

Schéma itératif [it]

Exercices de cours

Karine Zampieri, Stéphane Rivière



Version 17 mai 2018

Table des matières

1 Appréhender le cours	2
1.1 Le nombre de poissons / pgnpoissons	2
1.2 Carré d'un entier / pgcarres	3
1.3 Calcul par récurrence d'un minimum / pgvmin	4
1.4 Placement d'un capital / pgcapital	5
1.5 Prêt à taux constant / pgemprunt	6
2 Appliquer le cours	7
2.1 Comptage d'entiers / pgcomptage	7
2.2 Le prix de la communication / pgcxinet	9
3 Approfondir le cours	10
3.1 Approximation du sinus / pgapproxsin	10

Java - Exercices de cours (TP)



Mots-Clés Schéma itératif ■

Difficulté ••○ (2 h) ■



Remarque

Les exercices n'utilisent pas le module @[Algorithmes paramétrés].

1 Appréhender le cours

1.1 Le nombre de poissons / pgnpoissons



Objectif

Un poissonnier sert un client qui a demandé x Kg de poisson. Il pèse successivement différents poissons et s'arrête dès que le poids total égale ou dépasse x Kg. Cet exercice donne le nombre de poissons servis. Exemple d'exécution :

```
Poids voulu (en gr)? 1000
Poids du poisson? 350
==> 1 poisson(s) pour un poids total de 350 gr
Poids du poisson? 280
==> 2 poisson(s) pour un poids total de 630 gr
Poids du poisson? 375
==> 3 poisson(s) pour un poids total de 1005 gr
```



Comment faut-il procéder ?



Écrivez un programme qui saisit le poids voulu de poissons (réel). Affichez l'invite :

```
Poids voulu (en gr)?
```



Déclarez puis initialisez à zéro les variables du poids total (réel) et du nombre de poissons (entier).



Demandez successivement le poids du poisson (réel), actualisez vos variables et affichez l'état actuel jusqu'à ce que le poids total égale ou dépasse le poids voulu (où `[x]` désigne le contenu de `x`) :

```
==> [npoissons] poisson(s) pour un poids total de [poidsTotal]
```



Testez.

1.2 Carré d'un entier / pgcarres



Objectif

Une manière originale de calculer le carré d'un nombre entier n est de faire la **somme** des n premiers **nombre impairs**. Exemples :

- $1^2 = 1$: addition du premier impair
- $2^2 = 1 + 3 = 4$: addition des deux premiers impairs
- $3^2 = 1 + 3 + 5 = 9$: addition des trois premiers impairs
- $4^2 = 1 + 3 + 5 + 7 = 16$: addition des quatre premiers impairs
- $5^2 = 1 + 3 + 5 + 7 + 9 = 25$: addition des cinq premiers impairs



Écrivez un programme qui saisit un entier positif, variable `n`, puis affiche **tous** les carrés de 1 à `n` en utilisant la méthode des impairs. Exemples :

```
4 ==> 1 4 9 16  
10 ==> 1 4 9 16 25 36 49 64 81 100
```

Contrainte : Votre programme ne doit pas comporter de boucles imbriquées.



Testez. Exemple d'exécution :

```
n? 10  
1^2 = 1  
2^2 = 4  
3^2 = 9  
4^2 = 16  
5^2 = 25  
6^2 = 36  
7^2 = 49  
8^2 = 64  
9^2 = 81  
10^2 = 100
```

1.3 Calcul par récurrence d'un minimum / pgvmin



Objectif

Cet exercice calcule par récurrence le minimum d'une suite d'entiers.

Exemple d'exécution :

```
n? 5  
9  
4  
12  
-6  
3  
==> Le minimum est -6
```

Comment trouver le minimum de n entiers donnés au fur et à mesure ?



Donnez une définition par récurrence du minimum.



Comment initialiser la suite ?



Écrivez un programme qui saisit le nombre d'entiers (dans `n`), calcule le minimum des `n` entiers donnés au fur et à mesure (dans `vmin`) puis affiche (où `[x]` désigne le contenu de `x`) :

```
==> Le minimum est [vmin]
```



Testez.

1.4 Placement d'un capital / pgcapital



Objectif

Le 1^{er} janvier de l'année à venir, on place un capital pendant un certain nombre d'années à un certain taux. Le capital évolue suivant le système des intérêts composés.



Écrivez un programme qui saisit :

- Le capital de départ.
- Le nombre d'années.
- Le taux de placement (en %).

Puis affiche pour chaque année : la date, le capital ainsi que les intérêts obtenus au 1^{er} janvier de cette année. Exemple d'exécution :

```
Capital de depart? 10000
Nombre d'annees? 5
Taux de placement? 0.05
Annee 1: 10500 dont 500 interets
Annee 2: 11025 dont 525 interets
Annee 3: 11576.25 dont 551.25 interets
Annee 4: 12155.0625 dont 578.8125 interets
Annee 5: 12762.815625 dont 607.753125 interets
```



Testez.

1.5 Prêt à taux constant / pgemprunt



Objectif

Une banque fait un prêt à une personne X pour un montant total de s_0 euros. Cette personne rembourse chaque mois un montant fixe r et paye un intérêt variable $i = t \cdot s$ avec t le taux d'intérêt mensuel (fixe) et s la somme restant à rembourser (avant déduction du remboursement mensuel). Cet exercice détermine la somme des intérêts encaissés par la banque une fois le prêt remboursé.



Écrivez un programme qui saisit les valeurs du montant de prêt s_0 , du remboursement mensuel r et du taux $0 < t < 1$, puis calcule et affiche la somme des intérêts encaissés et la durée du remboursement.



Testez. Exemple d'exécution :

```
Somme pretee (0==fin)? 30000
Remboursement mensuel (>0)? 1200
Taux d'interet %? 1
Interets encaisses: 3900 (sur 25 mois)
```

2 Appliquer le cours

2.1 Comptage d'entiers / pgcomptage



Objectif

Demander une série d'entiers puis compter le nombre de valeurs positives, de valeurs négatives ainsi que la plus grande et plus petite valeur. Exemple d'exécution :

```
Entrez une serie d'entiers (Finir par 0)
```

```
5 3 8 -5 2 -2 0
```

```
Nombre de positifs est 4
```

```
Nombre de negatifs est 2
```

```
Plus petite valeur est -5
```

```
Plus grande valeur est 8
```



Définissez la constante `sentinelle=0`.



Écrivez un programme qui saisit une série d'entiers jusqu'à ce que l'utilisateur introduit la valeur `sentinelle`.



Testez.



Déclarez et initialisez les compteurs du nombre de valeurs positives `npos` (entier) et celui des valeurs négatives `nneg` (entier).



Complétez la structure `TantQue` afin d'actualiser le compteur concerné selon le signe du nombre saisi.



Affichez (où `[x]` désigne le contenu de `x`) :

```
Nombre de positifs est [npos]
```

```
Nombre de negatifs est [nneg]
```



Testez.



Complétez votre programme afin qu'il détermine :

- La plus grande des valeurs saisies dans `vmax`.
- Et la plus petite dans `vmin`.

Aide simple

Il faudra déclarer les variables ainsi que les initialiser (ici à la première valeur saisie).



Testez.

2.2 Le prix de la communication / pgcxinet



Objectif

Cet exercice aide un utilisateur d'Internet à calculer le prix de sa connexion téléphonique. Pour ce programme doit lui demander d'entrer l'heure à laquelle il s'est connecté et l'heure à laquelle il s'est déconnecté. Vous ferez les simplifications suivantes :

- Calculez avec des heures entières uniquement (pas de minutes).
- L'utilisateur n'est jamais connecté de avant minuit jusqu'à après minuit.
- La durée maximale d'une connexion est de 23 heures.

Les tarifs de connexion sont les suivants :

- Tarif 1 de 7h à 17h : 10 €/heure
- Tarif 2 de 0h à 7h et de 17h à 24h : 5 €/heure



Écrivez un programme qui demande les données et vérifie leur cohérence, puis calcule et affiche les résultats selon l'exemple d'exécution :

```
** hdebut et hfin? 10 5
OUPS... Heure debut apres la fin!
** hdebut et hfin? 10 10
OUPS... Pas connecte du tout!
** hdebut et hfin? 10 25
OUPS... Heure(s) hors journee!
** hdebut et hfin? 6 20
Connection 14 h pour 120 Euros
```

3 Approfondir le cours

3.1 Approximation du sinus / pgapproxsin



Propriété

Pour un réel donné x , la valeur de $\sin(x)$, représentant un angle en radians, peut être approximé par le développement limité suivant :

$$S_n = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^i \frac{x^{2i+1}}{(2i+1)!} + \dots + (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!}$$

où n est le nombre de termes.



Écrivez un programme permettant de calculer la valeur approchée de $\sin(x)$, où x est un réel représentant un angle en radians, pour un nombre de termes donné.



Testez. Exemple d'exécution :

```
x (en radians)? 0.5
Nombre de termes? 5
sin(0.5) = 0.4794255386164159
sin math = 0.479425538604203
```