

DS 2 DEV (27 octobre 2022)

Aucune machine et aucun document

Exercice 1 [4.5] Analyse de code source et vocabulaire de base en algorithmique

```
01 class Compter extends Program {
02     int compter(String chaine) {
03         int resultat = 0;
04         char lettre;
05         for (int idx = 0 ; idx < length(chaine) ; idx = idx+1) {
06             lettre = charAt(chaine, idx);
07             if (((lettre > 'A') && (lettre < 'Z')) || ((lettre > 'a') && (lettre < 'z'))){
08                 resultat = resultat + 1;
09             }
10         }
11         return resultat;
12     }
13 }
```

1. [0.5] Donner la ou les n° de ligne impliquant une condition.
2. [0.5] Donner la ou les n° de ligne impliquant une accumulation
3. [1] Dessiner l'organigramme de la fonction compter.
4. [1] Quels résultats sont produits par ce programme lors des appels :
compter("bed"), compter("4ez t"), compter("") et
compter("arbre").
5. [0.5] Un bug est présent dans cet algorithme. Pouvez-vous trouver une chaîne l'illustrant ?
6. [1] Indiquer ce qu'il suffit de modifier dans le programme pour corriger ce bug.

Exercice 2 [3.5] Exercice de cours sur les fonctions et tableaux

Échauffement sur les tableaux

- 1a. [0.5] Déclarer un tableau nommé mots pouvant stocker des chaînes de caractères
- 1b. [0.5] Allouer le tableau mots afin qu'il puisse contenir 42 mots.

Suite sur les tableaux

- 2a. [0.5] Donner l'expression permettant d'accéder à l'avant-dernier élément d'un tableau nommé unTableau.
- 2b. [1] Donner le code permettant de déclarer, allouer et initialiser en une seule ligne (déclaration en extension) un tableau de jours avec les 7 jours de la semaine représentés par des chaînes de caractères.

Un peu de fonction ...

- 3a. [0.5] Donner la **signature** de la fonction nommée consecutives indiquant si un mot contient deux lettres consécutives (située l'une après l'autre dans le mot) qui se suivent dans l'alphabet (situées l'une après l'autre dans

l'alphabet). Exemples : c'est le cas pour "cabot" ou "studieux", mais pas pour "algo".

3b. [0.5] Quel type de répétition faudrait-il utiliser pour écrire cette fonction ? Pourquoi ?

Exercice 3 [6] Exercice de manipulation sur les chaînes

1. [2] Les deux questions suivantes visent à déterminer des signatures de fonction en analysant leur appel.

a) [0.5] Donner la signature de la fonction testée dans le code ci-dessous.

```
void testEstValide() {  
    assertEquals(true, estValide("Hello"));  
    assertEquals(false, estValide("hello"));  
}
```

b) [1.5] Donner toutes les signatures des fonctions appelées dans ce programme, qu'elles soient prédéfinies ou non.

```
void algorithm() {  
    String mot = readString();  
    int valide = 0;  
    if (estNumerique(charAt(mot, 0)) &&  
        estAlphabétique(substring(mot, 1, length(mot))) {  
        valide = valide + 1;  
    }  
}
```

2. [2] Écrire une fonction `void algorithm()` qui effectue une saisie au clavier d'une phrase (suite de mots séparés par un unique espace). Chacun des mots est ensuite affiché seul sur sa ligne et précédé d'un numéro de ligne comme ci-dessous. Ainsi, en supposant que la phrase saisie soit "Hello dear friend", le programme affichera ceci :

```
1. Hello  
2. dear  
3. friend
```

Attention : il est interdit d'écrire d'autres fonctions que `void algorithm()`.

3. [2] On souhaite créer un jeu de la vie. Le programme affiche la génération courante et demande à l'utilisateur si il souhaite continuer. Si la saisie est "stop" le programme s'arrête, sinon une nouvelle génération est calculée puis affichée.

Voici les fonctionnalités déjà écrites par votre enseignant·e (qu'il ne faut donc pas écrire !) :

```
boolean[][] creerMonde()
int nombreVoisinesVivantes(boolean[][] monde, int idxLig, int idxCol)
void afficher(boolean[][] monde)
boolean evolution(boolean etatCelluleActuelle, int
nbVoisinesVivantes)
boolean[][] generationSuivante(boolean[][] monde)
```

Donner l'algorithme principal qui utilise les fonctions ci-dessus pour répondre à l'objectif décrit précédemment.

Exercice 4 [6] Opérateurs de téléphonie et réponse à toutes vos questions

1. [2] Un opérateur de téléphonie propose différents types de forfaits et options et souhaite disposer d'une fonction calculant automatiquement le prix mensuel. Voici les règles commerciales de cet opérateur :

- le forfait de base est à 15€ par mois
- si le client a besoin d'option
 - c'est 5€ en plus pour des appels illimités,
 - c'est 15€ en plus pour des data illimitées,
 - c'est 20€ en plus pour du roaming international.
- si le client a deux options ou plus, il bénéficie de 9€ de réduction
- si le client a 24 mois d'ancienneté ou plus, sa facture est divisée par 2

Écrire la fonction `int coutMensuel(int moisAnciennete, boolean appels, boolean data, boolean roaming)`.

2. [2] Un opérateur de téléphonie souhaite normaliser la représentation de numéros de téléphone. Les usagers saisissent leur numéro sous la forme d'une suite de chiffres, comme 0634567498, l'opérateur souhaite les stocker sous la forme : 06.34.56.74.98. Ce dernier vous demande donc d'écrire la fonction `normaliser` qui réalise ce traitement.

```
void testNormaliser() {
    assertEquals("06.34.56.74.98", normaliser("0634567498"));
}
```

3. [2] On souhaite vérifier qu'une chaîne de caractères contient 42. Par contre, on ne dispose pas de `substring` mais uniquement de `charAt` pour réaliser ce traitement. Écrire la fonction `boolean laGrandeQuestionSurLaVieLuniversEtLeReste(String palabres)` qui retourne `true` si `palabres` contient 42 et `false` sinon.