



Programmation élémentaire

Bistromathique

Responsable Astek astek_resp@epitech.eu



Table des matières

Consignes	2
Sujet	3
Détails techniques	4
Annexes	5



Consignes

- Le projet est à faire en binôme (2 et seulement 2).
- Vos exercices doivent être à la norme.
- Vous ne pouvez utiliser que les éléments vus à la piscine.
- Seul le rendu du chef de groupe sera ramassé.
- Le répertoire doit avoir un fichier auteur dans lequel vous devez mettre les deux logins.

```
1 (user@host h)cat auteur
2 login_1:login_2
3 (user@host h)
```

- Rendu :
svn+ssh ://kscm@koala-rendus.epitech.net/bistromathique-2016-login_x



Attention aux droits de vos fichiers et de vos répertoires



Sujet

- Il s'agit d'écrire un programme capable d'afficher le résultat de l'évaluation d'une expression arithmétique composée d'entiers de taille infinie exprimés dans une base quelconque. Ce programme traite les opérateurs suivant : "+-*/%", ainsi que les parenthèses. '(' ')'. Il gérera les priorités et les erreurs de syntaxe. L'ensemble des opérations est fait sur des entiers : $3/4*4=0$.
- usage : `./calc base opérateurs size_read`

- Exemples :

```
1 (user@host h)echo | ./calc
2 (user@host h)echo "3+6" | ./calc 0123456789 "()+-*/%" 3 ; echo
3 9
4 (user@host h)echo "---++-6(12)" | ./calc 0123456789 "()+-*/%" 10 ; echo
5 syntax error
6 (user@host h)echo "---++-6*12" | ./calc 0123456789 "()+-*/%" 9 | cat -e ; echo
7 -72
8 (user@host h)echo "-(12-(4*32))" | ./calc 0123456789 "()+-*/%" 12 | cat -e ; echo
9 116
10 (user@host h)
11 (user@host h)echo "-(12-(4*32))" | ./calc 0123456789 "()+-*/%" 11 | cat -e ; echo
12 syntax error
13 (user@host h)
14 (user@host h)echo "-( @-(*;!@))" | ./calc "~@\\!;i &[]" "()+-*/%" 13 | cat -e ; echo
15 ^^
16 (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
17 -744629760
18 (user@host h)
```



Détails techniques

- Un `main.c` et un `bistromathique.h` sont donnés dans la partie annexe. Il vous reste à coder la fonction `evalexpr`.
- En cas d'erreur de syntaxe, le programme affiche la chaîne de caractères définie par la macro `SYNTAXE_ERROR_MSG`.
- Vous ne pouvez utiliser que les fonctions : `my_putchar`, `malloc`, `free`.
- Vous pouvez poser vos questions dans le forum rubrique **B1-C-Prog Elem**
- Les programmes doivent être écrits en C (à la norme).
- Il devra y avoir un `Makefile` à la norme.
- L'exécutable doit s'appeler : `'calc'` et se trouver dans le répertoire principal.



Annexes

```
1 (user@host h)cat main.c
2 /*
3  ** main.c for bistromathique in .
4  **
5  ** Made by Charlie Root
6  ** Login
7  **
8  ** Started on Tue Oct 23 11:45:05 2001 Charlie Root
9  %% Last update Mon Apr 4 19:28:32 2011 guillaume bonetti
10 */
11
12 #include "bistromathique.h"
13
14 static void check_base(char *base);
15 static void check_ops(char *ops);
16 static char *get_expr(unsigned size);
17
18 int main(int ac,char **av)
19 {
20     char *expr;
21     unsigned int size;
22
23     if (ac != 4)
24     {
25         my_putstr("Usage : ");
26         my_putstr(av[0]);
27         my_putstr(" base ops\"()+-*/%\" exp_len\n");
28         exit(1);
29     }
30     check_base(av[1]);
31     check_ops(av[2]);
32     expr = get_expr(size = atoi(av[3]));
33     my_putstr(eval_expr(av[1],av[2],expr,size));
34 }
```



```

1 static void check_base(char *b)
2 {
3     if (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(1);
18     }
19     if ((expr = malloc(size+1))==0)
20     {
21         my_putstr("could not alloc\n");
22         exit(1);
23     }
24     if (read(0,expr,size)!=size)
25     {
26         my_putstr("could not read\n");
27         exit(1);
28     }
29     expr[size] = 0;
30     return (expr);
31 }
32
33 static void check_ops(char *ops)
34 {
35     if (strlen(ops) != 7)
36     {
37         my_putstr("Bad ops\n");
38         exit(1);
39     }
40 }
41 (user@host h)
42 (user@host h)cat bistromathique.h
43
44 /*
45 ** bistromatique.h for bistromathique in .
46 **
47 ** Made by Charlie Root
48 ** Login
49 **
50 ** Started on Tue Oct 23 11:48:35 2001 Charlie Root
51 ** Last update Tue Oct 23 11:52:38 2001 Charlie Root
52 */
53
54 /*
55 ** should be remove if you include stdlib.h (malloc.h does it)
56 */
57 void *malloc(unsigned int);
58
59 #define OP_OPEN_PARENT_IDX 0
60 #define OP_CLOSE_PARENT_IDX 1
61 #define OP_PLUS_IDX 2
62 #define OP_SUB_IDX 3
63 #define OP_NEG_IDX 3
64 #define OP_MULT_IDX 4
65 #define OP_DIV_IDX 5
66 #define OP_MOD_IDX 6
67
68 #define SYNTAXE_ERROR_MSG "syntax error"
69
70 char *eval_expr(char *base,char *ops,char *expr,unsigned int size);

```