

## Chapitre 10 : Boucles imbriquées - TD

### Exercice 1 : Puissance d'un circuit \*

Écrire un programme qui permet d'afficher la puissance d'un circuit en fonction de la tension  $u$  variant par pas de 0.5V et de l'intensité  $i$  variant par pas de 0.05A sachant que les valeurs initiales et finales de  $u$  et  $i$  seront demandées à l'utilisateur.

**Exemple d'affichage :** (pour  $u$  variant de 2 à 3V et  $i$  variant de 0.15 à 0.3A)

```
u=2.0 i=0.15 p=0.3
u=2.0 i=0.2 p=0.4
u=2.0 i=0.25 p=0.5
u=2.0 i=0.3 p=0.6
```

```
u=2.5 i=0.15 p=0.38
u=2.5 i=0.2 p=0.5
u=2.5 i=0.25 p=0.62
u=2.5 i=0.3 p=0.75
```

```
u=3.0 i=0.15 p=0.45
u=3.0 i=0.2 p=0.6
u=3.0 i=0.25 p=0.75
u=3.0 i=0.3 p=0.9
```

### Exercice 2 : Saisie de valeurs positives dans un tableau \*

Définir une fonction `saisieValPositives` prenant en paramètre un tableau d'entiers  $t$  et un nombre d'éléments `nbElements` à saisir, et ajoutant au tableau `nbElements` valeurs positives saisies par l'utilisateur. La saisie de chaque valeur sera répétée jusqu'à ce que la valeur donnée soit positive. Tester la la fonction avec comme tableau initial  $t = []$  et 3 éléments à saisir.

### Exercice 3 : Calcul des puissances entières avec des additions \*\*

Ecrire une fonction permettant de calculer  $x^y$  où  $x$  et  $y$  sont des entiers positifs. Seule l'addition est autorisée.

Le principe de l'algorithme est de considérer que  $x^y$  est équivalent à  $x * x * x * x \dots$  ( $y$  fois) et qu'une multiplication ( $z * x$ ) peut s'écrire  $z + z + z + z \dots$  ( $x$  fois). Il faut donc deux boucles imbriquées: une qui permet de faire  $y$  fois la multiplication, chaque multiplication étant elle-même une boucle de  $x$  additions. Cependant, ce qui doit être additionné  $x$  fois dans la boucle interne évolue. Au premier passage, c'est 1 qu'on additionne  $x$  fois ; au second passage, c'est  $x$  qu'on additionne  $x$  fois ; au troisième,

c'est  $x^2$ , etc. Il faut donc mémoriser le résultat de la multiplication (variable auxiliaire puissance) précédente pour faire la multiplication suivante.

## Pour aller plus loin

### Exercice 4 : Nombre de façons de rendre la monnaie \*\*

#### Question 1 : Nombre de façons de rendre 1€ avec de la monnaie

Ecrire un programme qui affiche toutes les manières possibles d'obtenir 1 € avec des pièces de 2 centimes, 5 centimes et 10 centimes. On veillera à ne pas afficher les 0 ( $20*5 = 100$  plutôt que  $0*2+20*5+0*10=100$ ). Le programme calculera et affichera également le nombre de manières d'obtenir ainsi 1 €.

#### Question 2 : Nombre de façons de rendre n'importe quelle somme avec de la monnaie

Modifier le programme précédent pour qu'il affiche le nombre de manières possibles et toutes ces manières d'obtenir une somme avec des pièces de 2 centimes, 5 centimes et 10 centimes. La somme (en centimes) sera saisie par l'utilisateur.

### Exercice 5 : Tri à bulles \*\*\*

On souhaite trier un tableau d'entiers à l'aide de l'algorithme du tri à bulles. Cet algorithme fonctionne de la manière suivante. Pour toute case du tableau sauf la dernière, on compare la valeur de cette case avec celle de sa voisine de droite. Si sa valeur est supérieure à celle de sa voisine, on échange les deux valeurs. On recommence ceci tant que l'on a échangé au moins deux valeurs.

#### Question 1 : Exemple

Appliquer le tri à bulles sur le tableau [3, 2, 1, 5, 7, 4, 8, 6].

#### Question 2 : Programme

Ecrire un programme implantant le tri à bulles.

#### Question 3 : Découpage en fonction

Découper l'algorithme précédent en fonctions.