

LES ENTREES / SORTIES (PRINT ET INPUT)

Exercice 1 Périmètre et surface d'un cercle

Calculer le périmètre et la surface d'un cercle dont le diamètre est donné.

Exercice 2 : Coefficient de perméabilité

Soit un échantillon de sable grossier de hauteur H et diamètre D donnés. Il est placé dans un *perméamètre* à charge constante. L'eau percole à travers l'échantillon sous une charge C donnée. Ecrire un programme qui calcule la valeur du coefficient de perméabilité k (en cm/s) connaissant le poids P de l'eau recueillie en un temps T donné.

On utilisera les formules suivantes :

$$\text{Débit : } Q = \text{Volume recueilli} / T$$

$$\text{Débit : } Q = \text{Vitesse} * \text{Section}$$

$$\text{Gradient hydraulique : } i = C / \text{Hauteur}$$

$$\text{Vitesse} = k * i$$

On pourra tester le programme pour les valeurs suivantes :

$$H = 15\text{cm} \quad D = 5,5\text{cm} \quad C = 40\text{cm} \quad P = 40\text{g} \quad T = 6\text{s}$$

Exercice 3 : Conversion de température

La valeur en Celsius d'une température exprimée en degrés Fahrenheit se calcule grâce à la formule suivante : $C = (5 / 9) * (F - 32)$

Ecrire une fonction qui convertisse une température donnée en Fahrenheit ou inversement une température donnée en Celsius.

Les températures sont fournies sous forme de chaînes de caractères comprenant la température et l'unité. Par exemple : "32.5C" ou "12.5F". La conversion sera faite dans un sens ou l'autre selon l'unité définie.

LES OPERATEURS // ET %

Exercice 4 : Décodage d'une date

Écrire un programme qui décode une date donnée sous la forme d'un entier à 6 chiffres (jour-mois-année) en 3 entiers à 2 chiffres.

Par exemple, la date 120191 sera décodée en 12/01/91, la date 91133 sera décodée en 09/11/33

Exercices

Aide : on pourra pour ce faire utiliser la division entière et le reste de la division entière
Challenge : le calcul du jour et de l'année peut se faire à l'aide d'une seule opération et celui du mois avec 2 opérations

LES CONDITIONS (IF)

Exercice 5 : Décomposition d'une somme d'argent

On souhaite programmer une machine qui rend la monnaie. Elle dispose de pièces de 2 euros, 1 euro, 50 cts d'euros, 20 cts d'euros, 10 cts d'euros.

Connaissant la somme à payer et la somme versée, écrire un programme qui calcule le nombre de pièces de chaque sorte à rendre, de façon à minimiser le nombre total de pièces rendues.

Indications :

- Les sommes sont saisies sous forme de centimes (valeurs entières)
- On demande de vérifier si la somme versée est suffisante sinon on affiche un message d'erreur
- Seules les valeurs non nulles seront affichées
- On suppose que la machine n'est jamais à cours de monnaie

Exercice 6 : Équation du second degré

Écrire un programme qui imprime les solutions réelles et complexes d'une équation du second degré.

Exercice 7 : Facturation d'un voyage organisé

Une agence de voyage propose un voyage "à la carte" en Italie. La formule est très souple, le nombre de nuits et la destination sont choisis par les clients.

On demande d'écrire un programme qui calcule et édite automatiquement la facture d'un groupe de participants à un même voyage (les options du voyage sont identiques pour chaque membre du groupe).

Chaque facture comporte 3 parties : la première concerne le transport, la seconde le séjour, et la troisième des frais de dossier.

Facturation du transport :

Il y a trois destinations possibles : Rome, Florence ou Venise.

Le départ se fait en train, au départ de Nancy. L'aller et le retour par personne est de :
710 € pour Rome en 2ème classe

1160 € - - en 1ère classe

630 € pour Florence en 2ème classe

1010 € - - en 1ère classe

620 € pour Venise en 2ème classe (pas de 1ère classe pour Venise)

Il convient de rajouter (obligatoirement) aux tarifs ci-dessus, au choix, l'une des deux réservations suivantes :

place assise : 22 (aller et retour)

couchette : 142 (aller et retour)

Facturation du séjour

Les personnes ont le choix entre deux catégories d'hôtel, le prix de la nuit est de :

130 par personne en hôtel 2ème catégorie

220 par personne en hôtel 1ère catégorie

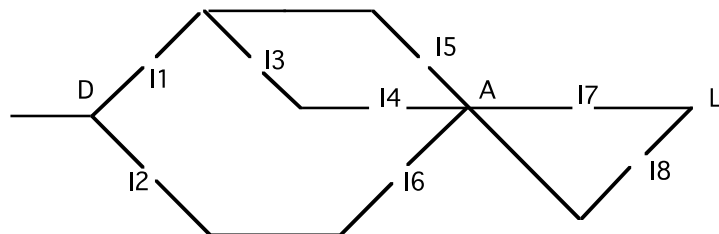
Frais de dossier

Il sera perçu 100 par dossier établi (un seul dossier par groupe de personnes voyageant ensemble). Si le nombre de personnes est supérieur à 20, une réduction de 3 par enfant participant au voyage est accordée. Si le nombre de personnes est inférieur à 10, il y a une majoration de 2 par adulte.

Écrire un programme qui édite la facture pour un groupe de personnes s'inscrivant à un voyage.

Exercice 8 : Circuit

Soit le circuit représenté par le schéma suivant :



Le point D est toujours alimenté en électricité. Les I1, I2, ...I8 représentent des interrupteurs pouvant être ouverts ou fermés.

Exercices

Écrire un programme qui, selon l'état des interrupteurs, indique si le courant arrive au point L. Pour représenter les interrupteurs on peut utiliser des variables booléennes.

Par exemple, $I1 = 1$ si $I1$ est fermé et $I1 = 0$ si $I1$ est fermé

LES FONCTIONS

Exercice 9 : Diviseur

Ecrire une fonction qui retourne vrai si d est diviseur de x, d et x étant donnés en paramètres.

Exemple : *diviseur(5, 10) retourne vrai*
diviseur(5, 11) retourne faux

Exercice 10 : Fonction arrondi_bancaire

- Ecrire une fonction *decimale_n* qui retourne la *n*ème décimale d'un nombre réel *x*, *n* et *x* étant passés en paramètres. Il est fortement conseillé de travailler au maximum et dès le départ avec des entiers.
- Écrire en python une fonction *arrondi_pair* qui prend en entrée un nombre entier positif *n* et qui retourne le nombre pair immédiatement supérieur ou égal à *n*.

Par exemple :

arrondi_pair(5) retourne 6

arrondi_pair(6) retourne 6

arrondi_pair(7) retourne 8

- La méthode dite *Méthode d'arrondissement au pair le plus proche, arrondi bancaire* ou encore *arrondi statistique* fonctionne suivant le principe suivant :
 - On précise la décimale *d* à laquelle le nombre doit être arrondi et la décimale *dd* qui suit la décimale *d*.
 - Si *dd* est inférieur ou égal à quatre, alors la décimale *d* reste inchangée.
 - Alors que si *dd* est six ou plus, la décimale est augmentée d'une unité.
 - Enfin si *dd* est le chiffre cinq lui-même suivi par des chiffres différents de zéro, la décimale sera augmentée d'une unité, tandis que si cinq n'est suivi d'aucun chiffre (ou que par des zéros) alors on applique un arrondi pair sur *d*.

Cette méthode est employée afin d'éliminer le biais qui surviendrait en arrondissant à chaque fois par excès les nombres dont le dernier chiffre est cinq.

Par exemple pour un arrondi au centième :

3.2464 est arrondi à 3.25

3.54398 est arrondi à 3.54

3.2356 est arrondi à 3.24

3.245 est arrondi à 3.24 parce que 4 est pair

3.715 est arrondi à 3.72 parce que 1 est impair

Ecrire une fonction *arrondi_bancaire* qui calcule l'arrondi bancaire à 2 chiffres après la virgule d'un nombre positif à 3 décimales. Cette fonction utilisera la fonction *arrondi_pair*.

Indication : une technique consiste à faire les calculs en entier puis à repasser en réel. On peut aussi utiliser la fonction *round* de Python

BOUCLES FOR

Exercice 11 : Impression des n premiers entiers pairs

Écrire un programme qui imprime les n premiers entiers pairs, n étant donné.

Exercice 12 : Diviseurs d'un nombre donné

Écrire un programme qui affiche tous les diviseurs d'un nombre entier donné.

Exercice 13 : Suite récurrente

Calculer pour x et n donnés le nième terme de la suite u_i définie par :

$$\begin{cases} u_0 = 1 \\ u_i = -2u_{i-1} / x \end{cases}$$

Exercice 14 : Min, Max et Moyenne d'une suite de n nombres

Ecrire un programme qui demande successivement n nombres à l'utilisateur, n étant donné par l'utilisateur, et qui imprime le minimum, le maximum et la moyenne des ces valeurs.

Exercice 15 : Affichage de la table des caractères ASCII

- 1) Imprimer la liste des codes ASCII de 32 à 255 et en face de chaque code imprimer le caractère correspondant (les 32 premiers caractères ne sont pas imprimables).

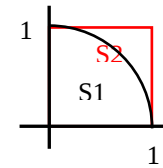
Aide : Utiliser la fonction *chr(i)* qui retourne le caractère dont le code ASCII est i.

- 2) Imprimer sous forme d'une grille de 22 lignes et de 10 colonnes les caractères dont le code ASCII varie de 32 à 251.

Exercices

Exercice 16 : Calcul de PI par la méthode de Monte-Carlo

Calculer Pi par la méthode de Monte-Carlo. La méthode de Monte-Carlo est une méthode pour le calcul de l'aire d'une courbe reposant sur l'utilisation de nombres aléatoires. Soit le schéma suivant :



Le principe consiste à tirer aléatoirement n couples d'abscisses et d'ordonnées comprises entre 0 et 1 et à comptabiliser le nombre *nbp* des points correspondant qui sont dans la surface S1.

Comme on a : $S1 = (\pi \cdot r^2) / 4 = \pi / 4$ (puisque $r = 1$)
 et $S2 = 1$ (surface totale du carré)
 on a donc : $\pi = 4 \cdot S1 = 4 \cdot S1 / S2$

Les surfaces S1 et S2 étant approximées respectivement par *nbp* et n, le calcul de π se fait donc par la formule : $\pi = (4 * nbp) / n$.

Ecrire un programme qui calcule une approximation de la valeur de PI par cette méthode.

BOUCLES WHILE

Exercice 17 : Carré de n

Ecrire un programme qui retourne le plus petit entier n tel que $n^2 > k$, k étant donné.

Exercice 18 : Puissance de x

Écrire un programme qui affiche toutes les puissances du nombre x qui sont $\leq n$, n étant donné. Par exemple : 5 25 125 625 sont les puissances de 5 ≤ 1000 sont les

Exercice 19 : Recherche du plus petit diviseur d'un entier donné

Ecrire un programme qui recherche le plus petit diviseur autre que 1 de n donné en paramètre.

Exercice 20 : Nombre premier

Ecrire une fonction qui recherche si n est premier ou non. Cette fonction retourne *True* si n passé en paramètre est premier et *False* sinon.

Indications :

- x est premier si les restes des divisions $x/2, x/3, \dots, x/E(\sqrt{x})$ sont tous non nuls.
- Utiliser la fonction `sqrt` (racine carrée) disponible dans la bibliothèque de fonctions mathématiques de Python.

Exercice 21 : Calcul de la valeur approchée de $\sqrt{a}$

Pour tout nombre positif a , on peut trouver une suite infinie qui converge vers $\sqrt{a}$:

$$r_n = (r_{n-1} + a/r_{n-1}) / 2 \text{ pour } n \geq 1 \text{ avec } r_0 \text{ quelconque positif}$$

Ecrire une **fonction** qui calcule et **retourne** une approximation de $\sqrt{a}$, le dernier terme $r_{\max}$ de la suite des approximations étant caractérisé par :

$$|r_{\max} * r_{\max} - a| < \text{epsilon, epsilon donné}$$

Pour calculer la valeur absolue, on peut utiliser la fonction *abs* disponible dans Python.

Exercice 22 : Recherche du pgcd de deux entiers naturels

Etant donné deux entiers naturels a et b , écrire une fonction qui calcule leur plus grand diviseur commun.

On pourra utiliser pour cela soit la propriété suivante du pgcd, soit le programme d'Euclide rappelé plus loin :

$$\begin{aligned} 1. \quad PGCD(a,b) &= PGCD(a-b,b) && \text{si } a > b \\ &\text{ou} && PGCD(a,b-a) && \text{si } b > a \\ PGCD(a,a) &= a \end{aligned}$$

2. Algorithme d'Euclide :

Soit a et b deux entiers naturels, si b est non nul, on peut effectuer la division euclidienne de a par b . Il existe un couple unique d'entiers (q,r) tels que :

$$a = bq + r \text{ et } 0 \leq r < b$$

$$\text{si } r = 0 \text{ alors } b \text{ divise } a \text{ et } PGCD(a,b) = b$$

si $r > 0$ alors tout diviseur commun à a et b est diviseur de r et réciproquement tout diviseur commun à b et r est diviseur de a . On a donc $PGCD(a,b) = PGCD(b,r)$.

Exercice 23 : Recherche d'un nombre tiré au hasard

Écrire un programme qui fasse deviner à l'utilisateur un nombre tiré au hasard et affiche le nombre d'essais effectués par le joueur. A chaque proposition de l'utilisateur, l'ordinateur doit préciser si le nombre proposé est plus petit ou plus grand que le nombre à deviner.

Variante : limiter le nombre de coups autorisés.