

Tutoriel: passage d'arguments à main() avec conversion de chaînes à-la-C

Ecrire un programme **som** qui affiche son nom suivi de la somme de ses arguments.

Exemple:

```
./som 9 12.5 -5.
```

affiche :

```
./som 16.5
```

Pour cela on va utiliser la forme générale de la fonction **main** :

```
int main(int argc, char* argv[])
```

On utilise traditionnellement les noms d'arguments formels **argc** et **argv** :

- **argc** est le nombre de chaînes transmises, il est au moins égal à 1 car le *nom de l'exécutable* est toujours transmis comme première chaîne.
- **argv** est un tableau de **argc** pointeurs sur des **char**

Chaque **argv[i]** pointe sur une chaîne à-la-C transmise à main().

argv[0] existe toujours car c'est le nom de l'exécutable. Dans l'exemple précédent, le nom de l'exécutable est la chaîne **"./som"**.

L'instruction suivante : **cout << argv[0]+2 ;**

Affiche **som** car on décale l'adresse indiquée par argv[0] de 2 caractères après le début de la chaîne argv[0] pour ne pas imprimer « ./ ».

Chaque **argv[i]** est donc l'adresse du début d'un tableau de caractères. Si on veut accéder à un caractère particulier de ce tableau, on peut utiliser l'indirection sur une adresse ou un second indice (recommandé car plus intuitif au début):

L'instruction suivante :

```
cout << *(argv[0]+2) << " " << argv[0][4] << endl ;
```

Affiche :

```
s m
```

Pour le problème à résoudre ici, il faut convertir les chaînes de caractères transmises à la suite du nom de l'exécutable dans un type numérique, par exemple en nombres à virgule flottante avant de pouvoir les utiliser pour des calculs. En effet on ne peut pas faire d'opération numérique directement sur les chaînes de caractères.

Pour cette conversion **chaîne -> nombre**, nous recommandons d'utiliser **stod** car cette fonction permet de détecter les erreurs: elle met à jour le nombre de caractères correctement lus.

Il faut ajouter : **include <cstring>** pour disposer de la fonction **strlen()**

Voici un exemple de conversion de la chaîne à-la-C **argv[1]** vers une valeur numérique de type double:

```
double val(0) ;
size_t nb(0) ;

val = stod(argv[1], &nb);
```

Si, après l'affectation, **nb** est différent de de la longueur de la chaîne donné par **strlen(argv[1])** il faut signaler une erreur.

```
#include <iostream>
#include <cstring>

Using namespace std;

int main(int argc, char* argv[])
{
    cout << argv[0]+2 ;

    if(argc >1)
    {
        double somme(0);
        size_t nb(0);
        for(argc--; argc > 0 ; argc--)
        {
            somme += stod(argv[argc],&nb);

            if(nb != strlen(argv[argc]))
            {
                cout << " Erreur chaine non numérique" << endl;
                return EXIT_FAILURE;
            }
        }
        cout << " " << somme ;
    }
    cout << endl;

    return EXIT_SUCCESS;
}
```