

Travaux dirigés 7

1 For ou while ?

Il est demandé de résoudre les questions suivantes sans définir de fonctions utilisateurs, directement dans le `main`, et en faisant le meilleur choix entre `for` et `while`.

1.1 Puissances successives d'un entier

Question A. Écrire un programme qui demande un entier n à l'utilisateur et calcule puis affiche n^{10} (n puissance dix).

Exemple d'exécution.

```
Donner un entier : 2
2 exposant 10 = 1024
```

Question B. Comment modifier simplement votre programme pour qu'il affiche le calcul des puissances successives de l'entier n saisi, de n^0 jusqu'à n^{10} ?

Exemple d'exécution.

```
Donner un entier : 2
2 exposant 0 = 1
2 exposant 1 = 2
2 exposant 2 = 4
...
2 exposant 10 = 1024
```

Question C. Comment modifier simplement votre programme pour qu'il affiche le calcul des puissances successives de l'entier n uniquement à partir de l'exposant 5 ?

Exemple d'exécution.

```
Donner un entier : 2
2 exposant 5 = 32
2 exposant 6 = 64
...
2 exposant 10 = 1024
```

1.2 Unicité des éléments d'un tableau

Nous disposons d'un tableau t de N entiers. Utiliser une constante symbolique pour N . Nous souhaitons savoir si chaque entier apparaissant dans le tableau n'y apparaît qu'une seule fois, autrement dit on veut savoir si chaque entier est unique dans le tableau.

Question D. Écrire un programme qui, étant donné un tableau initialisé t , teste si le premier élément du tableau est unique et affiche `Vrai` si c'est le cas, `Faux` sinon.

Question E. Écrire un programme qui étant donné un tableau initialisé t , teste si tous les éléments sont uniques et affiche `Vrai` si c'est le cas, `Faux` sinon.

2 Fractions

Question F. Une fraction $\frac{p}{q}$ est définie par deux entiers p et q . Le nombre q appelé dénominateur est nécessairement non nul et sera toujours positif, le nombre p , appelé numérateur, peut être négatif ou nul. Définir un type utilisateur pour les fractions (sans tenir compte des questions de signe qui n'ont d'importance que pour l'affichage).

Question G. Déclarer et définir une fonction `multiplier_fractions` qui prend deux fractions en argument et renvoie la fraction obtenue par multiplication des deux fractions (ne pas chercher à simplifier la fraction obtenue). *Répondre en faisant bien apparaître d'une part la déclaration, d'autre part la définition.*

Question H. Même question pour la somme de deux fractions `additionner_fractions`.

Question I. Déclarer et définir une procédure affichant une fraction passée en argument exactement comme dans l'exemple suivant où les fractions $\frac{34}{26}$, $\frac{-34}{26}$, $\frac{34}{1}$, $\frac{0}{1}$ sont affichées tour à tour :

```
34/26
-34/26
34
0
```

Attention à bien respecter les deux derniers affichages.

Question J. Comment feriez-vous pour simplifier automatiquement les fractions? Proposer une fonction `normaliser_fraction` qui pourra faire appel à des fonctions arithmétiques usuelles.

3 Fonctions

Question K. Déclarer et définir :

1. une fonction `minimum` qui prend en entrée deux nombres à virgule et retourne leur minimum ;
2. Une procédure `afficher_ligne` qui prend en entrée un entier `n` et un caractère `c` et affiche une ligne contenant `n` fois la caractère `c` ;
3. Une fonction `neper` qui prend en entrée un entier `n` et retourne la valeur de la somme suivante :

$$1 + \sum_{k=1}^{k=n} \frac{1}{k!}.$$

Vous pouvez faire appel à une fonction `int factorielle(int n)` calculant la factorielle de son argument.

4. Une fonction `saisie_utilisateur` sans paramètre qui demande un entier à l'utilisateur et retourne cet entier.
5. Une procédure `tester_mes_fonctions` sans paramètre, qui appelle chacune des fonctions et procédures précédentes sur des paramètres de votre choix.

Bonus

Question bonus. Au choix. Déclarer et définir une fonction qui calcule le pgcd de deux entiers positifs ou nuls. Ou bien, déclarer et définir une procédure `afficher_disque` qui prend en paramètre un entier `rayon` et affiche un disque d'étoiles de ce rayon.