

Nom : \_\_\_\_\_

Groupe (encercler) 1 – 2 – 3 – 4

**Examen synthèse – Introduction à la programmation [40% de la note finale]**

**CONSIGNES GÉNÉRALES**

- L'usage du téléphone cellulaire est interdit en tout temps;
- Le plagiat, sous toutes ses formes, ou le partage de documents entre étudiants entraînent la note de zéro (0) pour tout étudiant impliqué;
- Cet examen comporte un total de six (6) questions, réparties en deux (2) volets : théorique et pratique;
- Cette évaluation est d'une durée totale de trois (3) heures pour compléter les 2 volets.

**DÉROULEMENT DE L'EXAMEN**

- D'abord compléter le volet théorique comprenant les questions 1 à 4, inclusivement;
- Une fois le volet théorique complété, remettre votre copie au surveillant afin de récupérer le volet pratique comprenant les question 5 et 6.

**CONSIGNES POUR LE VOLET THÉORIQUE**

- Aucune documentation, numérique ou physique, n'est autorisée;
- L'usage de la calculatrice personnelle est autorisé, aucun partage entre étudiants;
- L'ordinateur peut être sous tension mais les moniteurs doivent demeurer fermés en tout temps;
- Répondre directement sur ce document aux endroits indiqués (total de 4 questions).

**CONSIGNES POUR LE VOLET PRATIQUE**

- Seul l'utilisation des ordinateurs du parc informatique du cégep sont permis;
- La documentation personnelle, numérique ou physique, est permise;
- Aucun partage entre étudiants;
- L'accès à internet est interdit;
- Créer un projet unique Java dans *Visual Studio Code* comprenant une classe par question (total de 2);
- Remettre le projet dans une archive compressée avant l'heure de fermeture des boîtes de remise Léa prévue à cet effet, sans quoi votre remise peut être refusée.

---

**POINTAGE DE L'EXAMEN**

Volet	Ventilation	Valeur	Pondération	TOTAL
<b>Théorique</b>	Question 1 – 10 points Question 2 – 10 points Question 3 – 10 points Question 4 – 10 points	40 points	= ____ / 10	____ / 40
<b>Pratique</b>	Question 5 – 15 points Question 6 – 15 points	30 points	= ____ / 30	

## VOLET THÉORIQUE

## Question 1

/10 points

Inscrire l'affichage produit par le programme principal suivant, en prenant bien soin de respecter les numéros de lignes indiqués :

```
public static void main(String[] args) {  
  
    int iValeur = 1;  
    String message = "";  
  
    while(iValeur <= 15) {  
        if(iValeur % 2 == 0) {  
            message += iValeur + " est pair\n";  
            iValeur++;  
        }  
        else if(iValeur % 5 == 0) {  
            message += iValeur + " est multiple de 3\n";  
            iValeur--;  
        }  
        else {  
            message += iValeur + "\n";  
            iValeur++;  
        }  
        iValeur += 3;  
    }  
  
    System.out.println(message);  
    System.exit(0);  
}
```

Affichage produit en console (lignes numérotées)

1	1
2	5 est multiple de 3
3	7
4	11
5	15 est multiple de 3
6	
7	
8	
9	
10	
11	
12	

## Question 2

/10 points

Inscrire l'affichage produit par le programme principal suivant, en prenant bien soin de respecter les numéros de lignes indiqués :

```
public static void main(String[] args) {  
  
    final int TAILLE = 6;  
    int ctr;  
    int tab1[] = new int[TAILLE];  
    int tab2[] = {11, 7, 5, 9, 10, 8};  
  
    for (ctr = 0; ctr < TAILLE; ctr++)  
        tab1[ctr] = tab2[ctr] % 2;  
  
    boucle1:  
    for (ctr=0 ; ctr <= 5; ctr++)  
        switch (ctr) {  
            case 1: case 3:  
                tab1[ctr] = ctr;  
                break;  
            case 2: case 5:  
                tab1[ctr] = ctr + 2;  
                break boucle1;  
            default: tab1[ctr] = 0;  
        }  
  
    for (ctr = 0; ctr < TAILLE; ctr++) {  
        if (ctr == 3) continue;  
        tab1[(TAILLE - 1) - ctr] = tab2[ctr];  
        System.out.println("tab1[" + ctr + "] = " + tab1[ctr]);  
        System.out.println("tab2[" + ctr + "] = " + tab2[ctr]);  
    }  
  
    System.exit(0);  
}
```

Affichage produit en console (lignes numérotées)

1	tab1[0] = 0
2	tab2[0] = 11
3	tab1[1] = 1
4	tab2[1] = 7
5	tab1[2] = 4
6	tab2[2] = 5
7	tab1[4] = 7
8	tab2[4] = 10
9	tab1[5] = 11
10	tab2[5] = 8
11	
12	

### Question 3

/10 points

Inscrire l'affichage produit par le programme principal suivant, en prenant bien soin de respecter les numéros de lignes indiqués :

```
public static void main(String[] args) {  
  
    int iTabNombre[][] = new int [4][1];  
    String strResult = "";  
    int nbr1, nbr2;  
  
    for(nbr1 = 0; nbr1 < iTabNombre.length; nbr1++)  
        for(nbr2 = 0; nbr2 < iTabNombre[0].length; nbr2++)  
            iTabNombre[nbr1][nbr2] = 3 * (nbr1 + nbr2) + 1;  
  
    for(nbr1 = 0; nbr1 < iTabNombre.length; nbr1++)  
        for(nbr2 = 0; nbr2 < iTabNombre[0].length; nbr2++)  
            strResult += "\n" + iTabNombre[nbr1][nbr2];  
  
    System.out.println(strResult);  
  
    System.exit(0);  
}
```

Affichage produit en console (lignes numérotées)

1		[Cette ligne est vide]
2	1	
3	4	
4	7	
5	10	
6		
7		
8		
9		
10		
11		
12		

#### Question 4

/10 points

Inscrire l'affichage produit par le programme principal suivant, en prenant bien soin de respecter les numéros de lignes indiqués :

```
public class Q4 {  
  
    public static void main(String[] args) {  
        final int MAX = 5;  
        int ctr, iResultat;  
        int iTableau[] = new int[MAX];  
  
        for (ctr=0; ctr <= MAX-1; ctr++)  
            iTableau[ctr] = ctr * ctr;  
  
        iResultat = fonctionExamen(iTableau, MAX-2);  
  
        for (ctr=0; ctr < iTableau.length; ctr++)  
            System.out.println("Élément " + ctr + " -> " + iTableau[ctr]);  
  
        System.out.println("Résultat-> " + iResultat);  
        System.exit(0);  
    }  
  
    static public int fonctionExamen(int tableau[], int iNbrN) {  
        int iNbr1, iNbr2;  
        iNbr2 = 0;  
  
        for (iNbr1=1; iNbr1 <= iNbrN; iNbr1++) {  
            iNbr2 += tableau[iNbr1];  
            if (iNbr1 > iNbrN) {  
                break;  
            }  
        }  
        return iNbr2;  
    }  
}
```

Affichage produit en console (lignes numérotées)

1	Élément 0 -> 0
2	Élément 1 -> 1
3	Élément 2 -> 4
4	Élément 3 -> 9
5	Élément 4 -> 16
6	Résultat-> 14
7	
8	
9	
10	
11	
12	

**VOLET PRATIQUE**

Nom : \_\_\_\_\_

**Question 5****/15 points**

Écrire un programme Java permettant de comparer le score d'un golfeur avec les normales enregistrées dans le tableau à 2 dimensions (matrice) pour vérifier si le score du golfeur est sous la normale, normale ou au-dessus de la normale. Vous devez déclarer et initialiser, **en une même opération**, le tableau à 2 dimensions des normales par groupes d'âge qui est présenté ci-dessous. Utilisez l'âge du golfeur ainsi que le numéro du trou pour accéder respectivement aux lignes et aux colonnes du tableau à 2 dimensions.

*Normales par trou selon l'âge*

	Trou 1	Trou 2	Trou 3	Trou 4	Trou 5	Trou 6	Trou 7	Trou 8	Trou 9
12 ans	6	5	6	5	4	6	5	4	5
13 ans	5	4	5	5	3	6	4	4	5
14 ans	5	4	5	4	3	5	4	4	4
15 ans	4	4	4	4	3	5	4	3	4
16 ans	4	4	4	3	3	4	4	3	3
17 ans	4	4	4	3	3	4	3	3	3

Votre programme doit présenter un menu en boucle proposant les trois (3) options suivantes :

**1. Saisie des données**

Vous devez choisir des types de données appropriées pour stocker les informations.

- Validations requises (la saisie recommence tant que les données sont invalides)
  - Le nom ne doit pas être vide;
  - L'âge doit être compris dans l'intervalle [12, 17] inclusivement;
  - Le score de chacun des trous doit être supérieur ou égal à 1 (un).

**2. Affichage des informations**

L'affichage peut être fait de la manière de votre choix (console ou boîtes de dialogue). Lorsque cette option est choisie, on indique le nom du golfeur, son âge, ainsi qu'une des 3 mentions suivante pour chacun des trous :

- *Au-dessus de la normale*
- *Normale*
- *Sous la normale*

On retrouve également au bas de l'affichage le pointage total (la somme pour les 9 trous) du golfeur.

Voici un exemple d'affichage attendu avec les données associées :

Données saisies	Affichage produit
Nom : Marianne Âge : 13 ans Scores : 6, 3, 6, 4, 5, 6, 5, 4, 4	Résultats pour Marianne, 13 ans : Trou 1 - score de 6 : au-dessus de la normale Trou 2 - score de 3 : sous la normale Trou 3 - score de 6 : au-dessus de la normale Trou 4 - score de 4 : sous la normale Trou 5 - score de 5 : au-dessus de la normale Trou 6 - score de 6 : normale Trou 7 - score de 5 : au-dessus de la normale Trou 8 - score de 4 : normale Trou 9 - score de 4 : sous la normale Pointage total de 43

**3. Quitter**

Le programme quitte correctement.

**Question 6****/15 points**

Concevoir les trois (3) fonctions suivantes en Java :

1. Générer une matrice carrée (autant de lignes que de colonnes) dont la taille est aléatoire entre 3 et 8 et dont les valeurs sont des entiers aléatoires entre 0 et 20 (inclusivement);
2. Pour afficher en console une matrice en lignes et en colonnes;
3. Pour trier une matrice ligne par ligne. Chaque ligne est triée de manière indépendante des autres lignes.

Les signatures des fonctions ne vous sont pas imposées, alors vous devez vous-même les définir de manière cohérente.

Dans le programme principal, vous devez faire appel à vos fonctions afin de générer une matrice carrée quelconque (aléatoirement), la faire afficher dans son état initial, la trier par ligne et la réafficher une fois triée.

*Exemples (2) d'exécutions possibles du programme :*

Matrice de départ					Matrice de départ						
3	12	15	7	17	20	0	11	12	6	11	1
18	19	14	11	1	20	0	2	4	20	4	7
4	3	17	5	13	4	12	18	11	20	5	3
13	16	6	9	0	16	9	0	17	10	4	9
19	18	11	2	2	18	15	17	18	2	11	11
Matrice triée					4	11	20	10	17	1	1
					15	9	11	18	1	6	18
Matrice de départ					Matrice triée						
3	7	12	15	17	0	1	6	11	11	12	20
1	11	14	18	19	0	2	4	4	7	20	20
3	4	5	13	17	3	4	5	11	12	18	20
0	6	9	13	16	0	4	9	9	10	16	17
2	2	11	18	19	0	11	11	15	17	18	18
					1	1	4	10	11	17	20
					1	6	9	11	15	18	18

*Bon succès!*

*-Maxime & Maxime, qui vous souhaitent d'excellentes vacances du temps des Fêtes!*