2.0

Matchstick

binary name:	matchstick
repository name:	CPE_matchstick_*$ACADEMICYEAR*
repository rights:	ramassage-tek
language:	C
compilation:	via Makefile, including re, clean and fclean rules

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

For this project, the **only** authorized function are `read`, `write`, `malloc`, `free`, `time`, `getpid`, `random`,`srandom` and `getline`.

This project is based on a very famous game based on matchsticks.
There is a certain number of matchstick lines.
The two players take turns; each player can, on a same line, take one or several matchsticks.
The losing player is the one to take the last matchstick.

The goal of the project is to create a program that you can play against.
The basic version must generate a game board with **n** matchstick lines (*1 < n < 100*) and have a simple interface so that the user could play against the computer.

The number of lines is given as parameter to the program.
The second parameter of the program indicates the maximum number of matches that can be taken out each turn (must be > 0).
When matchsticks are removed, they must be removed starting from the right (like in the example below).
The user will always start the game.

The program's output (error messages included) must correspond to the examples below. Error messages which happens during a game and which are destined to the player must be displayed on the standard output.
In case of bad input, you must ask for the line by displaying "**Line:** " again, and it's up to the player to indicate again the line he/she wants to play on.

If the user wins, the program must return 1.
If the AI wins, it must return 2.

Here is a list of bonuses you may implement (in the bonus, built with a separate Makefile's rule):

- a graphical interface,
- a more complete game, with score and several levels of difficulty,
- a human-machine interface device (such as a leap motion for instance) to select and remove the matchsticks.

Examples of usage and expected output can be found in a text file next to this subject.