

BOUCLES ET POINTEURS EN Y86

Exercice 1 : Boucle

Réalisez l'équivalent du code C suivant :

```
long sum = 0, i;  
for (i = 1; i <= 10; i++)  
    sum += i;
```

Exercice 2 : Multiplication par une constante

Réalisez l'équivalent du code C suivant :

```
long a = 7, b;  
  
b = 13 * a;
```

Rappels

- Le processeur Y86 dispose des trois modes d'adressage suivants :
 - immédiat : l'adresse mémoire à laquelle lire ou écrire est définie dans l'instruction elle-même. C'est donc une constante ;
 - indirect : l'adresse mémoire est contenue dans un registre, dont le numéro est défini dans l'instruction. On la note « **(%reg)** » ;
 - indexé : l'adresse mémoire est construite à la volée en additionnant une constante immédiate à l'adresse contenue dans un registre dont le numéro est lui aussi défini dans l'instruction. On la note « **cste(%reg)** ».
- Toute étiquette représente l'adresse, constante, de l'instruction ou de la directive située juste après cette étiquette.

Exercice 3 : Pointeur

Réalisez l'équivalent du code C suivant :

```
long a, b = 2, x = 3, *p;  
p = &x;  
...  
a = *p;  
*p = b;  
p = &b;  
...  
(*p) ++;
```

Pour des raisons d'efficacité, ne définissez pas d'emplacement mémoire pour `p`, mais utilisez le registre `%esi` à la place.

Exercice 4 : Parcours de tableau

Réalisez l'équivalent du code C suivant :

```
long t[4] = { 1, 2, 3, 4 };
long n = 4;

for (p = &t[0]; n > 0; p ++, n --)
    (*p) ++;
```

Pour des raisons d'efficacité, ne définissez pas d'emplacement mémoire pour `p` et `n`, mais utilisez les registres `%esi` et `%ecx` à la place.

Pour aller plus loin...

Exercice 5 : Écriture binaire

Soit un tableau de 32 entiers $p_{31}, p_{30}, p_{29} \dots p_1, p_0$. Écrire un programme qui calcule dans un registre l'entier p dont les chiffres de sa représentation binaire sont les p_i , i.e. qui convertit depuis la représentation binaire.

Conseil : chaque fois qu'on lit un nouvel élément p_i , remplacer p par $2 * p + p_i$.

Comment traiter le cas où les entiers seraient plutôt dans l'ordre $p_0, p_1, \dots, p_{30}, p_{31}$?

Exercice 6 : strcpy

Réalisez l'équivalent du code C suivant :

```
long t[] = { 1, 2, 3, 4, 0, 0, 0, 0, 0, 0 };
long *p, *q, n = 5;

p = t;
q = t + n;
do
    *(q++) = *p;
while (*(p++) != 0);
```