
SÉANCE 2



Objectif

Le but de cette deuxième séance est d'utiliser les expressions logiques ainsi que les instructions de contrôle permettant de faire des choix et des itérations. Ces outils sont illustrés par des situations classiques : vérification partielle de la validité des données (exercices 2.2 et 3.2), traitement de séries de données dont le nombre est connu (exercice 3.1) ou dont la fin est marquée par une donnée particulière (exercice 2.1).



Exercices

✎ Exercice 1 (Années bissextiles)

Écrivez un programme qui affiche à l'écran un message indiquant si une année saisie au clavier est bissextile. Une année est bissextile si son expression numérale est divisible par 4 mais pas par 100, ou si elle est divisible par 400.

Exemples :

- 1996 est bissextile (multiple de 4, mais pas de 100) ;
- 2000 est bissextile (multiple de 4, de 100, mais aussi de 400) ;
- 2100 n'est pas bissextile (multiple de 4, de 100, mais pas de 400).

✎ Exercice 2 (Comptage de caractères)

1. Écrivez un programme qui lit un texte tapé au clavier se terminant par le caractère # et affiche à l'écran le nombre de lettres et de voyelles qu'il contient.

On suppose que le texte ne contient que des lettres minuscules non accentuées, que les mots ne sont séparés que par des espaces, qu'il n'y a aucune ponctuation et qu'il ne contient pas de caractère « saut de ligne » (« *new line* »).

2. Modifiez votre programme afin de vérifier que chaque caractère saisi par l'utilisateur est bien une lettre minuscule non accentuée ou un espace. Dans le cas contraire, avant d'être interrompu, le programme doit afficher un message indiquant le caractère erroné et son code.

✎ Exercice 3 (Calcul de la moyenne)

1. En utilisant une boucle `for`, écrire un programme qui affiche à l'écran la moyenne d'une série de valeurs réelles. L'utilisateur donnera le nombre de valeurs dont il veut calculer la moyenne suivi des valeurs.
2. Modifiez votre programme afin de considérer uniquement des notes. Pour cela, vous vérifierez que chaque note est comprise entre 0 et 20. Si ce n'est pas le cas, demandez à l'utilisateur de saisir de nouveau la note tant qu'elle n'est pas comprise entre 0 et 20.

✎ Exercice 4 (Triangle d'étoiles)

Écrivez un programme qui affiche à l'écran un triangle isocèle constitué de n lignes. Ce nombre de lignes doit être demandé à l'utilisateur. Par exemple, pour $n = 5$, l'affichage obtenu doit être le suivant (la première étoile est précédée de $n - 1 = 4$ espaces) :

```
      *
     ***
    *****
   ********
  *********
 *****
```

✎ Exercice 5 (Table de multiplication)

1. Écrivez un programme qui affiche à l'écran la table de multiplication de deux entiers compris entre 0 et 10 sous la forme suivante :

x	0	1	2	3	4	5	6	7	8	9	10
<hr/>											
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10
2	0	2	4	6	8	10	12	14	16	18	20
3	0	3	6	9	12	15	18	21	24	27	30
4	0	4	8	12	16	20	24	28	32	36	40
5	0	5	10	15	20	25	30	35	40	45	50
6	0	6	12	18	24	30	36	42	48	54	60
7	0	7	14	21	28	35	42	49	56	63	70
8	0	8	16	24	32	40	48	56	64	72	80
9	0	9	18	27	36	45	54	63	72	81	90
10	0	10	20	30	40	50	60	70	80	90	100

Vous prendrez soin d'obtenir exactement l'affichage ci-dessus. Afin d'aligner les nombres, vous pouvez utiliser un gabarit d'affichage dans le spécificateur de format. Par exemple, le format `%3d` permet d'afficher un entier sur au moins 3 caractères, en complétant d'espaces à gauche, si nécessaire (alignement à droite).

2. Modifiez votre programme afin d'afficher à l'écran la table de multiplication de deux entiers compris entre 0 et N , $0 \leq N \leq 18$, où la valeur de N est demandée à l'utilisateur. Vous prendrez soin de demander cette valeur tant qu'elle ne sera pas à l'intérieur de l'intervalle $[0; 18]$.