Contrôle 1 – NSI1 (105-109) – 25/09/20219

Première partie : sur ordinateur. **Durée : du début jusqu’à 15h30**. Les programmes devront être enregistrés sur le bureau, ou bien dans un dossier qui restera ouvert afin que je puisse les récupérer facilement sur clé, sans vous déranger, pendant la deuxième partie du contrôle (QCM de 20 minutes). Enregistrer le programme de l’exercice 1 sous Nom\_Prénom\_ex1.py (et idem avec 2 pour l’exercice 2 !). Mettre en 1ère ligne de commentaire votre email, si vous souhaitez une correction personnalisée de vos erreurs.

Pour cette partie du contrôle uniquement, vous avez droit au cours, aux exercices que vous avez déjà fait, et uniquement à la documentation Python en ligne <https://docs.python.org/3/index.html> (ceci dit elle ne devrait vous servir à rien sur ce contrôle). **Rien d’autre**, notamment sur le web. **Vérifiez que tout programme qui n’est pas en rapport avec ce qui précède est fermé** (attention aux ouvertures automatiques en début de session). Le non-respect de ces consignes est considéré comme une fraude, et promet des conséquences très néfastes sur la note finale (au minimum -5 points, en général 0)

*Les fonctions seront spécifiées.*

Exercice 1 (10 points):

Dans cet exercice, il vous est demandé de programmer une version simplifiée du black jack (jeu de casino).

*Règles :* Deux joueurs jouent l’un contre l’autre. L’un des joueurs est humain, l’autre est l’ordinateur. Le but du jeu est de marquer plus de points que l’autre, sans dépasser 21, qui est la plus grande valeur possible. Les cartes de 2 à 9 ont leur valeur (donc de 2 à 9 points…). Le 10, le valet, la dame et le roi valent 10 points. Dans un premier temps, l’as (le 1) vaut 11 points. Toutes les cartes sont tirées face visible.

* Les joueurs tirent d’abord une première carte, face visible.
* Ensuite, le joueur humain tire autant de cartes qu’il le veut. S’il dépasse 21, il a perdu.
* Le deuxième joueur -l’ordinateur- tire ensuite autant de cartes qu’il le veut. S’il dépasse 21, il a perdu.
* Si un des joueurs a plus de points que l’autre, il a gagné. Si les deux joueurs ont le même nombre de points, il y a égalité.

1. Écrire une première fonction qui tire un nombre aléatoire entre 1 et 13, et renvoie les points correspondant à la valeur de la carte. On considérera que 11, 12 et 13 correspondent respectivement au valet, à la dame et au roi. Enregistrer le programme sous Nom\_Prénom\_ex1.py
2. Écrire une deuxième fonction qui initialise le jeu en tirant une carte pour le joueur humain et une carte pour l’ordinateur. Cette fonction renverra les points correspondant à chaque carte. On rappelle que l’on peut faire un retour multiple sur les fonctions : truc, bidule = fonction(machin , chose, zaffair)
3. Écrire une troisième fonction qui fait jouer le joueur humain. Cette fonction renvoie le nombre de points du joueur humain.
4. Écrire une quatrième fonction qui fait jouer l’ordinateur. Vous programmerez une intelligence artificielle (rudimentaire ☺) qui décide de la stratégie de l’ordinateur. Vous expliquerez cette stratégie dans les spécifications de la fonction. Cette fonction renvoie le nombre de points de l’ordinateur.
5. Écrire le programme principal, qui organise la partie et détermine le vainqueur.
6. Règles complémentaires que vous pouvez implémenter (hors barème, très peu de points au vu du travail demandé, à peu près 0,01 point) :

* L’as vaut soit 1 soit 11 points.
* Faire 21 en deux cartes est gagnant sur 21 en trois cartes ou plus.
* Enchaîner plusieurs parties

Exercice 2 (4 points).

Écrire une fonction nombre\_de\_a(*chaîne*) qui demande une chaîne de caractères à l’utilisateur, et compte le nombre de caractères « a » ou « A » dans la chaîne. On rappelle qu’une chaîne est un type particulier de liste, que len(*chaîne*) donne la longueur de la chaîne et que chaîne[*i*] permet d’accéder à l’élément d’indice *i* de la chaîne. Si vous le souhaitez, plutôt que d’écrire une chaîne vous pouvez utiliser une liste (inconvénient : c’est plus long à taper). Enregistrer le programme sous Nom\_Prénom\_ex2.py.

*Exemples :*

* nombre\_de\_a("bonjour Ernest") renvoie 0
* nombre\_de\_a("Salut Bernard") renvoie 2
* nombre\_de\_a("Au revoir Anatole ") renvoie 3

Deuxième partie : 6 points, 20 minutes.

Cet exercice est un QCM où chaque question comporte 4 ou 5 réponses. Une seule est bonne. Toute bonne réponse rapporte 3 points, toute mauvaise réponse enlève 1 point, toute absence de réponse rapporte 0 point.

Répondre dans le cadre ci-dessous en mettant la lettre correspondant à la bonne réponse

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| NOM Prénom | Q1 | Q 2 | Q3 | Q4 | Q5 | Q6 |
|  |  |  |  |  |  |  |

1. On donne un programme faux, qui est censé donner les valeurs de  pour *x* de 1 à 20.

def parabole(x):

y =x\*\*2 + 3\*x + 1

x = 1

for i in range(20):

x = x + i

print(parabole(x))

Quelle version fonctionne correctement ?

|  |  |
| --- | --- |
| Code A | Code B |
| def parabole(x):  y =x\*\*2+3\*x + 1  return  x = 1  for i in range(21):  x = x + i  print(parabole(x)) | def parabole(x):  y =x\*\*2 + 3\*x + 1  return (x,y)  x = 1  for i in range(20):  x = i  print(parabole(x)) |
| Code C | Code D |
| def parabole(x):  y =x\*\*2 + 3\*x + 1  return (y)  x = 1  for i in range(1,21):  x = i  print(parabole(x)) | def parabole(x):  y =x\*\*2 + 3\*x + 1  return y  for i in range(20):  x = i  print(parabole(x)) |

1. Que donne l’exécution du programme suivant ?

x = 5

y = 4

if x == y - 1 :

y = x

else :

x = y

if y == x - 1 :

x = x + 1

else :

y = y + 1

A. *a* = 5 et *b* = 5 B. *a* = 4 et *b* = 4

C. *a* = 5 et *b* = 4 D. *a* = 4 et *b* = 5

E. autre réponse

1. Qu’affiche le programme suivant

Somme = 0

for i in range(5):

somme = i

print(somme, end = "")

A. 5 B. 4

C. 10 D. 0 1 2 3 4

1. Quel est le type de la variable nombre = "3.0" ?

A. string B. float

C. integer D. autre réponse

1. Quelle valeur contient la variable x après exécution du programme suivant ?

x = 400

while x >= 25 :

x = x//2

A. 25 B. 200 C. 12 D. 13 E. 12,5

1. Le programme suivant affiche :

somme = 0.1 + 0.1 + 0.1

if somme == 0.3 :

print("somme = ",somme)

else :

print("résultat = ",somme)

A. somme = 0.3

B. somme = 0.1 + 0.1 + 0.1

C. résultat = 0.1 + 0.1 + 0.1

D. résultat = 0.30000000000000004

E. un message d’erreur