B1 - B-CPE-101 - UNIX & C LAB SEMINAR

Final Stumper
    binary name: rush3
    language: C
    compilation: via Makefile, including re, clean and fclean rules

/!\ WARNING /!\
    · 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).

INSTRUCTIONS
    You must develop a program that reads a character string (that represents the output of the
    first rush) and displays the assignment's name and dimensions. If it doesn't match any of the
    assignments, you must display "none". Regardless of the response, you must terminate your
line
    with '\n'.

/!\ WARNING /!\
    malloc, free, read and write are the only syscalls allowed.

/!\ HINT /!\
    You may use your lib, but it must be rebuild when your project is.

EXAMPLES
    ~/B-CPE-101> ./rush1-1 4 4
    o--o
    | |
    | |
    o--o
    ~/B-CPE-101> ./rush1-1 4 4 | ./rush3
    [rush1-1] 4 4
    ~/B-CPE-101> ./rush1-2 3 4 | ./rush3
    [rush1-2] 3 4
    ~/B-CPE-101> ./rush1-3 1 1
    B
    ~/B-CPE-101> ./rush1-4 1 1
    B
    ~/B-CPE-101> ./rush1-5 1 1
    B
    ~/B-CPE-101> ./rush1-4 1 1 | ./rush3
    [rush1-3] 1 1 || [rush1-4] 1 1 || [rush1-5] 1 1

TEST MAIN
Below you will find a main function that you can either deliver as-is (beware to respect the
current coding style!) or modify:

```
#include <unistd.h>
#include "rush3.h"
int main()
{
    char buff[BUFF_SIZE + 1];
    int offset = 0;
    int len ;
    while ((len = read(0 , buff + offset , BUFF_SIZE - offset)) > 0) {
        offset = offset + len ;
    }
    buff[offset] = '\0';
    if (len < 0)
        return (84) ;
    rush3(buff) ;
    return (0) ;
}
```