

Licence 1 - Info1A - Examen Janvier - 2019/2020 - Durée 2H

Seuls les documents issus des cours, TDs et Tps sont autorisés

Soignez l'écriture de vos programmes, et en particulier l'indentation, il en sera tenu compte dans la notation.

Exercice 1. Jolie frise (4pts). Ecrire un programme Java qui lit un entier positif n et qui affiche à l'écran le motif suivant. On donne ci-dessous trois exemples pour $n = 4$, $n = 5$ et $n = 6$:

Pour $n = 4$: 10 1100 111000 11110000

Pour $n = 5$: 10 1100 111000 11110000 1111100000

Pour $n = 6$: 10 1100 111000 11110000 1111100000 111111000000

Exercice 2. Déroulons un programme (4pts). Qu'affiche le programme suivant? On justifiera la réponse en donnant les contenus des deux variables i et T à la fin de chaque passage dans la boucle.

```
int i,taille;
String S="Luc,Bob,Tom";
String T="", mot="";
taille=S.length();
for(i=0;i<=taille-1;i++)
{ if(S.charAt(i)==' ')
  { T=mot+" "+T;
    mot="";
  }
  else
    mot=mot+S.charAt(i);
}
T=mot+" "+T;
System.out.println("La valeur de T est:"+T);
```

Exercice 3. Les cases isolées (6pts). On considère un tableau tab à deux dimensions, de taille $m \times n$ avec $1 \leq m, n \leq 100$, contenant des entiers. On souhaite faire un programme Java permettant de proposer à l'utilisateur les différentes fonctionnalités suivantes.

- Saisir les entiers m et n et faire la déclaration du tableau tab .
- Remplir le tableau tab avec des chiffres de 0 à 9 choisis aléatoirement.
- Pour chaque colonne du tableau, calculer la moyenne et la stocker dans un tableau moy à une dimension.
- Calculer et afficher la moyenne maximale située dans le tableau moy .
- Une case $\text{tab}[i][j]$ du tableau est dite *isolée* lorsqu'elle ne contient pas 0 et que toutes ses cases voisines contiennent 0 (les cases voisines de $\text{tab}[i][j]$ sont $\text{tab}[i-1][j]$, $\text{tab}[i][j+1]$, $\text{tab}[i][j-1]$ et $\text{tab}[i+1][j]$). Afficher le nombre de cases isolées du tableau tab . On ne considèrera pas les cases situées sur les bords du tableau.

Exercice 4. Fonction de parité (2pts). Ecrire une fonction Java qui renvoie *true* si exactement deux des trois entiers passés en paramètres sont pairs, et *false* sinon.

Exercice 5. Les sushis d'Akatsuki (4pts). Au début du repas, on dispose de 100 sushis sur la table. Un joueur et l'ordinateur choisissent chacun leur tour un nombre de sushis à manger (le nombre doit impérativement être choisi, de façon aléatoire pour l'ordinateur, parmi 1, 2 et 3, et c'est le joueur qui débute le jeu). Celui qui mange le dernier sushi a perdu. Ecrire un programme Java qui simule ce jeu et qui affiche le joueur gagnant. Indication: dans le cas où le joueur choisit plus de sushis qu'il n'en reste sur la table, il perd le jeu.