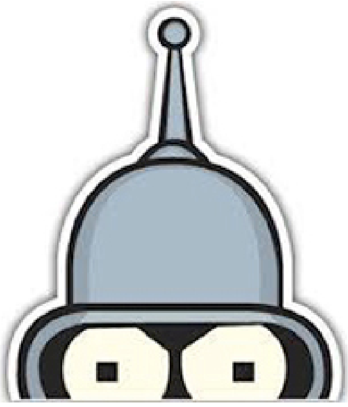

B1- Elementary Programming in C

B-CPE-085

GetNextLine

a very first parsing tool

v1.3

GetNextLine

a very first parsing tool

repository name:	CPE_$YEAR_getnextline
repository rights:	ramassage-tek
language:	C
group size:	1

- 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 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).

You must submit **get_next_line.c** and **get_next_line.h** (both of which must be located at the directory's root), but no **Makefile** nor **main** function.

The goal of this project is to write a function that returns a read line from a file descriptor.
If there are no more lines to return, or if there is an error during the reading, the function will come back **NULL**.

You must define a macro called **READ_SIZE** in your **get_next_line.h** file, which indicates the amount of characters to be read for each **read()** call, defined (including the **#ifndef** protection) as below:

```
# ifndef READ_SIZE
#  define READ_SIZE (/* value here */)
# endif /* !READ_SIZE */
```

The value of this macro may be changed (by another strictly positive value) during the evaluation in order to verify that you are using it correctly.

You must use one (or several) static variable(s) in order to save the characters that were read but not returned.

The function should be prototyped as follows:

```
char *get_next_line(const int fd);
```

get_next_line should return the results **without** the '\n'.

The only authorized functions are **read**, **malloc** and **free**.

Example

Here is an example of use of the function:

```c
/*
** main.c for get_next_line_main_test in
**
** Made by Richard Stallman
** Login   <richard.stallman@epitech.net>
**
** Started on  Mon Nov 5 14:59:03 2031 Richard Stallman
** Last update Mon Nov 5 14:59:49 2031 Richard Stallman
*/

#include "my.h"
#include "get_next_line.h"

int         main()
{
        char    *s;
        while ((s = get_next_line(0)))
        {
                my_putstr(s);
                my_putchar('\n');
                free(s);
        }
        return (0);
}
```