

ECOLE SUPERIEURE D'INFORMATIQUE DE GESTION



Travail d'Algorithmes et Langage de Programmation Septembre 2021

Informations générales

Branche : ALGORITHMES ET LANGAGE DE PROGRAMMATION (ALP)

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

Date : 29 septembre 2021

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\2021-2022\ESIG-1\ALP\ALP-septembre-21** de façon à avoir un répertoire **C:\ESIGUsers\ALP-septembre-21**.
4. Ouvrez ce dossier **C:\ESIGUsers\ALP-septembre-21**.
5. Ouvrez Thonny puis ouvrez (dans Thonny) les fichiers **.py** qui se trouvent dans **C:\ESIGUsers\ALP-septembre-21**.
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 :

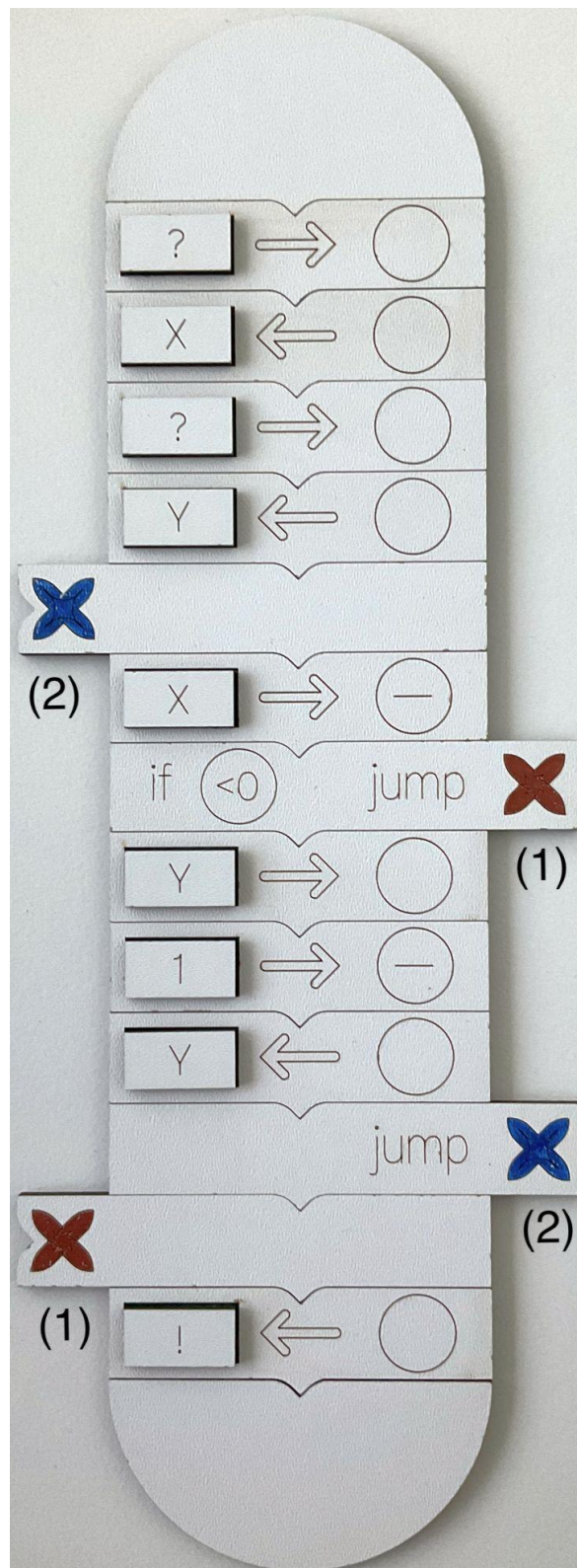


Si vous détachez les feuilles de l'énoncé Partie 1 :
écrivez **sur chaque page** vos **nom et prénom**.

Exercice 1 – Lire du code hp!

Voici un programme hp ! Remplissez les historams ci-dessous pour les entrées spécifiées.

Sur la dernière ligne de l'historam vous devez indiquer l'état final de toutes les zones de mémoire utilisées et de l'accumulateur.



Valeurs lues (dans cet ordre) : 5, 2

[illegible]

Valeurs lues (dans cet ordre) : 3, 4

[illegible]

Page vide

Exercice 2 – 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 a(x, y):
2      print('a(', x, ',', y, ')')
3      y = x + 1
4      print('y', y)
5      b(y, x)
6      print('x', x)
7      print('y', y)
8
9
10 def b(x, y):
11     print('b')
12     print('y', y)
13     print('x', x)
14     x += 1
15     print('x', x)
16
17
18 x = 10
19 y = -1
20 a(x, 5)
21 print('x', x)
22 print('y', y)
23
```

Sorties produites

B

```
1 def b(a):
2     print('b(', a, ')')
3     return a+a
4
5
6 def z(a):
7     print('z(', a, ')')
8     return a - 10
9
10
11 def a(c, d):
12     print('a(', c, ',', d, ')')
13     x = 0
14     if c < d:
15         x = b(c)
16     elif c > d:
17         x = b(d)
18     else:
19         x = z(c)
20     print(x)
21     return x
22
23
24 x = 10
25 x += a(x, 6)
26 y = a(x, x)
27 print('x', x)
28 print('y', y)
29
```

Sorties produites

Partie 2 : Programmation Python avec Thonny

Nom étudiant·e :

Prénom étudiant·e :

Professeur :

Numéro du poste :

Numéro de la clé USB :



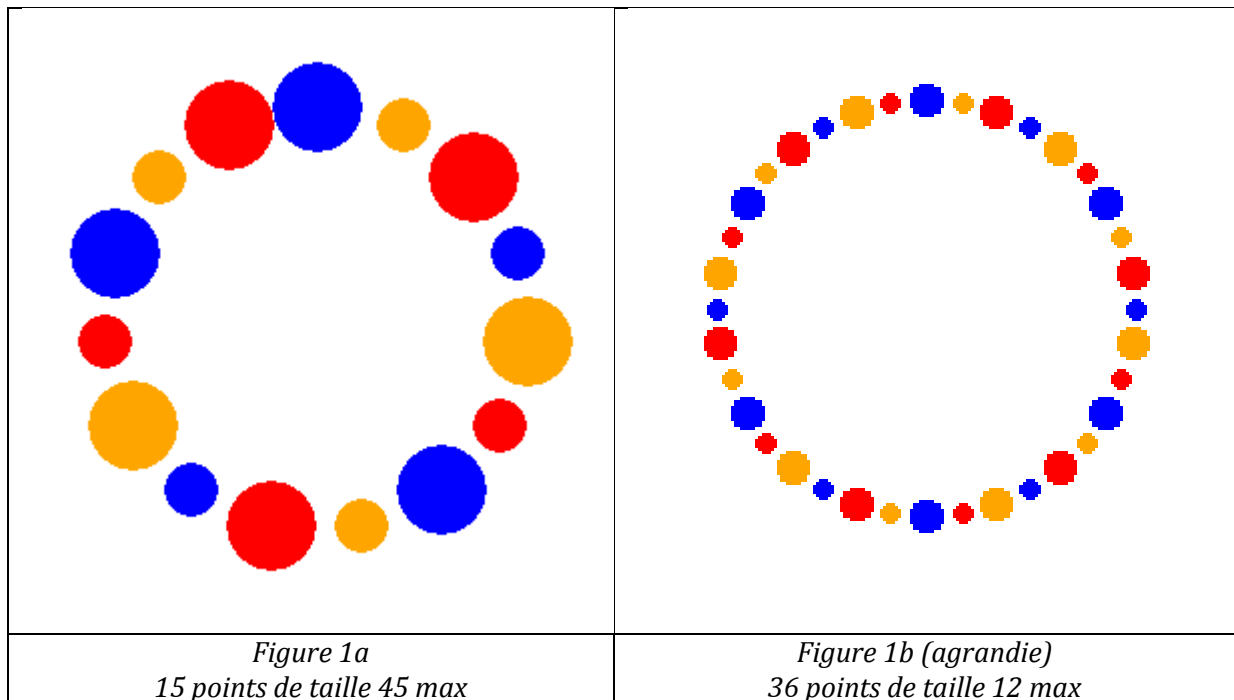
La documentation personnelle est autorisée pour cette partie.

N'oubliez pas de sauvegarder régulièrement votre travail.

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

Exercice 1 – Collier de perles -- Exercice tortue avec boucle

Il s'agit d'obtenir un graphique comme ceux-ci :



Pour y parvenir, nous allons procéder par étapes.

Le principe de base est de s'appuyer la méthode bien connue de dessin des polygones :

```
for c in range(nbCôtés):  
    forward(tailleCoté)  
    right(360/nbCôtés)
```

Vous retrouverez ce code (en commentaire) dans le fichier `CollierPerles.py` fourni et à compléter.

Si, avant d'appeler `forward()`, qui dessine un trait, nous utilisons `dot()` qui dessine un point, nous aurons ceci dans le cas où on choisit un nombre de côtés égal à 5 :

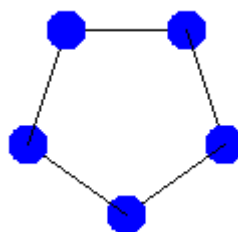


Figure 2

Remarque : on a conservé les traits dans ces exemples pour que ce soit plus facile à comprendre mais ils ne sont pas nécessaires dans cet exercice. (Si vous voulez les voir dans votre programme, faites un `pendown()` après l'appel à `my_init()` dans le code fourni.)

La Figure 2 n'est pas satisfaisante car nous voulons plutôt quelque chose comme ceci :

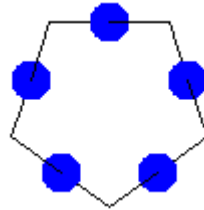


Figure 3

Pour cela il faut dessiner le point à mi-chemin. A vous de trouver comment ! L'image ci-dessous (tirée d'Astérix, Le bouclier arverne) vous aidera sans doute... Notez que la taille d'un côté est la même que celle d'un (gros) point.



Une fois qu'on a les points au bon endroit, il faut gérer les changements de taille et de couleur. Pour cela il faut appeler `my_dot()` (fournie) qui prend en paramètres la taille (comme `dot()`) mais aussi la couleur. Nous utiliserons ces 3 valeurs pour les couleurs : 'blue', 'orange' et 'red'.

Vous pouvez traiter le changement de taille et celui de couleur dans l'ordre que vous voulez (mais si vous traitez les deux en même temps, ce sera plus compliqué !). Par exemple, nous allons commencer par les changements de taille.

Si vous êtes observateurs, vous aurez remarqué qu'il n'y a que deux tailles, disons *grand* et *petit*, en alternance. Il suffit alors d'utiliser une des stratégies possibles pour alterner entre deux valeurs (par exemple un booléen qui est inversé ou la parité d'un nombre). La taille *petit* est 60% de la taille *grand*.

Pour le changement de couleur, il s'agit d'une alternance sur les 3 valeurs de couleur : 'blue', 'orange' et 'red'. Il est recommandé de définir une fonction pour cela qui renverra la bonne couleur à utiliser au tour de boucle suivant.

Cette fonction testera tous les cas possibles et renverra la couleur suivante dans l'enchaînement souhaité (à savoir 'orange' si c'est 'blue', 'red' si c'est 'orange' ou 'blue' si c'est 'red'). Vous aurez besoin d'une variable pour la couleur dans `polydot()`. Vous pouvez aussi travailler avec 3 entiers qui codent les 3 couleurs.

Quelques dernières recommandations :

- Respectez les zones à ne pas modifier dans le fichier fourni.
- Laissez en place la première instruction dans `polydot()`. Elle permet de ne pas voir la tortue et de déplacer son point de départ pour permettre des dessins plus larges.

Exercice 2 – Soirée Escalade

L'ESIG organise une soirée pour la fête de l'Escalade. Elle vous demande de lui fournir un petit outil permettant de calculer et d'afficher le prix à payer pour les entrées et boissons souhaitées.

Le prix dépend principalement du nombre de personnes, du fait qu'elles viennent déguisées (type n°1) ou non (type n°2), et du nombre de boissons supplémentaires commandées à l'entrée :

- Le prix d'entrée est de 15.- par personne, ou de 10.- par personne si elles sont toutes déguisées (lorsqu'un groupe achète plusieurs billets, les personnes doivent être TOUTES déguisées; dans le cas contraire, c'est le tarif de 15.- qui sera appliqué pour tout le groupe). Dans le cas où le groupe est déguisé, il bénéficie en plus d'une entrée gratuite dès 4 personnes.
- Lors de l'achat des billets d'entrée, on peut également commander des boissons supplémentaires au prix de 7.- la boisson.
- Pour chaque personne, un bon pour une boisson est automatiquement compris. Les boissons supplémentaires sont ajoutées. De plus, à partir de 50.- payés, on reçoit 1 bon pour une nouvelle boisson, et en plus 1 bon par tranche de 100.- payés. Autrement dit : 1 bon dès 50.- ; 2 bons dès 100.- ; 3 bons dès 200.- ; 4 bons dès 300.- ; etc.

Le programme que vous devez écrire effectue les calculs conformément aux règles énoncées ci-dessus. Vous devez respecter le format d'affichage représenté ci-dessous (voir la colonne « Message à afficher » dans Résultats).

Remarques :

- On part du principe que toutes les données saisies par l'utilisateur sont correctes.
- En Python, une division entière se fait avec l'opérateur `//` par exemple : `5 // 2` vaut 2.
- Et le reste de la division entière s'obtient avec `%` par exemple `5 % 2` vaut 1.

Contraintes impératives

Le fichier renommé 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
Nombre de personnes	Type de personne	Nombre de boissons supplémentaires commandées	
1	2	0	Entrée pour 1 personnes, Bon pour 1 boissons, Prix à payer : 15.-
2	2	0	Entrée pour 2 personnes, Bon pour 2 boissons, Prix à payer : 30.-
3	2	0	Entrée pour 3 personnes, Bon pour 3 boissons, Prix à payer : 45.-
4	2	0	Entrée pour 4 personnes, Bon pour <u>5</u> boissons, Prix à payer : 60.-
5	2	0	Entrée pour 5 personnes, Bon pour 6 boissons, Prix à payer : 75.-
1	1	0	Entrée pour 1 personnes, Bon pour 1 boissons, Prix à payer : 10.-
2	1	0	Entrée pour 2 personnes, Bon pour 2 boissons, Prix à payer : 20.-
3	1	0	Entrée pour 3 personnes, Bon pour 3 boissons, Prix à payer : 30.-
4	1	0	Entrée pour 4 personnes, Bon pour 4 boissons, Prix à payer : 30.-
5	1	0	Entrée pour 5 personnes, Bon pour 5 boissons, Prix à payer : 40.-
6	1	0	Entrée pour 6 personnes, Bon pour <u>7</u> boissons, Prix à payer : 50.-
2	2	1	Entrée pour 2 personnes, Bon pour 3 boissons, Prix à payer : 37.-
2	2	2	Entrée pour 2 personnes, Bon pour 4 boissons, Prix à payer : 44.-
2	2	3	Entrée pour 2 personnes, Bon pour <u>6</u> boissons, Prix à payer : 51.-
5	2	5	Entrée pour 5 personnes, Bon pour <u>12</u> boissons, Prix à payer : 110.-
5	1	5	Entrée pour 5 personnes, Bon pour <u>11</u> boissons, Prix à payer : 75.-
100	1	14	Entrée pour 100 personnes, Bon pour 125 boissons, Prix à payer : 1088.-
100	2	14	Entrée pour 100 personnes, Bon pour 130 boissons, Prix à payer : 1598.-

Remarques

- Le format des données saisies et affichées doit être impérativement respecté.
- Le soulignement de certains éléments de la colonne « Message à afficher » du tableau ci-dessus sert à attirer votre attention mais il ne doit évidemment pas être reproduit dans vos affichages.

Page vide