Correction du CC1 Informatique
J1 MI 1003, groupe B3, Université Bordeaux

Exercice 1 (8 pts) Quelles valeurs contiennent les variables `a`, `b`, `c`, `d`, `e`, `f`, `g` et `h` après les instructions suivantes ?

```
a = 12
b = 19 % 4
c = b % a
a = 9
d = (a == 12)
e = 3
f = (a != e) or (a == e + 6)
def tordu(a,b):
    return (a+b)//2
g = tordu(a,e)/2
e == 7
h = tordu(e,8)
```

Variable	Valeur finale
a	~~12~~ 9
b	3
c	3
d	False
e	3
f	True
g	3.0
h	5

Vous êtes pas mal à être tombés dans le piège `e == 7` qui n'est en aucun cas une affectation... L'instruction `e == 7` est un **test**, elle renvoie `True` si e = 7 (au sens mathématique), `False` sinon. En gros, cette ligne ne fait strictement rien, si ce n'est que de calculer un `False` dans le vide.

Parmi les autres erreurs :

- Non `3 % 12` n'est pas égal à `0` mais à `3`... (vu dans le cours)
- Seule la première lettre de `True` et `False` est en majuscule.
- Un petit bonus si vous avez pensé à mettre le `.0` au `3.0` (division à virgule).

Exercice 2 (6 pts)
Ecrire une procédure Python `divise60(n)` qui renvoie `True` si 60 est multiple de `n` (Exemples : 10, 12...), `False` sinon.

On a vu cet exercice en cours dans un cas plus général que 60... Peu ont su reproduire ce qu'on avait vu. Beaucoup ont utilisé des `if`, ce qui n'est pas interdit, mais qui est moins élégant et qui a provoqué des oublis de deux points.

Là encore une fois, il ne faut pas confondre `=` et `==`...

Première solution :

```
def divise60(n):
    return ( (60 % n) == 0 )
```

Deuxième solution : (moins élégante, `(60 % n) == 0` renvoie de base un booléen)

```
def divise60(n):
    if (60 % n) == 0 :
        return True
    else:
        return False
```

Troisième solution : (carrément moins élégante mais ça marche si on suppose que `n` ne peut être négatif)

```
def divise60(n):
    return (n==1) or (n==2) or (n==3) or (n==4) or (n==5) or (n==6)
    or  (n==10) or (n==12) or (n==20) or (n==30) or (n==60)
```

Exercice 3 (8 pts)

Le premier ministre du Groenland souhaite donner une conférence à Bordeaux sur le réchauffement climatique. Organisateur local de cette conférence, vous devez trouver des interprètes afin que la traduction soit assurée du Français au Groenlandais. Bien sûr, vous souhaitez recruter un minimum d'interprètes.

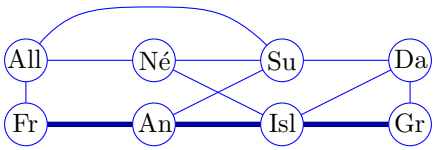
Les interprètes disponibles sur Bordeaux sont listés dans le tableau ci-contre. Notez que certains parlent trois langues, alors que d'autres uniquement deux.

Interprète	Traductions assurées
I1	Allemand-Néerlandais-Suédois
I2	Groenlandais-Islandais-Danois
I3	Français-Anglais
I4	Français-Allemand
I5	Suédois-Danois
I6	Anglais-Suédois
I7	Français-Allemand
I8	Anglais-Islandais
I9	Néerlandais-Islandais

Modélisez cette situation par un graphe : À quoi correspondent les sommets ? et les arêtes ? Comment se transpose le problème ?
Au final, quel est le nombre minimal d'interprètes à engager ?

Ici plusieurs solutions, je vous présente la mieux adaptée. C'était quasiment le même problème que les vols avec correspondance pour relier Bordeaux et Oulan Bator qu'on a vu en cours.

Les sommets correspondent aux langues. Deux sommets sont reliés par une arête si et seulement si un interprète assure la traduction entre les deux langues correspondantes. Il faut maintenant relier le français au groenlandais avec un minimum d'arêtes.



Il y a une solution avec 3 arêtes évidentes : Grolandais-Islandais, Islandais-Anglais, Anglais-Français. On ne peut pas faire avec moins car les sommets Grolandais et Français ne sont pas voisins et n'admettent pas de voisins communs. Le minimum est donc de 3.

Exercice 4 (Hors barème, à faire en dernier !)

Écrire une procédure Python `arrondimultiple(n,k)` sans utiliser `if` (si possible) qui renvoie le multiple de `k` le plus proche de `n`. Si deux multiples sont à égale distance de `n`, renvoyer le plus petit.

Personne n'a trouvé, je mets quand même la solution :

```
def arrondimultiple(n,k):
    resultat = ((n + k / 2) // k )*k
    return  int(resultat)
```

La commande `int` permet de transformer un nombre en virgule en entier. Sans elle, `arrondimultiple(14,3)` renverrait par exemple `15.0` au lieu de `15`. Bien sûr, je ne m'attendais pas à ce que vous l'utilisiez.