





Elementary programming Bistromathique

Astek in charge astek_resp@epitech.eu

Abstract: This document is the subject of the Bistromatique Elementary programming project.





Contents

| Ι | Instructions | 2 |
|-----|-------------------|---|
| II | Subject | 3 |
| III | Technical details | 5 |
| IV | Appendices | 6 |





Chapter I

Instructions

- You must be two on this project (and only two)
- Your code must comply with the norm
- You can only use things that you learned in C-Pool
- Only the team leader turn-in will be checked
- The repository shall contain an author file in which you must write your two logins

```
user@host h)cat auteur
```

login_1:login_2

3 (user@host h)

• Turn-in:

Turn-in directory : Bistro



Pay attention to the permissions of your files and directories ...





Chapter II

Subject

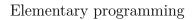
• The goal of the project is to write a program able to display the result of the evaluation of an arithmetic expression composed with integers of infinite size expressed in any base.

This program shall handle the following operators: "+-*/%", and the parenthesis '(' ')'.

It shall handle priorities and syntax errors. All the operations are done with integers: 3/4*4=0.

- usage : ./calc base operators size_read
- Examples :

```
(user@host h)echo | ./calc
      (user@host h)echo "3+6" | ./calc 0123456789 "()+-*/%" 3 ; echo
2
3
      (user@host h)echo "3v6" | ./calc 0123456789 "{}vwxyz" 3 ; echo
4
5
      (user@host h)echo "--++-6(12)" | ./calc 0123456789 "()+-*/%" 10 ; echo
6
7
      syntax error
      (user@host h)echo "--++-6*12" | ./calc 0123456789 "()+-*/%" 9 | cat -e ;
8
      -72
9
      (user@host h)echo "-(12-(4*32))" | ./calc 0123456789 "()+-*/%" 12 | cat -e
10
      116
      (user@host h)
12
      (user@host h)echo "-(12-(4*32)" | ./calc 0123456789 "()+-*/%" 11 | cat -e ;
13
          echo
      syntax error
14
      (user@host h)
      (user@host h)echo "-( @-(;*\!@))" | ./calc "~^@\!;i &[]" "()+-*/%" 13 | cat
16
          -e; echo
17
```









Chapter III

Technical details

- A main.c and a bistromathique.h are given in annex. You just have to code the evalexpr function.
- In case of syntax error, the program displays the string defined by the SYN-TAXE_ERROR_MSG macro.
- You can only use the functions: my_putchar, malloc, free.
- You can ask your questions in the forum, on the B1-C-Prog Elem part.
- All the programs shall be written in C (and comply with the norm)
- There shall be a Makefile, also compliant with norm.
- The executable file must be named 'calc' and be located in the main repository.





Chapter IV

Appendices

```
1 (user@host h)cat main.c
2 /*
3 ** main.c for bistromathique
5 ** Made by Charlie Root
6 ** Login <rn@epita.fr>
8 ** Started on Tue Oct 23 11:45:05 2001 Charlie Root
9 ** Last update Mon Sep 17 12:00:27 2012 Mickael Wiart
10 */
12 #include <stdlib.h>
13 #include <unistd.h>
14 #include <string.h>
#include "bistromathique.h"
17 /*
18 ** Remplacer cette ligne par un include de votre my.h
19 */
20 void my_putstr(char *);
21 int my_strlen(char *);
22 int my_atoi(char *);
24 static void check_base(char *base);
25 static void check_ops(char *ops);
26 static char *get_expr(unsigned size);
28 int main(int ac, char **av)
29 {
30
    char *expr;
    unsigned int size;
31
    if (ac != 4)
34
       my_putstr("Usage : ");
35
       my_putstr(av[0]);
```





```
my_putstr(" base ops\"()+-*/%\" exp_len\n");
37
        exit(1);
38
      }
39
    check_base(av[1]);
40
    check_ops(av[2]);
41
    size = my_atoi(av[3]));
42
    expr = get_expr(size);
    my_putstr(eval_expr(av[1], av[2], expr, size));
44
    return (0);
45
46 }
```





```
1 static void check_base(char *b)
   if (my_strlen(b) < 2)
      {
 4
        my_putstr("Bad base\n");
5
 6
        exit(1);
7
8 }
10 static char *get_expr(unsigned int size)
11 {
    char *expr;
12
13
    if (size <= 0)
14
15
        my_putstr("Bad expr len\n");
16
        exit(2);
17
      }
18
19
    expr = malloc(size+1);
    if (expr == 0)
20
21
        my_putstr("could not alloc\n");
22
        exit(3);
23
24
    if (read(0, expr, size) != size)
25
26
        my_putstr("could not read\n");
27
        exit(4);
28
      }
29
    expr[size] = 0;
30
    return (expr);
31
32 }
33
34 static void check_ops(char *ops)
35 {
    if (my_strlen(ops) != 7)
36
37
        my_putstr("Bad ops\n");
        exit(5);
39
      }
40
41 }
42
43 (user@host h)
44 (user@host h)cat bistromathique.h
45
46 /*
47 ** bistromathique.h for bistromathique in .
49 ** Made by Charlie Root
50 ** Login
```



```
51 **
52 ** Started on Tue Oct 23 11:48:35 2001 Charlie Root
53 ** Last update Tue Oct 23 11:52:38 2001 Charlie Root
55
56 /*
** should be remove if you include stdlib.h (malloc.h does it)
59 void *malloc(unsigned int);
61 #define OP_OPEN_PARENT_IDX O
62 #define OP_CLOSE_PARENT_IDX 1
63 #define OP_PLUS_IDX 2
64 #define OP_SUB_IDX 3
65 #define OP_NEG_IDX 3
66 #define OP_MULT_IDX 4
67 #define OP_DIV_IDX 5
68 #define OP_MOD_IDX 6
70 #define SYNTAXE_ERROR_MSG "syntax error"
71
72 char *eval_expr(char *base,char *ops,char *expr,unsigned int size);
```