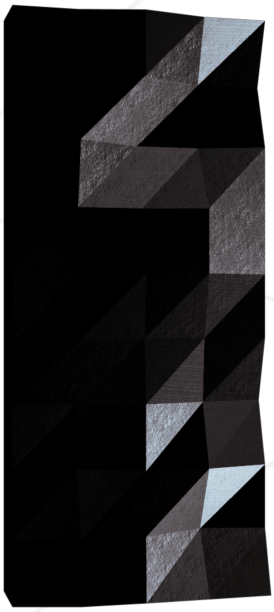

Bistromathique

n.f. : branche relativiste des mathémathiques

42 staff staff@42.fr
ol ol@42.fr
marvin marvin@42.fr

Résumé: Logique gastronomique italienne au service de l'espace-temps

Table des matières

Chapitre I

Préambule

Voilà ce que le `Guide du Voyageur Galactique` a à dire à propos de la Bistromathique :

" The Bistromathic Drive is a wonderful new method of crossing vast interstellar distances without all the dangerous mucking about with Improbability Factors. Bistromathics itself is simply a revolutionary new way of understanding the behaviour of numbers. Just as Einstein observed that time was not an absolute but depended on the observer's movement in space, and that space was not an absolute, but depended on the observer's movement in time, it is now realized that numbers are not absolute, but depended on the observer's movement in restaurants.

" The first non-absolute number is the number of people for whom the table is reserved. This will vary during the course of the first three telephone calls to the restaurant, and then bear no apparent relation to the number of people who actually turn up, or to the number of people who subsequently join them after the show/match/party/gig, or to the number of people who leave when they see who else has turned up. The second non-absolute number is the given time of arrival, which is now known to be one of those most bizarre of the mathematical concepts, a recipriversexcluson, a number whose existence can only be defined as being anything, other than itself. In other words, the given time of arrival is the one moment of time at which it is impossible that any member of the party will arrive.

" Recipriversexclusons now play a vital part in many branches of math, including statistics and accountancy and also form the basic equations used to engineer the Somebody Else's Problem field. The third and most mysterious piece of non-absoluteness of all lies in the relationship between the number of items on the check, the cost of each item, the number of people at the table and what they are each prepared to pay for. (The number of people who actually brought any money is only a sub-phenomenon in this field.) The baffling discrepancies that used to occur at this point remained uninvestigated for centuries simply because no one took them seriously. They were at the time put down to such things as politeness, rudeness, meanness, flashiness, tiredness, emotionality or the lateness of the hour, and completely forgotten about on the following morning. They were never tested under laboratory conditions, of course, because they never occurred in laboratories - not in reputable laboratories at least.

" And so it was only with the advent of pocket computers that the startling truth became finally apparent, and it was this : Numbers written on restaurant checks within the confines of restaurants do not follow the same mathematical laws as numbers written

on any other pieces of paper in any other parts of the Universe. This single statement took the scientific world by storm. It completely revolutionized it. So many mathematical conferences got held in such good restaurants that many of the finest minds of a generation died of obesity and heart failure and the science of math was put back by years. Slowly, however, the implications of the idea began to be understood. To begin with it had been too stark, too crazy, too much like what the man in the street would have said "Oh, yes, I could have told you that." Then some phrases like "Interactive Subjectivity Frameworks" were invented, and everybody was able to relax and get on with it.

" The small groups of monks who had taken up hanging around the major research institutes singing strange chants to the effect that the Universe was only a figment of its own imagination were eventually given a street theater grant and went away."

Non, vous n'avez pas besoin d'aller dans un bistro pour réussir votre bistromathique.

Chapitre II

Consignes

- Votre projet doit être à la Norme.
- Vous ne pouvez utiliser que les éléments vus durant votre Piscine.
- Le répertoire doit avoir un fichier auteur dans lequel vous devez mettre vos logins.

```
$>cat auteur
login_1:login_2
$>
```

Chapitre III

Sujet

Ce projet consiste en la réalisation d'un programme appelé Bistromathique.

- La Bistromathique est capable d'afficher le résultat de l'évaluation d'une expression arithmétique composée d'entiers de taille indéterminée exprimés dans une base quelconque.
- Elle traite les opérateurs suivants : "+-*/%".
- Elle permet l'emploi de parenthèses : "(" et ")".
- Elle gère les priorités et les erreurs de syntaxe.
- L'ensemble des opérations de la Bistromathique est fait sur des entiers. Ainsi, 3/4*4 est égal à 0.
- L'usage de la Bistromathique est : `usage : ./calc base opérateurs size_read`
- L'argument opérateurs sera tel que le charactère parenthèse ouvrante soit en premier, suivi de la parenthèse fermante, le plus, le moins, le multiplier, le diviser et pour finir le modulo
- Si un opérateur est dans la base, il faudra considérer cela comme une erreur.

- Exemples de comportement :

```
$>echo  | ./calc
syntax error
$>echo "3+6" | ./calc 0123456789 "()+-*/%" 3
9
$>echo "--++-6(12)" | ./calc 0123456789 "()+-*/%" 10
syntax error
$>echo "--++-6*12" | ./calc 0123456789 "()+-*/%" 9 | cat -e
-72$
$>echo "-(12-(4*32))" | ./calc 0123456789 "()+-*/%" 12 | cat -e
116$
$>echo "1234567890*9876543210" | ./calc 0123456789 "()+-*/%" 21 | cat -e
121932631112635269 00$
$>echo "-(12-(4*32)"  | ./calc 0123456789 "()+-*/%" 11 | cat -e
syntax error$
$>echo "+(~^-~^^^^)*^^~^~^"  | ./calc "~^" "()-+*/%" 18 | cat -e
+^^~^~^~~~~$
$>echo "^&*;-i&"  | ./calc '~^@!;i &[]' "()+-*/%" 7 | cat -e
^^$
$>echo "-(12*(13+15/5*(6/(12+14%(30%5+(10*25)-46)+16)-20)/43)*20)*(-(12-98*42)*(16+63-50/3))"  | ./
 calc "0123456789" "()+-*/%" 84 | cat -e
-744629760$
$>
```

Chapitre IV

Détails techniques

- Un fichier `bistromathique.h` et un fichier `main.c` sont donnés dans la partie Annexes. Il vous reste à coder la fonction `eval_expr`.
- En cas d'erreur de syntaxe, le programme affiche la chaîne de caractères définie par la macro `SYNTAXE_ERROR_MSG`.
- En cas d'erreur votre fonction `eval_expr` doit aussi renvoyer une valeur différente de 0
- Vous ne pouvez utiliser que les fonctions `exit`, `write`, `read`, `malloc` et `free`.
- Vous pouvez poser vos questions dans le forum.
- Vous pouvez refaire votre main.(attention il doit fonctionner comme celui de la partie Annexes)
- Les programmes doivent être écrits en C à la Norme.
- Un Makefile à la Norme doit également être fourni.
- L'exécutable doit s'appeller `calc` et se trouver dans le répertoire principal.

Chapitre V

Notation

La notation de la Bistromathique s'effectue en deux temps.

- Nous commençons par tester `las funcionalidades` (sur 10 points). Votre programme doit passer les tests suivants. La correction s'arrête au premier tier de fonctionnalités incorrect.
  - Addition, soustraction et multiplication de deux opérandes de taille arbitraire en base 10.
  - Addition, soustraction et multiplication de deux opérandes de taille arbitraire en base `n`.
  - Addition, soustraction et multiplication de `n` opérandes de taille arbitraire en base `n` et gérant les priorités des opérateurs.
  - Addition, soustraction et multiplication de `n` opérandes de taille arbitraire en base `n` et gérant les priorités des opérateurs et des parenthèses.
  - Addition, soustraction, multiplication, division et modulo de `n` opérandes de taille arbitraire en base `n` et gérant les priorités des opérateurs et des parenthèses.
- La deuxième partie est `la optimización` du code (sur 10 points).
  - Elle n'est vérifiée que si toute la première partie est correcte.
  - Toutes les Bistromathiques ayant validé l'intégralité de la première partie seront en compétition sur un test de rapidité.
  - Le projet étant le plus rapide remportera l'intégralité des points de cette partie. Combiné à la partie technique, ce projet remportera donc la note de 20/20.
  - Les projets suivants remporteront des points en fonction de leur classement : les plus rapides auront plus de points que les plus lents.

Bon courage à tous !

Chapitre VI

Annexes

- bistromathique.h

```
#ifndef FT_BISTROMATHIQUE_H
# define FT_BISTROMATHIQUE_H

# define SYNTAXE_ERROR_MSG       "syntax error\n"

int     eval_expr(char *base, char *ops, char *expr);

#endif
```

- main.c

```
#include <stdlib.h>
#include <sys/types.h>
#include <sys/uio.h>
#include <unistd.h>
#include "bistromathique.h"

static size_t       ft_strlen(char *strlen)
{
    size_t  i;

    i = 0;
    while (strlen[i] != 0)
        i = i + 1;
    return (i);
}

static unsigned int ft_uatoi(char *nbr)
{
    unsigned int    res;

    res = 0;
    while (*nbr)
    {
        res = *nbr - '0' + res * 10;
        nbr = nbr + 1;
    }
    return (res);
}

static int          ft_fill_buff(char* buf, size_t size)
{
    size_t      size_read;
    ssize_t     ret;

    size_read = 0;
    ret = read(0, buf, size);
    if (ret == -1)
    {
                write(2, SYNTAXE_ERROR_MSG, ft_strlen(SYNTAXE_ERROR_MSG));
```

```
                return (1);
        }
    while (ret != 0)
    {
        size_read = size_read + ret;
        ret = read(0, buf + size_read, size - size_read);
        if (ret == -1)
        {
                    write(2, SYNTAXE_ERROR_MSG, ft_strlen(SYNTAXE_ERROR_MSG));
                    return (1);
            }
    }
    buf[size_read] = 0;
    return (0);
}

static int          ft_check_args_n_malloc_buf(int argc, char **argv, char **buf)
{
    size_t  size;

    if (argc != 4)
    {
        write(2, SYNTAXE_ERROR_MSG, ft_strlen(SYNTAXE_ERROR_MSG));
        return (1);
    }
    size = ft_uatoi(argv[3]);
    *buf = malloc(sizeof(**buf) * (size + 1));
    if (*buf == 0)
    {
        write(2, SYNTAXE_ERROR_MSG, ft_strlen(SYNTAXE_ERROR_MSG));
        return (1);
    }
    return (ft_fill_buff(*buf, size));
}

int                 main(int argc, char **argv)
{
    char        *buf;

    if (ft_check_args_n_malloc_buf(argc, argv, &buf))
        return (1);
    return (eval_expr(argv[1], argv[2], buf));
}
```