
IE Informatique et Société Numérique 1

FC, AS, EUR, AMER

Décembre 2023



Durée: 1h

Documents et calculatrice interdits

Attention : Un programme **mal indenté**, ou avec de **mauvais choix de noms de variables** sera **sanctionné** (*jusqu'à -1 point*).

« Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live. » *Martin Golding*

Exercice 1 Correction de code (3 pts)

Le programme suivant contient 6 erreurs syntaxiques (pas d'erreurs de convention de codage). Identifiez chaque erreur en indiquant le numéro de ligne et le problème.

Voici un exemple de réponse attendue, indépendante du code ci-dessous :

ligne 5 -> le symbole '=' a été utilisé dans une comparaison, alors qu'il faut utiliser un '=='.

```
1 # Init des variables
2   revenu = 25000
3 nb_parts = 2
4 ratio = revenu / nb_parts
5 impot = 0
6
7 # Calcul de l'impot
8 if ratio < 20000:
9     impot = revenu*0.125 - 50*nb_part
10 elseif ratio < 30000:
11     impot = revenu*0.25 - 100*nb_parts
12 else
13     impot = revenu*0.5 - 200nb_parts
14
15 # Résultats
16 print(f"Vous devez payer impot euros")
```

```
1 # Init des variables
2   revenu = 25000 #err1: mauvaise indentation
3 nb_parts = 2
4 ratio = revenu / nb_parts
5 impot = 0
6
7 # Calcul de l'impot
8 if ratio < 20000:
9     impot = revenu*0.125 - 50*nb_part #err2: erreur sur le nom de la variable
10 elseif ratio < 30000: #err3: elseif au lieu de elif
```

```

11     impot = revenu*0.25 - 100*nb_parts
12 else #err4: il manque les ':'
13     impot = revenu*0.5 - 200nb_parts #err5: il manque '*' entre 200 et nb_parts
14
15 # Résultats
16 print(f"Vous devez payer impot euros") #err6: impot est une variable,
17                                     #donc mettre entre accolades

```

Exercice 2 Codage (2.5pts)

(Q2.1) Donnez la représentation **binaire** du nombre entier (non signé) : 102_{10}

Correction:

$102_{10} = 110\ 0110_2$

(Q2.2) Donnez la représentation **hexadécimale** du nombre binaire (non signé) : $1011\ 1001_2$

Correction:

$10111001_2 = B9_{16}$

(Q2.3) Donnez la représentation **décimale** du nombre binaire suivant : $1111\ 0011_2$

Correction:

$1111\ 0011_2 = 243_{10}$

Exercice 3 Lecture de code (2.5pts)

(Q3.1) Qu'affiche exactement le programme suivant :

```

1 liste = [3, 2, 7, 6, 10]
2 f = 1
3 for i in range(len(liste)):
4     if liste[i] % 3 == 0:
5         print(f'{f} {liste[i]//3}')
6     else:
7         print(f'f {liste[i-1]}')

```

Correction:

1 1
f 3
f 2
1 2
f 6

Exercice 4 Ecriture de Code (8pts)

Une fête forraine propose les activités suivantes : "pêche aux canards", "machine à grapin", "manège dancing Fly", "manège evolution", "manège Fire Ball". Les prix de ces activités sont : 2€, 1€, 4€, 4€ et 4€ respectivement.

(Q4.1) Ecrivez une fonction `prix_activite` qui pour une activité donnée en paramètre, renvoie en résultat le prix de cette activité.

```
1 # Correction 4.1
2 def prix_activite(activite):
3     prix = 0
4     if activite == "peche aux canards":
5         prix = 2
6     elif activite == "machine a grapin":
7         prix = 1
8     else:
9         prix = 4
10    return prix
```

(Q4.2) En faisant appel à la fonction `prix_activite`, écrivez un code permettant de calculer et d'afficher (avec une f-string) ce que dépense Rémi pour jouer successivement à la machine à grapin et au manège Fire Ball

```
1 # Correction 4.2
2 print(f'prix total payé = {prix_activite("machine a grapin") +
3       prix_activite("manège Fire Ball")}')
```

Théo arrive à la fête foraine avec une liste d'activités à faire. La liste d'activités est représentée par une liste.

Exemple :

```
1 activites = ["machine a grapin", "manège dancing fly",
2             "manège Fire Ball", "machine a grapin"]
```

(Q4.3) Ecrivez un code Python qui calcule et affiche le budget nécessaire pour satisfaire la liste de Théo. Ce code doit fonctionner quelle que soit la liste des activités choisies.

```
1 # Correction 4.3
2 activites = ["machine a grapin", "manège dancing fly", "manege Fire Ball",
3             "machine a grapin"]
4 budget = 0
5 for act in activites:
6     budget += prix_activite(act)
7 print(f'Theo a besoin de {budget} euros pour faire les activités choisies')
```

Les parents de Lou ont préféré acheter une carte prépayée pour maîtriser leur budget. Lou arrive à la fête foraine avec une liste d'activités qu'elle souhaite faire. Ses parents lui expliquent qu'elle pourra faire autant d'activités que le montant crédité sur la carte le permet. On suppose jusqu'à la fin de l'exercice, que Lou respecte l'ordre des activités dans sa liste. On souhaite écrire un code Python qui affiche les activités qui ont pu être réalisées. Votre code doit afficher combien d'activités ont été réalisées ainsi que le montant restant sur la carte quel que soit le nombre d'activités réalisées.

Exemple : avec la liste de Théo de la question (Q4.3) et un budget sur la carte de 5€, l'affichage obtenu serait :

manège dancing fly
2 activités sur 4
Il reste 0€

(Q4.4) Proposez dans l'ordre : (1) un jeu de tests du code à écrire (2) un code qui réalise ce qui est demandé.

Correction:

correction du (1)

carte = 30€ et liste = ["machine à grapin", "manège dancing fly", "manege Fire Ball", "machine à grapin"], Affichage de toutes les activités (4 activités sur 4), il reste 20 €.

carte = 10 € et ["machine à grapin", "manège dancing fly", "manege Fire Ball", "machine à grapin"], Affichage de toutes les activités (4 activités sur 4, il reste 0€

carte = 10 € et ["machine à grapin", "manège dancing fly", "manege Fire Ball", "manège evolution"], affichage de 3 activités sur 4, il reste 1 €

carte = 12€ et ["machine à grapin", "manège dancing fly", "manege Fire Ball", "machine à grapin"], affichage de 3 activités sur 4, il reste 2€

carte = 12€ et ["machine à grapin", "manège dancing fly", "manege Fire Ball", "manege Fire Ball", "machine à grapin"], affichage de 3 activités sur 5, il reste 3€

```
1 # Correction du (2)
2 i = 0
3 while i < len(activites) and carte >= prix_activite(activites[i]):
4     carte -= prix_activite(activites[i])
5     print(activites[i])
6     i = i + 1
7 print(f'Lou a pu faire {i}/{len(activites)} activités')
8 print(f'Il reste {carte} euros sur la carte')
```

Imane est arrivée avec la liste suivante : ["machine à grapin", "manège dancing fly", "manege Fire Ball", "manege Fire Ball", "machine à grapin"] et un montant de 12€. Mais elle constate qu'elle a pu faire seulement les 3 premières activités et il lui reste 3€ sur la carte à ce moment là, ce qui aurait pu lui permettre de faire l'activité "machine à grapin".

(Q4.5) Quel code python ajouter à la suite de celui de la question (Q4.4) pour afficher l'indice de la prochaine activité possible dans la liste (et -1 s'il n'y en a pas). Pour l'exemple ci-dessus, l'indice à afficher est : 4.

```
1 # Correction 4.5
2 while i < len(activites) and prix_activite(activites[i]) > carte:
3     i = i + 1
4 if i == len(activites):
5     indice = -1
6 else :
7     indice = i
8 print(indice)
```

Exercice 5 Ecriture d'algorithme (4pts)

Le jeu de Nim (aussi appelé Fan Tan en Chine ou Tiouk Tiouk en Afrique) est un jeu de stratégie pure à deux joueurs. Le principe du jeu pour cet exercice est le suivant :

Le jeu est constitué de N tas, numérotés de 0 à $N - 1$. Dans chaque tas τ , il y a k_τ pions, sachant que le nombre de pions peut être différents d'un tas à un autre.

Les joueurs jouent alternativement. Lorsque c'est son tour, un joueur indique le numéro du tas dans lequel il veut retirer des pions. Et une fois ce choix fait, il choisit le nombre de pions qu'il souhaite prendre dans le tas choisi (autant qu'il veut). On fera l'hypothèse pour cet exercice que les choix faits par les joueurs sont tout le temps valides, c'est à dire que vous ne décrivez pas les tests de vérification des choix faits.

Le gagnant est celui qui retire le dernier pion, tous tas confondus. Le perdant est donc celui qui doit jouer alors que tous les tas sont vides.

(Q5.1) Proposez, en langage naturel ou avec un graphique, un algorithme détaillé pour le jeu de Nim. Cet algorithme devra en particulier bien mettre en évidence vos choix de structures de contrôle.

Remarque : le niveau de détail attendu pour la réponse à cette question est comparable à l'algorithme suivant décrivant le tri par sélection :

```
1 # ENTREES DE L'ALGO
2 #   l : liste de nombres a trier
3 # DEROULE DE L'ALGO
4 # Pour chaque indice i_pivot de l
5 #   Chercher l'indice i_min du plus petit élément, a partir de l'indice i_pivot
6 #   Si nécessaire
7 #       échanger les éléments situés aux indices i_pivot et i_min
8 # Affichage du résultat
```

Correction:

VERSION TEXTUELLE

ENTREE :

- N nb de tas,
- jeu tableau de longueur N , où chaque valeur correspond au nombre de pions k_i pour le tas i
- J_1 et J_2 les deux joueurs

INITIALISATION

- le joueur J_1 commence, il devient actif
- l'état du jeu est affiché auprès des joueurs

DEROULE

Tant qu'il reste des pions dans le jeu :

- le joueur actif choisit un tas valable
- le joueur actif choisit un nombre de pions valable dans le tas choisi
- le jeu est mis à jour
- l'état du jeu est affiché auprès des joueurs
- on change de joueur

Le joueur actif a perdu.

VERSION GRAPHIQUE :

IE/AlgoNim.pdf