
Examen VARI1 (NFP135)

ChatGPT ou logiciels similaires interdits. Durée : 2h

Chaque exercice vaut 3 points pour un total de maximum 21 (dont 1 point bonus).

Vous pouvez aussi écrire des fonctions Processing à la place de Java, car les différences sont minimales.

Exercice 1 Écrire une fonction Java `vitesseMoyenne(...)` qui prend deux arguments de type `double` : une distance (en km) et une durée (en heures). Cette fonction doit renvoyer un `double` qui représente la vitesse moyenne, c.à.d., la distance divisée par la durée. On dit que l'en-tête (ou la signature) de la fonction est :

```
double vitesseMoyenne(double distance, double durée)
```

Le corps de cette fonction (votre réponse) devrait avoir une seule ligne pour faire le calcul $\frac{\text{distance}}{\text{durée}}$ et renvoyer le résultat avec `return`.

Exercice 2 Écrire une fonction Java `grandeFortune(...)` qui prend un seul argument de type `int` ; cette fonction doit renvoyer un `boolean` qui indique si l'argument x représente une grande fortune ou pas. On dit que l'en-tête (ou la signature) de la fonction est :

```
boolean grandeFortune(int x)
```

Note : On dit qu'on a à faire à une grande fortune si x est supérieur à 1 million.

Exercice 3 Écrire une fonction Java `sommeInterval(...)` qui prend deux arguments x et y de type `int` et qui renvoie la $x + (x+1) + (x+2) \cdots + y$. Le résultat sera un `int`. L'en-tête (ou la signature) de la fonction est :

```
int sommeInterval(int x, int y)
```

Attention : si y est inférieur à x , il faut renvoyer -1 et afficher « impossible de calculer cette somme car x est plus grand que y ».

Exemples :

- Si $x = 3$ et $y = 6$, il faut renvoyer 18, parce que $3 + 4 + 5 + 6 = 18$
- Si $x = 3$ et $y = 7$, il faut renvoyer 25, parce que $3 + 4 + 5 + 6 + 7 = 25$
- Si $x = 3$ et $y = 3$, il faut renvoyer 3.
- Si $x = 3$ et $y = 2$, il faut renvoyer -1 et afficher « impossible de calculer cette somme car 3 est plus grand que 2 ».

Exercice 4 Écrire une fonction d'en-tête

```
int lePlusGrandDiviseur(int n)
```

qui renvoie le plus grand diviseur de n . Par exemple, il faut renvoyer 50 pour $n = 100$, 3 pour $n = 9$ ou 11 pour $n = 55$.

Note : la condition `if (x%y==0)` passe uniquement si y est un diviseur de x .

Exercice 5 Écrire une fonction d'en-tête

```
int laPlusGrandeSuite(int[] tab)
```

qui renvoie la plus longue suite (séquence de valeurs comme 11 12 13 14) qu'on trouve dans le tableau `tab` passé comme argument. Exemples :

- Si `tab = [1 2 10 5 4 3 2 1 0]`, il faut renvoyer 2 à cause de la suite 1 2.
- Si `tab = [1 2 3 10 5 4 3 2 1 0]`, il faut renvoyer 3 à cause de la suite 1 2 3.
- Si `tab = [1 2 3 11 12 13 14 5 4 3 2 1 0]`, il faut renvoyer 4 à cause de la suite 11 12 13 14.
- Si `tab = [1 2 3 10 11 12 13 14 5 4 3 2 1 0]`, il faut renvoyer 5 à cause de la suite 10 11 12 13 14.
- Si `tab = [1 3 10 3 2 1 4]`, il faut renvoyer 1, car il n'y a aucune suite de deux valeurs type $x (x + 1)$.

Exercice 6 Écrire une fonction d'en-tête

```
void fib(int[] tab, int n, int x, int y)
```

qui remplit le tableau `tab` avec n éléments calculés selon la formule de Fibonacci.

Voici les règles : il faut mettre

- x à la case 0,
- y à la case 1
- $tab[i] = tab[i - 1] + tab[i - 2]$ pour toute case $i \in \{2, 3, \dots, n\}$

Par exemple, si $n = 5$, $x = 4$, $y = 10$, il faut remplir le tableau avec les valeurs [4, 10, 14, 24, 38].

Exercice 7 Question de théorie :

A.) Que représente la figure à droite ? Décrire en quelques lignes (maximum 10) les interactions et les modules dans cette figure. Par exemple, expliquer ce que représente les flèches des programmes utilisateurs vers l'interface utilisateur, ou détailler l'expression « mémoire RAM (vive ou virtuelle) »

[1.2pt]

B.) Donner six commandes Linux (resp.) pour :

- afficher le contenu d'un fichier `Toto.java`.
- compiler le programme `Toto.java` et lancer ce programme
- se placer dans le dossier `/toto/licencep/`
- se placer dans le dossier parent (c.-à.-d, père)
- afficher les fichiers du dossier courant
- afficher la manuel de la commande `cat`.

[1.8pt]

