*Licence d’Informatique*

TD d’IP : Feuille 6

# Visibilité, allocation, récursivité

**Exercice 1 :** On considère la fonction suivante, calculant la somme des $n$ premiers entiers naturels.

```
int somme(int n) {
  int resultat = 0;
  int i;

  for (i = 0; i <= n; i++ )
    resultat += i;

  return resultat;
}
```

Modifier ce code, sans rajouter de variable supplémentaire, de sorte que si on appelle `somme` plusieurs fois consécutivement avec le même paramètre $n$, seul le premier appel nécessite de calculer le résultat.

---

**Exercice 2 :** Ecrire une commande `int interpreter(char *argv[])` permattent d’interpréter des expressions respectant la syntaxe suivante :

```
<expression> = + <expression> <expression>
<expression> = x <expression> <expression>
<expression> = - <expression>
<expression> = [0-9]+
```

---

**Exercice 3 :** Ecrivez une fonction `char *lire_chaine()` stockant dans un tableau de caractères une suite de caractères tapés au clavier et se finissant par `\n`, et renvoyant ce tableau comme résultat. On rappelle que la fonction `int getchar()` retourne le premier caractère entré par l’utilisateur.

---

**Exercice 4 :**

```
#include <stdio.h>

char v1[]   = {’a’,’b’,’c’,’d’,’e’};
char v2[]   = "abcde";
char v3[10] = {’a’,’b’,’c’,’d’,’e’};
char v4[10] = "abcde";
char *p     = "abcde";

main() {
  printf("Taille de v1 : %d\n", sizeof(v1));
  printf("Taille de v2 : %d\n", sizeof(v2));
```

```
  printf("Taille de v3 : %d\n", sizeof(v3));
  printf("Taille de v4 : %d\n", sizeof(v4));
  printf("Taille de p  : %d\n", sizeof(p));
  printf("Taille de *p : %d\n", sizeof(*p));

  v2[0] = 'A';
  printf("Modification de v2\n");
  *p = 'A';
  printf("Modification de p\n");
}
```

Quel sera le résultat de l'exécution du programme précédent ? Qu'en déduisez-vous concernant l'utilisation de pointeurs ?

---

**Exercice 5 :** On rappelle que la mémoire est divisée en mots et que lors d'une allocation, la zone allouée est alignée sur un mot-mémoire. De plus, au début de la zone allouée, on trouve un en-tête enregistrant la taille du bloc alloué. Quel sera le résultat de l'exécution de la commande suivante ?

```
#include <stdio.h>

main() {
  char *p = calloc(1, sizeof(int));
  printf(``Adresse initiale : %d\n'', (int) p);
  p = realloc(p, 256);
  printf(``- extension 1 : %d\n'', (int) p);
  p++;
  p = realloc(p, 1024);
  printf(``- extension 2 : %d\n'', (int) p);
}
```

En quoi l'en-tête en début de la zone allouée est-il nécessaire à une gestion correcte de la mémoire ?

---