TD3 : Instructions itératives

Partie 1 : Série d'exercices préliminaires en Java :

Exercice 0.1 :

Écrire un programme java qui lit un entier « X » et qui affiche « X » fois le message « coucou ».

Exercice 0.2 :

Écrire un programme java qui lit un entier « X » et qui affiche les messages : coucou1, coucou2, coucou3, ..., coucouX. Exemple pour X=10 on affiche : « coucou1 » à « coucou10 » et non de « coucou0 » à « coucou9 ».

Exercice 0.3 :

Écrire un programme java qui lit un entier « X » et qui affiche « X » fois le message « Pair » si ce nombre est pair et qui affiche « X » fois le message « Impair » si ce nombre est impair.

Exercice 0.4 :

Écrire un programme java qui lit un entier « X » et qui affiche les messages : Impair1, Pair2, Impair3, Pair4, … Jusqu'à ImpairX ou PairX selon si ce nombre est impair ou pair.

Exercice 0.5 :

Écrire un programme java qui lit un entier « X » et qui affiche les messages : X, X-1, X-2, …, 2, 1, 0.
Exemple : X =8 On affiche : 8,7,6,5,4,3,2,1,0.

Exercice 0.6 :

Écrire un programme java qui lit un entier et qui vérifie si celui-ci est positif. Si l'entier est positif, le programme affiche « ok » et se termine. Sinon, le programme indique que le nombre entré n'est pas positif et redemandera à l'utilisateur d'encoder un nouveau nombre positif. Cette opération ne s'arrêtera que lorsque l'utilisateur recevra un nombre positif.

Partie 2 : Série d'exercices simple avec boucle for en Java :

Exercice 1 :

Écrire un programme java qui lit un réel x et un entier positif p et affiche le résultat du calcul x^p (sans utiliser la bibliothèque mathématique).

Exercice 2 :

Écrire un algorithme qui génère un entier aléatoire compris entre 0 et 20, puis il calcule et imprime sa factorielle.
n! = n*(n-1)*….*3*2*1
0! = 1
Exemple : 5 !=5*4*3*2*1=120

Exercice 3 :

Écrire un algorithme qui génère un entier aléatoire N compris entre 10 et 20, puis qui calcule et affiche le produit des entiers impairs de 1 à N.
Exemple : N=10, calcul des entiers impairs 1*3*5*7*9=945

Exercice 4 :

Écrire un algorithme qui demande un nombre N à l'utilisateur et affiche tous ses diviseurs (avec l'affichage ci-dessous).

Exemple : l'ensemble des diviseurs de 10 est {1, 2, 5, 10}

Partie 3 : Série d'exercices simple avec boucle while en Java :

Exercice 5 :

Écrire un programme qui simule l'introduction d'un code, avec un maximum de 3 essais, lors d'un retrait d'argent à un distributeur.

Exercice 6 :

Écrire un programme java qui saisit un nombre et qui détermine combien de fois il est divisible par deux successivement. Exemples :
11 est divisible 0 fois par 2
8 est divisible 3 fois par 2

Exercice 7 :

Écrire un programme java qui lit au clavier une suite de nombres réels positifs ou nuls (des notes dans l'intervalle 0..20) et calcule la moyenne de ces valeurs selon le principe suivant :

- On vérifie que la note est bien dans l'intervalle 0..20 ou que c'est la valeur -1 pour terminer la saisie.
- On vérifie qu'au moins trois notes ont déjà été saisies.

Exemple de trace d'exécution:
Donner une note entre 0 et 20 ou -1 pour arrêter: 10
Donner une note entre 0 et 20 ou -1 pour arrêter: 12
Donner une note entre 0 et 20 ou -1 pour arrêter: 9
Donner une note entre 0 et 20 ou -1 pour arrêter: 12
Donner une note entre 0 et 20 ou -1 pour arrêter:-1
Nombre de notes saisies 4
Moyenne : 10,75

Exercice 8 :

Écrire un programme qui demande tout d'abord une valeur « max ». Ensuite le programme demande une liste de valeurs entières. Chaque valeur de cette liste doit être strictement supérieure à la valeur précédente. Le programme s'arrêtera dès que la somme des valeurs de la liste sera supérieure à la valeur « max ».

Exercice 9 :

Écrire un programme java permettant de lire un nombre entier N et afficher la décomposition en facteurs premiers.
Exemple : n = 240 = 2×2×2×2×3×5

Exercice 10 :

Écrire un programme qui calcule (et affiche) le PGCD (plus grand commun diviseur) de 2 nombres entiers a et b en testant les nombres entiers successifs décroissants, en partant de a. Le premier nombre qui est diviseur de a et de b est le PGCD. Pour accélérer la méthode, a est le minimum de a et b, et b le maximum.

Note : a et b sont lus au clavier mais a n'est pas forcément le minimum.
Exemple : PGCD de 12 et 18
Comme 12 est plus petit que 18, il faut tester si 12 divise 12 et 18, si
non, tester si 11 divise 12 et 18, si non, tester si 10 divise 12 et 18,
…..
6 divise 12 et 18. Donc 6 est le PGCD de 12 e t18

Partie 4 : Série d'exercices principaux en Java :

Exercice 11 :

Écrire un programme java qui lit un entier et affiche ce nombre est parfait. Un nombre est dit parfait s'il est égal à la somme de ses diviseurs. Exemple : 6=3+2+1

Exercice 12 :

Écrire un programme java permettant de lire un nombre N entier positif et créer un nouveau nombre par l'écriture des chiffres de N de droite à gauche.

Exemple : si N=9876, le nouveau nombre sera 6789

Exercice 13 :

Écrire un programme Java qui convertit un nombre entier décimal en binaire (stocké dans un entier) avec la méthode mathématique.

Exercice 14 :

Écrire un programme qui demande une série de nombre strictement positif (sentinelle 0) et indique si elle est croissante, décroissante, strictement croissante, strictement décroissante ou non triée.

Exercice 15 :

Écrire un programme qui affiche les tables de multiplication des entiers de 1 à 10.

Exercice 16 :

Écrire un programme affiche un motif triangulaire dont la hauteur du triangle (c'est-à-dire son nombre de lignes) sera fournie en donnée (lue au clavier). Exemple de trace d'exécution :

Donner le nombre de lignes : 5

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

Exercice 17 :

Écrire un programme java qui calcule les lignes suivantes. Le deuxième membre est obtenu par calcul. Les premiers membres doivent être générés par une boucle.

```
      1 * 8 + 1 = 9
     12 * 8 + 2 = 98
    123 * 8 + 3 = 987
   1234 * 8 + 4 = 9876
  12345 * 8 + 5 = 98765
 123456 * 8 + 6 = 987654
1234567 * 8 + 7 = 9876543
12345678 * 8 + 8 = 98765432
123456789 * 8 + 9 = 987654321
```

Série alternative d'exercices en Java :

Exercice 1 :

Écrire un programme calculant la somme des n premiers termes de la "série harmonique", c'est-à-dire la somme : 1 + 1/2 + 1/3 + 1/4 + ... + 1/n La valeur de n sera lue en donnée.

Exercice 2 :

Écrire un programme java qui calcule les racines carrées de nombres fournis en donnée. Il s'arrêtera lorsqu'on lui fournira la valeur 0. Il refusera les valeurs négatives. Son exécution se présentera ainsi :

Donnez un nombre positif : 2

Sa racine carrée est : 1.4142135623730951

Donnez un nombre positif : -3

SVP positif donnez un nombre positif : 5
Sa racine carrée est : 2.23606797749979
Donnez un nombre positif : 0

Exercice 3 :

Écrire un programme qui demande une liste de valeurs entières dans l'intervalle [-100,100] terminée par une valeur sentinelle > 100 et qui affiche le nombre de valeur < 0 de cette liste.

Exercice 4 :

Écrire un programme qui affiche les nombres entiers strictement positifs dont le carré est inférieur à un nombre entier introduit par l'utilisateur.

Exercice 5 :

Écrire un programme qui demande d'abord 2 valeurs « min et « max ». Ensuite, le programme demande et affiche 10 valeurs entières triées par ordre croissant comprises entre « min » et « max ». Les valeurs peuvent être saisies dans un ordre quelconque mais elles seront affichées en respectant un ordre croissant et borné par min et max. Les valeurs saisies qui ne respectent pas le tri et le bornage ne seront pas affichées.

Exercice 6 :

Écrire un programme java qui lit 20 valeurs réelles et qui détermine la moyenne des valeurs strictement positives et la moyenne des valeurs strictement négatives.

Exercice 7 :

Écrire un programme qui permet de saisir une suite de chiffres se terminant par 0, et qui calcule et affiche le nombre constitué des chiffres lus à l'envers.

Exemple : Si la suite est constituée des chiffres : 5,3,8,2,0 Alors le programme devra afficher le nombre 2835