



B1 - Unix & C Lab Seminar

B-CPE-101

EvalExpr

Evaluating an Arithmetic Expression





EvalExpr

binary name: `eval_expr`

language: C

build tool: via Makefile, including `re`, `clean` and `fclean` rules



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.



You are allowed not to use any system functions, except *write*, *malloc* and *free*.

The goal of this project is to write the `eval_expr` function, which must be prototyped as follows:

```
int eval_expr(char const *str);
```

It takes a string as parameter, which represents a mathematical expression, evaluates this expression and returns the result as an integer.

The string received as parameter will always be valid (no syntax errors, no divisions by zero,...).

The following five operators must be supported:

- '+' for addition,
- '-' for subtraction,
- '/' for division,
- '*' for multiplication,
- '%' for modulo.

The function must also handle any number of parenthesis.

You must use the following main function:

```
int main(int ac, char **av)
{
    if (ac == 2) {
        my_put_nbr(eval_expr(av[1]));
        my_putchar('\n');
        return (0);
    }
    return (84);
}
```



EXAMPLE

Here is an example of how the function will be tested:

```
Terminal
~/B-CPE-101> make clean; make all
~/B-CPE-101> ./eval_expr "(3+2)*5"
25
```