

420-0P1-GG

Introduction à la programmation structurée

Projet 1 (Partie 1)

Structures de programmation

Objectifs: Développer des applications simples utilisant les structures de décision et les boucles

Durée: Deux séances de 3h

Pondération: 20%

Travaux à remettre: Remise électronique, d'un fichier *zip* contenant tous les codes sources (6 janvier 2026 - remise avant 23 :59 :59)

Modalités de remise

Commencez par créer un nouveau projet et nommez-le Projet1. Ensuite, pour chaque question, créez un nouveau fichier avec une fonction main() (Question1, Question2...).

Pour la remise, zippez votre projet contenant l'ensemble des questions. Entrez le nom du fichier zippé dans le format suivant : PROJET1_INITIALS.zip. Vous pouvez alors la téléverser sur le site Léa.

Questions

1. Calculez et affichez le résultat de la somme suivante : $\sum_{i=1}^{20} i^2$
2. Demandez à l'usager d'entrer des nombres entiers appartenant à l'intervalle [0,100]. Pour chaque nombre lu, le programme doit afficher si la valeur entrée est paire ou impaire. Le programme s'arrête lorsque l'usager entre une valeur erronée, en affichant « FIN. ».

```
Entrez un nombre entre 0 et 100 :  
25  
Ce nombre est impair.  
Entrez un nombre entre 0 et 100 :  
42  
Ce nombre est pair.  
Entrez un nombre entre 0 et 100 :  
151  
Ce nombre est impair.  
FIN.
```

3. Demandez à l'usager d'entrer trois nombres entiers. Déterminez si un des nombres est la somme des deux autres. Si tel est le cas, affichez la somme correspondante. Sinon, affichez « aucune somme possible! ». L'affichage obtenu doit être semblable aux suivants :

Entrez le premier nombre entier : 4 Entrez le deuxième nombre entier : 5 Entrez le troisième nombre entier : 1 $4 + 1 = 5$	Entrez le premier nombre entier : 4 Entrez le deuxième nombre entier : 5 Entrez le troisième nombre entier : 2 Aucune somme possible!
---	--

4. Demandez à l'usager d'entrer un nombre entier. S'il est inférieur ou égal à 1, affichez un message d'erreur. Sinon, déterminez et affichez tous les diviseurs de ce nombre. L'affichage obtenu doit être semblable aux suivants :

Entrez un nombre positif : 26 Les diviseurs de 26 sont : 1 2 13 26	Entrez un nombre positif : 1 Erreur!
---	---

5. Demandez à l'usager d'entrer deux valeurs, a et b . Ensuite, calculez et affichez le résultat de la somme suivante : $\sum_{i=1}^a \sum_{j=1}^b (i * j)$

6. Demandez à l'usager d'entrer un temps en secondes. Ensuite, affichez les nombres d'heures, de minutes et de secondes correspondants. L'affichage obtenu doit être semblable au suivant :

Entrez un nombre de secondes : 63648 $63648s = 17h + 40m + 48s$
--

7. Demandez à l'usager d'entrer une phrase. Ensuite, déterminez et affichez le nombre de voyelles (a , o , u , i , e ou y) dans la phrase. **Vous n'avez pas le droit d'utiliser la structure if.** L'affichage obtenu doit être semblable au suivant :

Entrez une phrase : je suis UNE phrase qui utilise toutes les VOYELLES Nombre de voyelles : 21
--

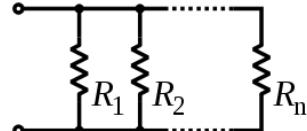
Indice: « str.length() » retourne la taille de la chaîne str et « str.charAt(i) » retourne le caractère à la position i (de 0 à taille-1).

8. Demandez à l'usager d'entrer un nombre entier. Ensuite, calculez la somme de tous les chiffres de ce nombre et affichez le résultat. L'affichage obtenu doit être semblable aux suivants :

```
Entrez un nombre entier : 5361  
La somme des chiffres est : 15
```

9. On considère un circuit de n résistances en parallèle. La valeur de la résistante équivalente est obtenue grâce à la formule suivante :

$$R_{equi} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_n}}$$



Source : Wikipédia

Demandez à l'usager d'entrer un nombre de résistances. Ensuite, demandez à l'usager d'entrer les valeurs de chacune des résistances (valeurs réelles). Finalement, calculez et affichez la valeur de la résistance équivalente, avec deux chiffres après la virgule. L'affichage obtenu doit être semblable au suivant :

```
Entrez le nombre de résistances : 3
Entrez les valeurs des 3 résistances : 2.0 4.3 6.8
La valeur de la résistance équivalente est 1.14
```

10. Demander à l'usager d'entrer un nombre supérieur ou égal à 2 (redemander indéfiniment, au besoin).

Ensuite, afficher la forme suivante, avec le nombre de lignes égal au nombre entré par l'usager.

L'affichage obtenu doit être semblable aux suivants :

Entrez le nombre de lignes : 0 Entrez le nombre de lignes : 1 Entrez le nombre de lignes : 5 1 12 123 1234 12345	Entrez le nombre de lignes : 12 1 12 123 1234 12345 123456 1234567 12345678 123456789 1234567890 12345678901 123456789012
---	---

Bonus :

Implémentez le jeu qui permet à l'usager de deviner un nombre secret (compris entre 0 et 99) et qui lui affiche des indices (« plus haut » ou « plus bas »). Le jeu se termine lorsque l'usager découvre le nombre. On doit alors lui afficher le nombre d'essais dont il a eu besoin. L'affichage obtenu doit être semblable au suivant :

```
Entrez un nombre entre 0 et 99 : 25
Plus haut!
Entrez un nombre entre 0 et 99 : 50
Plus haut!
Entrez un nombre entre 0 et 99 : 75
Plus bas!
Entrez un nombre entre 0 et 99 : 70
Plus haut!
Entrez un nombre entre 0 et 99 : 72
Plus haut!
Entrez un nombre entre 0 et 99 : 74
Bravo! Vous avez trouvé le nombre en 6 essais.
```

Indice : vous pouvez générer le nombre secret aléatoirement grâce à l'instruction suivante:

```
int nombreSecret = (int)(Math.random()*100);
```