

ECOLE SUPERIEURE D'INFORMATIQUE DE GESTION



Travail d'Algorithmes et Langage de Programmation Septembre 2022

Informations générales

Branche : ALGORITHMES ET LANGAGE DE PROGRAMMATION (ALP)

Classes : ESIG 1 Classes A et B (groupes 1, 2 et 3)

Date : 21 septembre 2022

Modalités

1. Durée : 180 minutes en tout (parties 1 et 2).
2. Travail individuel.
3. Documentation personnelle (livres, papiers et électronique) : Autorisée uniquement pour la partie 2 de l'épreuve (cf. ci-dessous).
Aucune documentation n'est autorisée pour la partie 1.
4. La partie 1 dure 60 minutes au maximum et 30 minutes au moins. **Aucune sortie** aux toilettes n'est autorisée pendant la première heure. Tout le temps restant après la reddition de la partie 1 pourra être mis à profit pour la partie 2.
5. Tout partage de ressources de votre poste de travail avec le réseau ainsi que toute autre tentative de communication seront considérés comme de la fraude et sanctionnés comme tels par la note minimale. Cela comprend la présence de clés USB personnelles, d'ordinateurs personnels, smart watches ou similaires à votre place de travail.

Démarrage et préparation

1. Connectez-vous au réseau sur votre poste de travail.
2. **Avant** le commencement de l'épreuve, votre documentation électronique (disque externe, clé USB, ...) : doit être recopiée sur **C:\ESIGUsers** dans un répertoire à votre nom.
3. **Une fois l'épreuve commencée**, copiez dans **C:\ESIGUsers** le dossier réseau qui vous sera indiqué au début de l'épreuve, normalement
G:\ESII-Uldry\ESIG\Distribution\2022-2023\ESIG-1\ALP\ALP-septembre-22
de façon à avoir un répertoire **C:\ESIGUsers\ALP-septembre-22**.
4. Ouvrez ce dossier **C:\ESIGUsers\ALP-septembre-22**.
5. Ouvrez Thonny puis ouvrez (dans Thonny) les fichiers **.py** qui se trouvent dans **C:\ESIGUsers\ALP-septembre-22**.
6. **Enregistrez une copie (Enregistrer sous) de ces deux fichiers en ajoutant vos nom et prénom à la fin.**
7. Ensuite, restez connecté mais éteignez l'écran. L'écran pourra être rallumé pour la partie 2.

Partie 1 : Débranchée

Nom étudiant-e :

Prénom étudiant-e :

Professeur :



Si vous détachez les feuilles de l'énoncé Partie 1 :
écrivez **sur chaque page** vos **nom et prénom**.

Exercice – Lire du code Python

Lisez les programmes ci-dessous. Pour chacun des deux, indiquez ce qu'il affiche (les sorties produites via les instructions **print**) dans l'ordre de l'exécution.

A

```
1 def t():
2     c()
3     print("Do")
4     c()
5
6 def q():
7     print("Ré")
8
9 def c():
10    q()
11    print("Mi")
12
13 print("Fa")
14 d = True
15 e = 4
16 if e < 4:
17     print("Sol")
18     t()
19 else:
20     e -= 1
21     if d == False or e <= 3:
22         print("La")
23         c()
24     elif d == True and e >= 3:
25         t()
26     else:
27         q()
28     print("Si")
```

Sorties produites

B

```
1 def f(a,b):
2     print(a-b)
3
4 def g(a,b,c):
5     if a < 5:
6         h(b+5)
7     else:
8         h(a+2)
9     print(c*2)
10    f(b,a)
11    x = 5
12
13 def h(a):
14     b = a+2
15     print(b)
16
17 b=5
18 x=6
19 g(4,b,"6")
20 print(x)
```

Sorties produites

C

```
1 def i(u):
2     print('i(', u, ')')
3     return k(u+1)
4
5 def j(v):
6     print('j(', v, ')')
7     return 5
8
9 def k(w):
10    print('k(', w, ')')
11    v = j(w-1)
12    print('v', v)
13    return v-1
14
15 print(i(3) + 2)
```

Sorties produites

D

```
1 def k(x):
2     print("k(", x, ")")
3     return x*3
4     print(x)
5
6 def l(y,z):
7     print("l(",y,"",z,"")")
8     y += k(z)
9     a = m(y)
10    print("a",a)
11
12 def m(x):
13    print("m(", x, ")")
14    if x >= 10:
15        return x+5
16    elif x >= 15:
17        return x+10
18    print("x",x)
19    return x
20
21 x = 5
22 y = 10
23 z = 15
24 y += m(y)
25 l(z,y)
```

Sorties produites

Partie 2 : Programmation Python avec Thonny

Nom étudiant·e :

Prénom étudiant·e :

Professeur :

Numéro du poste :

Numéro de la clé USB:



La documentation personnelle est autorisée pour cette partie.

N'oubliez pas de sauvegarder régulièrement votre travail.

N'oubliez pas de mettre vos noms et prénoms au début des 2 fichiers

Exercice 1 – Sapin de Noël

Un magasin de jouets pour enfants a décidé de mettre en place, durant la période hivernale, une activité ludique permettant aux clients de dessiner un sapin de Noël au travers d'un programme informatique.

a) Flèches successives

Pour dessiner le sapin de Noël, il faut premièrement dessiner une flèche. En effet, ledit sapin est composé de flèches qui se succèdent (= qui sont mises bout à bout).

Le mandant souhaite avoir une procédure `flèche` qui recevra la longueur des segments droits (x) ainsi qu'un angle (α) et qui dessinera une flèche en fonction de ces critères. Cette procédure servira à dessiner le sapin.

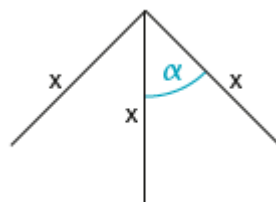


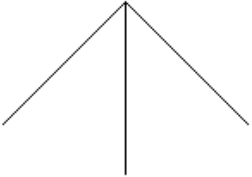
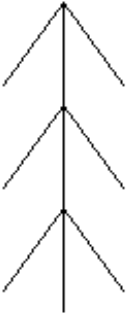

Illustration d'une flèche avec segments droits de longueur x et un angle α

Il souhaite également avoir une procédure `sapin` qui recevra le nombre de flèches à dessiner, la longueur des segments droits ainsi qu'un angle et qui dessinera le sapin en entier.

Le programme que vous devez écrire dessine le sapin conformément aux règles énoncées ci-dessus sur la base du nombre de flèches, de la longueur des segments droits et d'un angle. De plus, vous devez respecter le format d'affichage représenté ci-dessous (voir **Résultats**).

On part du principe que toutes les données saisies par l'utilisateur sont correctes.

Résultats

		
<i>Sapin avec 1 flèche de longueur 100 et un angle de 45°</i>	<i>Sapin avec 3 flèches de longueur 50 et un angle de 36°</i>	<i>Sapin avec 7 flèches de longueur 30 et un angle de 25°</i>

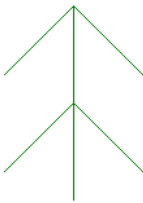
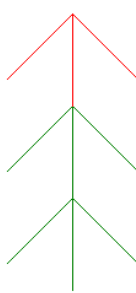

b) Alternance de couleurs

Complétez le programme créé au point précédent afin d'ajouter un peu de couleurs au sapin de Noël.

Le mandant souhaite que le sapin soit principalement dessiné en vert (`'green'`) mais que chaque trois flèches, une couleur choisie par le client soit utilisée à la place de la couleur verte.

Attention, les premières flèches du sapin sont toujours de couleur verte (`'green'`).

Résultats (voir PDF pour une version en couleur)

		
<i>Sapin avec 2 flèches de longueur 75, un angle de 45° et une couleur <code>'red'</code></i>	<i>Sapin avec 3 flèches de longueur 75, un angle de 45° et une couleur <code>'red'</code></i>	<i>Sapin avec 7 flèches de longueur 30, un angle de 25° et une couleur <code>'yellow'</code></i>

Remarque :

- Les sapins ont été dessinés de bas en haut.

Contraintes impératives

Le fichier `SapinNoel.py` complété avec vos nom et prénom contiendra le programme qui résout le problème décrit ci-dessus.

Pour le développement de votre solution, vous respecterez les conventions de formation des identificateurs et de disposition du code que nous avons définies en cours. Vous respecterez également la syntaxe Python 3.0 et ferez attention à ne pas laisser de code ni de déclarations inutiles.

Les tâches spécifiques identifiées seront implémentées sous forme de procédures ou fonctions dont les paramètres seront soigneusement choisis.

Hormis la déclaration d'éventuels imports, procédures, fonctions ou constantes aucun élément ne doit être déclaré globalement.

La procédure `main()` doit être le point d'entrée de votre programme.

Vous respecterez impérativement la structure utilisée dans le cadre du cours pour la création de scripts

Exercice 2 – Tunnel du Gothard

À la suite du percement du tunnel du Gothard, les CFF ont décidé de mettre en place un outil permettant aux futurs usagers de calculer le prix du billet pour une traversée de leur véhicule à bord du train qui effectue la navette.

Le système de tarification distingue trois catégories de véhicules différentes (numérotées 1, 2 et 3):

- La catégorie 1 représente les voitures.
- La catégorie 2 représente les camions sans remorque.
- La catégorie 3 représente les camions avec remorque.

Les règles de calcul du prix du billet sont les suivantes :

- Les voitures paient un tarif unique de 15 CHF.
- Les camions de moins de 4 tonnes paient également 15 CHF.
- Les camions de 4 tonnes et plus paient en fonction de leur poids:
 - pour un poids inférieur à 10 tonnes : 20 CHF
 - pour un poids compris entre 10 et 25 tonnes : 30 CHF
 - pour un poids compris entre 26 et 38 tonnes : 42 CHF
 - pour un poids supérieur à 38 tonnes : 50 CHF
- Les camions de 4 tonnes et plus peuvent aussi payer en fonction de leur longueur :
 - Si la longueur du camion est supérieure à 8 mètres, une taxe de 5 CHF est perçue par mètre supplémentaire ;
 - pour les autres, aucune taxe n'est perçue.
- Enfin, un supplément de 30% sur ce prix est demandé pour les camions avec remorque quels que soient leur poids et leur longueur.

Le programme que vous devez écrire effectue les calculs conformément aux les règles énoncées ci-dessus et affiche la catégorie du véhicule ainsi que le prix total du billet que l'utilisateur devra payer pour la traversée.

La catégorie du véhicule devra être affichée explicitement (voiture, camion sans remorque ou camion avec remorque).

On part du principe que toutes les données saisies par l'utilisateur sont correctes.

Contraintes impératives

Le fichier `TunnelGothard.py` complété avec vos nom et prénom contiendra le programme qui résout le problème décrit ci-dessus.

Pour le développement de votre solution, vous respecterez les conventions de formation des identificateurs et de disposition du code que nous avons définies en cours. Vous respecterez également la syntaxe Python 3.0 et ferez attention à ne pas laisser de code ni de déclarations inutiles.

Les tâches spécifiques identifiées seront implantées sous forme de procédures ou fonctions dont les paramètres seront soigneusement choisis.

Hormis la déclaration d'éventuels imports, procédures, fonctions ou constantes aucun élément ne doit être déclaré globalement.

La procédure `main()` doit être le point d'entrée de votre programme.

Vous respecterez impérativement la structure utilisée dans le cadre du cours pour la création de scripts

Résultats

Voici les résultats que vous devez obtenir pour un certain nombre de valeurs de test. L'obtention de ces résultats ne garantit évidemment pas que le programme que vous avez conçu est correct et complet. Vous devez également mener des tests d'exécution personnels, adaptés au code que vous avez développé.

Données saisies (valeurs de test)			Message à afficher
Catégorie	Longueur	Poids	
1	5	2	Le véhicule est une voiture Le prix total du billet est de 15 CHF
1	10	10	Le véhicule est une voiture Le prix total du billet est de 15 CHF
2	6	3	Le véhicule est un camion Le prix total du billet est de 15 CHF
2	6	4	Le véhicule est un camion Le prix total du billet est de 20 CHF
2	7	12	Le véhicule est un camion Le prix total du billet est de 30 CHF
2	12	7	Le véhicule est un camion Le prix total du billet est de 40 CHF
3	20	35	Le véhicule est un camion avec remorque Le prix total du billet est de 132.6 CHF
3	25	41	Le véhicule est un camion avec remorque Le prix total du billet est de 175.5 CHF

Remarques

- Le format des données saisies et affichées doit être impérativement respecté.