

Séance 4

Tableaux; Chaînes de caractères;

Déclaration d'un tableau:

```
int tab[100];
```

Utilisation:

```
int i;  
tab[i] = 4;  
int x = tab[j];
```

tab est un pointeur (`int*`) sur la première case du tableau.

```
int * t = tab;  
t++;  
t += i;  
*(t+i) est un contenu de case de tableau, identique à t[i]  
&t[i] est une adresse de case de tableau, identique à t+i
```

Allocation dynamique de tableau:

```
calloc, malloc, free
```

Chaîne de caractères:

```
char s[100];  
char * ps;
```

Fonctions classiques C de gestion de chaîne de caractères:

```
strlen, strcpy, strcat, strcmp
```

Fonctions de conversions

```
atoi, atof,  
sprintf, sscanf,  
toupper, tolower
```

Recopie « rapide » d'une zone mémoire dans une autre:

```
memcpy
```

Exercices

1) Ecrire un programme qui déclare un tableau d'entiers de taille `TAILLE` (définie par un `#define`), qui demande à l'utilisateur une valeur pour chaque case du tableau, et qui affiche le tableau.

2) Soit `P` un pointeur sur un tableau `A`:

```
int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};
int * P = A;
```

Quelles valeurs ou adresses fournissent ces expressions ?

<code>*P+2</code>	<code>*(P+2)</code>	<code>&P+1</code>	<code>&A[4]-3</code>
<code>A+3</code>	<code>&A[7]-P</code>	<code>P+(*P-10)</code>	<code>*(P+*(P+8)-A[7])</code>

3) Dans la suite d'instructions suivantes, relevez les erreurs, puis afficher les résultats des `printf`.

```
char * ptab;          char tab[32];
char ch1[]="Bonjour"; char ch2[15];
tab = "QW";          ch2 = ch1;
strcpy(ptab, "ASDF");
printf("tab: %s      ptab: %s\n", tab, ptab);
printf("tab: %c      ptab: %c\n", tab, ptab);
printf("tab: %c      ptab: %c\n", tab[1], ptab[1]);
printf("tab: %c      ptab: %c\n", *(tab+1), *(ptab+1));
printf("tab: %c      ptab: %c\n", *tab+1, *ptab+1);
```

4) Ecrire le programme `formalismeTableau.c`, écrit en formalisme tableau, en formalisme pointeur.

5) Ecrire un programme qui lit une chaîne de caractères et écrit la chaîne de caractères en majuscules.

6) Ecrire un programme qui transforme un nombre écrit en chiffres romains en un nombre écrit en chiffres arabes.

7) Ecrire un programme utilisant l'allocation dynamique de mémoire et effectuant les traitements suivants:

- Déclarer une chaîne de caractères statique `ch` et initialiser `ch` avec la chaîne suivante : "voila une chaine de caracteres".
- Compter le nombre de `a` dans `ch` et stocker les adresses de chacun d'eux dans un tableau d'adresses `char **adr`. Stocker également leur position dans un tableau d'entiers `int *pos`.
- Supprimer tous les `a` dans `ch` en utilisant `adr` et `pos` et afficher la nouvelle chaîne `ch`.
- Afficher en ordre inverse (en partant du `s` final) et en majuscule `ch`.
- Libération mémoire.

