Module Python - TP1 - Les bases du langages

Base : Eric Thomas - Mise à jour : Nicolas 'Lomens' Resin

Révisez avec ce site

Chaque exercice devrait être fait dans un fichier python pour pouvoir le conserver.

Exercice 01

Écrire un programme qui calcule l'indice de masse corporelle (IMC) d'un individu. Votre programme programme doit commencer par lire la taille et le poids de l'utilisateur.

Il doit ensuite utiliser l'une des deux formules suivantes pour calculer l'IMC avant de l'afficher. Nous indiquerons la taille en mètres et le poids en kilogrammes.

L'indice de masse corporelle est calculé à l'aide de cette formule légèrement plus simple :

- IMC = poids / (taille × taille)

Exercice 02

Créez un programme qui lit N entiers de l'utilisateur. N sera défini par le nombre d'entier rentré jusqu'à recevoir un nombre négatif (<0) Une fois tous les nombres rentrés, affichez dans la console:

- Les nombres dans un ordre trié (du plus petit au plus grand)
- Le minimum
- Le maximum
- La moyenne

Exercice 03

On dit généralement qu'une année humaine équivaut à 7 années canines. Cependant, cette simple conversion ne tient pas compte du fait que les chiens atteignent l'âge adulte en deux ans environ. Par conséquent, certains pensent qu'il est préférable de compter chacune des deux premières années humaines comme 10,5 années de chien.

1 années humaines équivalent à 10,5 années canines, puis de compter chaque année humaine supplémentaire comme 4 années canines.

Écrire un programme qui implémente la conversion des années humaines en années canines décrite dans le paragraphe précédent. Assurez-vous que votre programme fonctionne correctement pour les conversions de moins de deux années humaines et pour les conversions de deux années humaines ou plus.

Votre programme devra afficher un message d'erreur approprié si l'utilisateur entre un nombre négatif.

Exercice 04

La valeur de π peut être approchée par la série infinie suivante :

$$\pi \approx 3 + \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \frac{4}{8 \times 9 \times 10} + \frac{4}{10 \times 11 \times 12} - \cdots$$

Écrire un programme qui affiche 15 approximations de π . La première approximation doit utiliser uniquement le premier terme de la série infinie. Chaque approximation supplémentaire affichée par votre programme doit inclure un terme supplémentaire dans la série, ce qui en fait une meilleure approximation de π que toutes les approximations affichées précédemment

Puis améliorer le programme pour afficher N approximations, N étant un entier positif.

Votre programme devra afficher un message d'erreur approprié si l'utilisateur entre un nombre négatif.

Exercice 05

Écrire un programme qui convertit un nombre décimal (base 10) en binaire (base 2). Lire le décimal de l'utilisateur sous la forme d'un entier, puis utilisez l'algorithme de division illustré ci-dessous pour ffectuer la conversion.

Lorsque l'algorithme est terminé, le résultat contient la représentation binaire du nombre inversé. Affichez le résultat dans le bon ordre, ainsi qu'un message approprié.

L'algo est le suivant :

```
le résultat est une chaîne vide
q représente le nombre à convertir
répétez
    r égal au reste lorsque q est divisé par 2
    Convertir r en une chaîne de caractères et ajoutez-la au début du résultat.
    Division entière q par 2 et enregistrez le résultat dans q
jusqu'à ce que q soit égal à 0
```

Exercice 06

Vous devez écrire un programme revision-ex1.py qui .

- demande le type d'opération souhaité : (a)ddition, (s)oustraction, (m)ultiplication et (d)ivision.
- Saisir les deux variables.
- En fonction de la saisie de l'utilisateur, enregistrez le résultat dans une variable result.
- Affichez le résultat avec l'opération complète (ex : 4 + 5 = 9).
- Gère le cas de la division par 0

Exemples de résultats :

```
type d'opération souhaité : (a)ddition, (s)oustraction, (m)ultiplication et
(d)ivision :a
Saisir un chiffre 1 :1
Saisir un chiffre 2 :2
résultat : 1 + 2 = 3
```

```
type d'opération souhaité : (a)ddition, (s)oustraction, (m)ultiplication et
(d)ivision :d
Saisir un chiffre 1 :5
Saisir un chiffre 2 :0
résultat : 5.0 / 0.0 = division impossible
```

Puis vous améliorerez le programme avec ces fonctionnalités :

- Ajouter le cas où la saisie n'est pas correcte.
- Ajouter un choix pour quitter l'application : Un autre calcul ? o/n (boucle while)
- Rendre les choix insensibles à la casse (a et A sont les même valeurs)

```
type d'opération souhaité : (a)ddition, (s)oustraction, (m)ultiplication et
(d)ivision : zz
calcul non compris !!
Un autre calcul ? o/n o
type d'opération souhaité : (a)ddition, (s)oustraction, (m)ultiplication et
(d)ivision : M
x = 3
y = 4
3 * 4 = 12
Un autre calcul ? o/n N
```

Exercice 07

Réalisez un programme qui va permettre de générer à la volée des plaques d'immatriculations françaises.

Le format est donc AA-111-AA (A représente une lettre et 1 un chiffre). Il est impossible d'avoir les lettres I, O ou U. Il est aussi impossible d'avoir SS comme combinaison.

```
Valide : AS-145-RE
Invalide : SS-945-OP
```