

CORRIGÉ ÉPREUVE REGROUPÉE

Discipline: INFORMATIQUE

Cours: 1 IN DP 06 / 07 / 08

Maître: JC Barros

Date: 04/06/2025

Durée: 105 minutes

Programmation

Ex 1.

/ 4 pts

- a. Écrire une fonction **carre** qui **renvoie** le carré du nombre donné en paramètre. (3 pts)

```
def carre(n):
    return n*n
```

- b. Qu'affiche le code `print(carre(carre(2)))`? (1 pt)

→ 16

Ex 2.

/ 4 pts

```
1 # Demander nb_final à l'utilisateur
2 nb_final = input(int('Entrer le nombre final:'))
3 # Créer liste avec tous nombres de 1 à nb_final
4 liste = []
5 for num in range(1, nb_final):
6     liste.append(nb_final)
7 # Calculer et afficher la somme des nombres
8 somme = sum(liste)
9 print('La somme des nombres est', sum)
```

Ligne	Modification(s)
2	<code>nb final</code> → <code>nb_final</code>
2	<code>input(int(</code> → <code>int(input(</code>
5	<code>nb_final</code> → <code>nb_final+1</code>
6	<code>nb_final</code> → <code>num</code>
9	<code>sum</code> → <code>somme</code>

Ex 3.

/ 10 pts

- a. Décrire en quelques mots le but (court) de **f1**, **f2** et **f3**: (3 pts)

f1 → retourne la valeur **True** si `val` se trouve dans `liste` et **False** sinon

f2 → retourne la valeur maximale qui se trouve dans `liste`

f3 → retourne la valeur moyenne des nombres dans `liste`

- b. Expliquer le fonctionnement (en détail) de **f1**, **f2** ou **f3**: (3 pts)

f1 → la boucle **for** donne chaque valeur de la liste à la variable `i`. A chaque fois, on vérifie si la valeur est égale à celle qu'on cherche (`val`). Si c'est le cas on **return True** (donc on quitte la fonction et on retourne la valeur **True**). Dans le cas contraire, si on n'a jamais d'égalité, on arrive à la fin de la boucle puis on en sort. On retourne alors la valeur **False** car on n'a pas trouvé `val` dans `liste`.

f2 → on stocke dans la variable les candidats au titre de valeur maximale dans `liste`. Le premier d'entre eux n'est autre que le premier élément de `liste`: `m = liste[0]`. Ensuite, avec la boucle **for** `i in liste` : , on passe en revue chaque valeur dans la liste (`i`) qu'on compare avec notre candidat actuel: **if** `i > m`: s'il est supérieur, on stocke cette valeur (`i`) dans `m` et on recommence jusqu'au moment où on a essayé avec toutes les valeurs de la liste (fin de la boucle **for**). On est alors sûr d'avoir le maximum dans `m` et on peut le **return**.

f3 → on initialise la variable `somme` avec la valeur 0. On démarre alors une boucle (**for** `i in liste` :) où `i` va prendre successivement chaque valeur de `liste`. A chaque passage, on ajoute cette valeur à la variable `somme` (`somme += i`). Finalement, après la fin de la boucle, `somme` contient donc la somme de tous les éléments de la liste et on **return** cette somme divisée par le nombre d'éléments de la liste (**return** `somme / len(liste)`), ce qui correspond à la moyenne des valeurs de liste.

c. A quelle ligne démarre le programme principal? (1 pt)

Le programme principal est la partie du code où on exécute des instructions. Il vient donc après les définitions des fonctions, à la ligne 23

d. Qu'affiche le programme (exactement) ? (3 pts)

il y a 3 **print()**, donc trois lignes vont s'afficher:

False
10
5

PROGRAMME PYTHON:

```

1  def f1(liste, val):
2      for i in liste:
3          if i == val:
4              return True
5      return False
6
7
8  def f2(liste):
9      m = liste[0]
10     for i in liste:
11         if i > m:
12             m = i
13     return m
14
15
16 def f3(liste):
17     somme = 0
18     for i in liste:
19         somme += i
20     return somme / len(liste)
21
22
23 liste_pairs_desordre = [2, 6, 10, 4, 8, 0]
24
25 print(f1(liste_pairs_desordre, 3))
26 print(f2(liste_pairs_desordre))
27 print(f3(liste_pairs_desordre))

```

barème

points	note
0	1,5
11,5	2
19	2,5
27	3
34,5	3,5
42	4
49,5	4,5
57	5
65	5,5
72,5	6

Ex 4.

/ 2 pts

Qu'est-ce que les instructions suivantes affichent à l'écran ?

```
1 ma_liste = ['apple', 'banana', 'cherry']
2 print(ma_liste[1])
3 print(ma_liste[-1])
```

banana
cherry

Ex 5.

/ 2 pts

Ajouter une deuxième ligne à ce programme pour qu'il affiche le nombre d'éléments de `ma_liste` (dans ce cas précis: 3).

```
1 ma_liste = ['apple', 'banana', 'cherry']
2 print(len(ma_liste))
```

Ex 6.

/ 12 pts

Remplir le tableau selon l'**exemple**:

programmes	Affichage à l'écran (texte exact!)	valeurs des variables (écrire toutes les valeurs successives que prennent les variable et entourer la valeur finale)
<pre>nombre = 0 somme = 0 while nombre < 4: nombre += 1 print(nombre) somme += nombre print(somme)</pre>	<pre>1 2 3 4 10</pre>	<pre>nombre : 0 / 1 / 2 / 3 / 4 somme : 0 / 1 / 3 / 6 / 10</pre>
<pre>nombre = 0 somme = 0 while nombre < 4: print(nombre) nombre += 1 somme += nombre print(somme)</pre>	<pre>0 1 2 3 10</pre>	<pre>nombre : 0 / 1 / 2 / 3 / 4 somme : 0 / 1 / 3 / 6 / 10</pre>
<pre>somme = 0 for nombre in range(4): somme += nombre print(nombre) print(somme)</pre>	<pre>0 1 2 3 6</pre>	<pre>nombre : 0 / 1 / 2 / 3 somme : 0 / 0 / 1 / 3 / 6</pre>

Réseaux

Ex 7. / 1 pt

Comment s'appelle l'élément réseau qui permet de connecter plusieurs réseaux locaux ?

routeur

Ex 8. / 2 pts

Quel est le sigle de trois lettres qui représente un réseau local (de taille modeste) ? (1 pt)

LAN

Quelle est la signification de chaque lettre ? (1 pt)

Local Area Network (réseau de taille petite)

Ex 9. / 4 pts

Citer deux protocoles utilisés dans la couche **transport** du modèle OSI (2 pts)

- a. TCP
- b. UDP

Citer deux différences entre eux? (par exemple, "a est plus ... que b") (2 pts)

- Les en-têtes **TCP** sont **plus** longues (**lent**) **que** celles d'**UDP**.
- **TCP** est **plus fiable** (connexion, renvoi des paquets perdus, etc.) **que UDP**.

Ex 10. / 2 pts

Expliquer la fonction d'un **routeur** en quelques lignes (donner 2 informations pertinentes).

- Le routeur route les paquets en fonction de l'adresse IP de destination de ces paquets.
- Ils doivent donc décapsuler les en-têtes des couches 1, 2 et 3 pour voir cette adresse.
- Ensuite, grâce à leur table de routage, ils envoient le paquet à l'un des routeurs voisins.
- Si le paquet a un TTL (Time To Live) égal à 0, le routeur le supprime.
- etc.

Ex 11. / 4 pts

Dans le protocole IPv4:

- a. Combien d'octets composent une adresse ? (1 pts)

4

- b. Combien d'adresses IP différentes peut-on utiliser dans le système IPv4 ? (1 pt)

2^{32} (2 à la puissance 32 car il y a 32 bits au total), ce qui fait 4 294 967 296.

- c. Est-ce que c'est suffisant pour l'internet actuel ? (2 pts)

Non, il y a beaucoup plus d'objets connectés sur l'internet (serveur, ordinateurs, smartphones, montres et autres objets connectés)

On contourne ce problème de plusieurs manières:

- en passant à la version IPv6 qui propose plus de bits
- en utilisant les adresses publiques pour les réseaux locaux, ce qui permet de n'utiliser qu'une seule adresse publique (unique dans le monde) pour tout un LAN.
- en jonglant avec les masques de réseaux pour distinguer, par exemple, l'hôte 114 dans le réseau 192.168.1.0 (192.168.1.114/24) de l'hôte 1.114 dans le réseau 192.168.0.0 (192.168.1.114/16).

Ex 12. / 3 pts

- a. Que signifie « HTTP » ? (1 pt) Entoure la bonne réponse.

- High Transmission Text Protocol
- Hyperlink Text Transfer Program
- **HyperText Transfer Protocol**
- Host Transfer and Text Process

- b. A quoi sert HTTP ? (1pt)

c'est le protocole (= la langue) qu'utilisent serveurs web et clients web pour communiquer.

- c. Comment s'appelle l'application installée sur votre ordinateur qui utilise HTTP ? (1 pt)

Le **navigateur web** (Firefox, Chrome, etc.) est le client web installé sur votre ordinateur.

Ex 13.

/ 1 pts

À quoi sert le DNS dans un réseau ? (entourer la bonne réponse)

- a. **À traduire les noms de domaine en adresses IP**
- b. À stocker les mots de passe des utilisateurs
- c. À connecter les imprimantes au réseau
- d. À crypter les données échangées

Ex 14.

/ 2 pts

Expliquer brièvement la différence entre les architectures "client-serveur" et "pair à pair" (p2p)
L'architecture **client-serveur** est hiérarchique: Le serveur détient les informations et est toujours disponible sur le réseau. Le client, lui, se connecte quand il veut et sollicite des informations au serveur.

Dans l'architecture **pair à pair** (peer to peer ou p2p), tous les participants sont égaux. Pas de hiérarchie. Chacun détient des informations et peut en solliciter aux autres. C'est un partage décentralisé.

Ex 15.

/ 1 pt

Quelle est la fonction principale d'un switch (commutateur) ?

- a. ~~Connecter plusieurs réseaux différents et choisir le meilleur chemin pour chaque paquet~~
- b. **Connecter plusieurs appareils au sein d'un même réseau local.**
- c. ~~Fournir une connexion Internet à tous les appareils~~
- d. Protéger le réseau contre les virus

Ex 16.

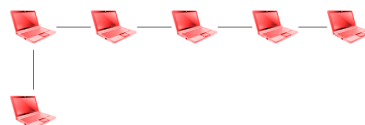
/ 4 pts

- Un navigateur qui demande une page web à un serveur doit **ENCAPSULER** sa demande avant qu'elle ne parte sur l'internet.
- Un serveur web qui répond au navigateur en envoyant le contenu d'une page doit **ENCAPSULER** la réponse pour qu'elle puisse repartir sur les lignes puis atteindre l'ordinateur où le navigateur l'attend.
- Un commutateur qui reçoit une trame (message) doit **DÉCAPSULER** la trame pour connaître l'adresse du destinataire de ce message.
- Un routeur qui renvoie un paquet (message) doit **ENCAPSULER** le paquet pour qu'il puisse continuer sa route.

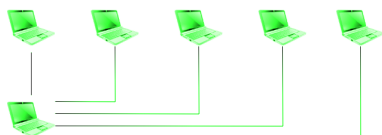
Ex 17.

/ 8 pts

- a. Indiquer le nom de la topologie représentée à côté de chaque diagramme. (4 pts)

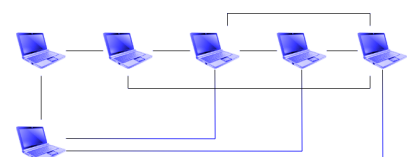


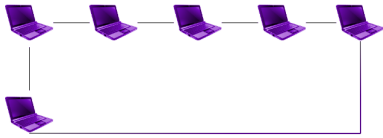
Linéaire (ou bus)



Etoile

partiellement maillé (mesh)





Anneau

- b. Quelle topologie est la plus adaptée aux réseaux locaux? (1 pt)
étoile
- c. Quelle topologie est mise en place dans l'internet? (1 pt)
partiellement maillé
- d. Donner un avantage et un inconvénient de la topologie en anneau: (2 pts)
Avantage: peu de câbles (**bon marché**) / **routage simple**
Inconvénients: **confidentialité** nulle / **fragilité** en cas de panne d'un noeud

Sécurité

Ex 18.

/ 7 pts

Application malveillante opérant un chantage en échange de la récupération de données ou d'un service

Récupération de données personnelles (mot de passe ou autre) par un pirate se faisant passer pour une société connue

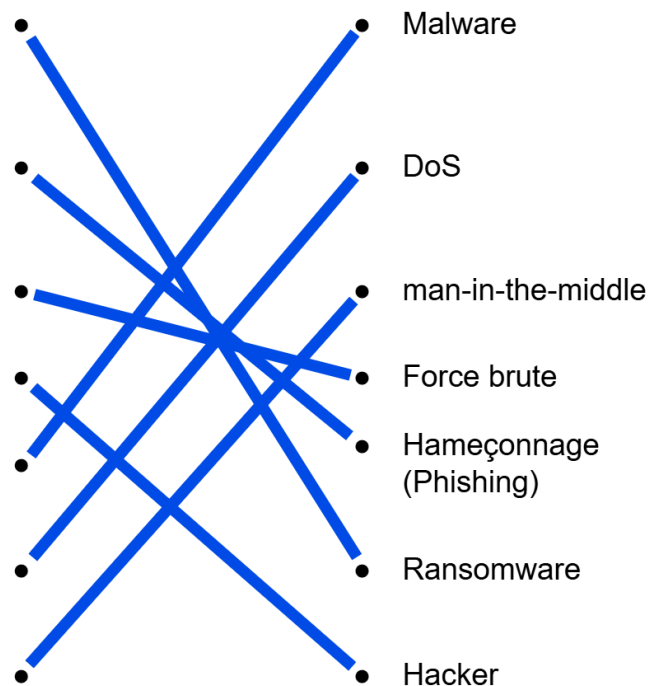
Méthode de décryptage en testant toutes les possibilités une par une

Pirate informatique

Terme générique désignant une application malveillante

attaque informatique ayant pour but de rendre indisponible un service

Interception de communications entre deux parties, sans que celles-ci s'en rendent compte



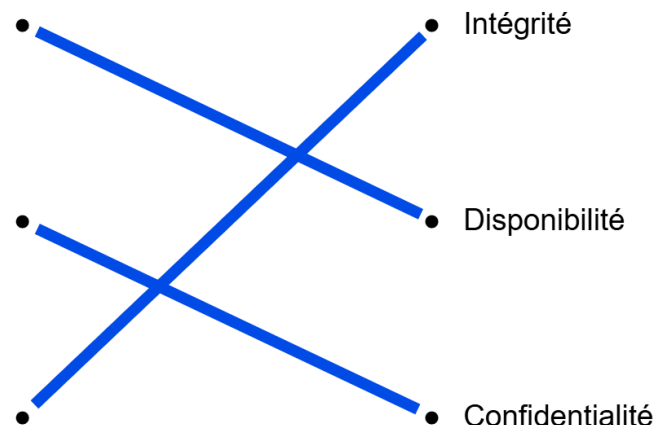
Ex 19.

/ 3 pts

Lorsque vous tapez une adresse URL dans la zone prévue dans votre navigateur web, la page désirée ne se charge pas et vous voyez un message d'erreur "404 Not found".

Tou-te-s les camarades de votre classe ont lu le dernier message que vous avez envoyé à votre petit-e ami-e alors que vous avez bien vérifié dans la boîte d'envoi qu'il n'y avait qu'un-e seul-e destinataire.

Le dernier message que vous avez reçu de votre enseignant d'informatique était presque illisible. Pourtant vous savez qu'il met toujours beaucoup de soin à se relire avant de les envoyer



FIN