



Elementary programming

Bistromathique

Astek in charge astek_resp@epitech.eu

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



Contents

I	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

```
1 (user@host h)cat auteur
2 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 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 :

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



```
18 (user@host h)echo "-(12*(13+15/5*(6/(12+14%(30%5+(10*25)-46)+16)-20)/43)
    *20)*(-(12-98*42)*(16+63-50/3))" | ./calc "0123456789" "()+-*/%" 84 |
    cat -e ; echo
19 -744629760
20 (user@host h)
```



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 `SYNTAXE_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
4 **
5 ** Made by Charlie Root
6 ** Login <rn@epita.fr>
7 **
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 */
11
12 #include <stdlib.h>
13 #include <unistd.h>
14 #include <string.h>
15 #include "bistromathique.h"
16
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 *);
23
24 static void check_base(char *base);
25 static void check_ops(char *ops);
26 static char *get_expr(unsigned size);
27
28 int main(int ac, char **av)
29 {
30     char *expr;
31     unsigned int size;
32
33     if (ac != 4)
34     {
35         my_putstr("Usage : ");
36         my_putstr(av[0]);
```



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




```
1 static void check_base(char *b)
2 {
3     if (my_strlen(b) < 2)
4     {
5         my_putstr("Bad base\n");
6         exit(1);
7     }
8 }
9
10 static char *get_expr(unsigned int size)
11 {
12     char *expr;
13
14     if (size <= 0)
15     {
16         my_putstr("Bad expr len\n");
17         exit(2);
18     }
19     expr = malloc(size+1);
20     if (expr == 0)
21     {
22         my_putstr("could not alloc\n");
23         exit(3);
24     }
25     if (read(0, expr, size) != size)
26     {
27         my_putstr("could not read\n");
28         exit(4);
29     }
30     expr[size] = 0;
31     return (expr);
32 }
33
34 static void check_ops(char *ops)
35 {
36     if (my_strlen(ops) != 7)
37     {
38         my_putstr("Bad ops\n");
39         exit(5);
40     }
41 }
42
43 (user@host h)
44 (user@host h)cat bistromathique.h
45
46 /*
47 ** bistromathique.h for bistromathique in .
48 **
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
54 */
55
56 /*
57 ** should be remove if you include stdlib.h (malloc.h does it)
58 */
59 void *malloc(unsigned int);
60
61 #define OP_OPEN_PARENT_IDX 0
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
69
70 #define SYNTAXE_ERROR_MSG "syntax error"
71
72 char *eval_expr(char *base,char *ops,char *expr,unsigned int size);
```