C - Pool - Tek1

Subject EvalExpr

C Pool Managers
looneytunes@epitech.eu

Contents

Instructions

- The subject may change until one hour before turn-in.
- Respect the norm takes time, but is good for you. This way your code will respect the norm since the first written line.
- Turn-in directory:
  `Piscine_C_evalexpr`

  Hints `Remember it is always better to create your repository at the beginning of the day and to turn-in your work on a regular basis`

  Hints `Only the project turned in by your project leader will be picked up`
- Remember to discuss on the pool forum!

Evalexpr

- The purpose is to write the `eval_expr` function.
- It must be prototyped like this:

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

- This function take a character string as parameter that represents an arithmetical expression.
  Example:

```
"3 + 42 * (1 - 2 / (3 + 4) - 1 % 21) + 1"
```

- This expression will have to be calculated, and the result returned as return value by the function.
- The string that you will receive will be valid (no bugs, no bad address, no letter nor syntax error, no division by zero...).
- The 5 operators must be supported :
  - `+` for addition
  - `-` for subtraction
  - `/` for division
  - `*` for multiplication
  - `%` for modulo
- The function also has to handle any number of parentheses.
- You must realize a Makefile that permits to generate an executable `eval_expr` with the rules `all, clean, fclean, re` and must not relink.

- Your `main` must be the following one:

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

Hints `If everything went right, your program must return 0`

- We will test in this way:

```
$> make clean
$> make all
$> ./eval_expr '(3+2)*5'
...
$> make fclean
```

That's all Folks!