

## Algorithmes et structures de données : TD 2 Corrigé

### Tableaux

#### Exercice 2.1 *Occupation de la mémoire*

Considérer les déclarations et les affectations de variables suivantes :

```
{ Declaration des variables}
var c : char;
var i : integer;
var j,k : integer;
var tableau : array[1..4] of byte;

begin
  { Affectation des variables}
  c := 65;
  i := 10;
  j := 300;
  k := 65540;
  tableau[1] := 10;
  tableau[2] := 12;
  tableau[3] := 20;
  tableau[4] := 5;

end;
```

1. Combien d'octets occupent ces variables dans la mémoire vive ?

Ce tableaux occupe  $4*1+4*4=20$  octets car il y a 4 éléments dans le tableau et chaque entrée de type `byte` occupe 1 octet et car il y a 4 variables de type `integer` qui occupe 4 octets chacune.

2. Ebaucher l'occupation de la mémoire d'un ordinateur avec 32 Octets de mémoire vive (les adresses sont à choisir arbitrairement).

#### Exercice 2.2 *Tableau*

Considérer le tableau suivant :

```
var tableau : array[1..100] of integer;
```

1. Combien d'octets occupe ce tableau de 100 éléments dans la mémoire vive ?

400 octets car un `integer` occupe 4 octets.

2. Dans un ordianteur avec 1 Kilo-octet (Ko) de mémoire vive, combien d'éléments de type `integer` peut-on stocker dans ce tableau ?

	#31	
tableau[4] #19	5	
tableau[3] #18	20	
tableau[2] #17	12	
tableau[1] #16	10	
k #15	0	
k #14	1	
k #13	0	
k #12	4	
j #11	0	
j #10	0	
j #9	1	
j #8	44	
i #7	0	
i #6	0	
i #5	0	
i #4	10	
i #3		
i #2		
i #1		
c #0	65	

$1024/4 = 256$  éléments.

3. Dans un ordianteur avec 512 Mega-octets (Mo) de mémoire vive, combien d'éléments de type `integer` peut-on stocker dans ce tableau ?

$$512 * 1024 * 1024 / 4 = 128 * 1024 * 1024 = 134217728 \text{ éléments.}$$

**Exercice 2.3** Considérer le tableau à une dimension de taille 12 qui stocke le nombre de jours pour le mois correspondant :

```
{ Declaration des variables}
```

```
var jours : array[1..12] of byte;
```

```
{ Affectation des variables}
```

```
jours[1] := 31;
```

```

jours[2] := 28;

```

• •

```
jours[12] := 31;
```

1. Combien d'octets occupe ce tableau dans la mémoire vive ?

Ce tableaux occupe  $12 \times 1 = 12$  octets car il y a 12 éléments dans le tableau et chaque entrée de type `byte` occupe 1 octet.

2. Ecrivez un algorithme qui utilise ce tableau et qui détermine dans quel mois tombe le 245ème jours de l'année.

```
somme := 0;
mois := 1;

tant que somme < 245 faire
    somme := somme + jours[mois];
    mois := mois + 1;
fin tant que
afficher "Le 245eme jours tombe dans le mois numero :"
afficher mois-1;
```

3. Faites tourner cet algorithme dans un tableau.

somme	mois
0	
	1
31	
	2
59	
	3
90	
	4
120	
	5
151	
	6
181	
	7
212	
	8
243	
	9
273	
	10

4. Ebaucher l'occupation de la mémoire vive de votre algorithme après la déclaration des variables pour un ordinateur avec 1 kilo-octet (1 ko) de mémoire vive (les adresses sont à choisir arbitrairement).

## Exercice 2.4 *Tableau*

1. Déclarez une variable d'un tableau qui permet de stocker le montant de pluie en millimètres qui est tombé pour le mois de Juillet (Démarche : Combien de jours y-a-t-il dans le mois de juillet ? Avec quel type (de base) peut-on encoder un montant en millimètres ?)

```
type t_tableau = array[1..31] of integer; {seulement de nombres entiers en millimetres}
var pluie_juillet : t_tableau;
```

2. Combien d'octets occupe ce tableau dans la mémoire vive ?

Ce tableaux occupe  $31 \times 4 = 124$  octets car il y a 31 éléments dans le tableau et chaque entrée de type `integer` occupe 4 octets.

3. Ecrire un algorithme qui compte le nombre de jours de pluie dans le mois de juillet.

```
nombre_jours := 0;
jour := 1;

tant que jour <= 31 faire
    si pluie_juillet[jour] > 0 alors
        nombre_jours := nombre_jours + 1;
    fin si
    jour := jour + 1;
fin tant que
afficher "Dans le mois de juillet, il y avait nombres de jours de pluie: "
afficher nombre_jours;
```

4. Ecrire un algorithme qui calcule la moyenne du montant de millimètres de pluie par jour dans le mois de juillet.

Seulement la dernière ligne de l'algorithme change

```
montant_pluie := 0;
jour := 1;

tant que jour <= 31 faire
    montant_pluie := montant_pluie + pluie_juillet[jour];
    jour := jour + 1;
fin tant que

moyenne_plui = montant_pluie/31;

afficher "Dans le mois de juillet, il y avait en moyenne de pluie en mm: "
afficher moyenne_pluie;
```

5. Ecrire un algorithme qui détermine le jour avec le montant maximum de pluie du mois de juillet.

```
pluie_max := pluie_juillet[1];
jour_max := 1;
jour := 2;
tant que jour <= 31 faire
    si pluie_juillet[jour] > pluie_max alors
        pluie_max := pluie_juillet[jour]
        jour_max := jour;
    fin si
```

```
        jour := jour + 1;  
fin tant que  
afficher "Le jour en juillet avec le max de pluie c etait: "  
afficher jour_max;
```

Les exercices sont à rendre dans le prochain TD.