

ÉPREUVE REGROUPEE

Discipline: INFORMATIQUE

Date: 04/06/2025

Cours: 1 IN DP 06 / 07 / 08

Durée: 105 minutes

Maîtres: JC Barros

Matériel autorisé:

- Aucun (aide-mémoire en fin d'épreuve)

Consignes de travail:

- Ecriture lisible et propre demandée
- Mettre en évidence les résultats
- 10 pages dont 2 d'aide-mémoire

Nom et prénom:

Total: 76 points

Programmation (34 pts)

Ex 1.

/ 4 pts

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

Exemple d'utilisation:

Instruction	Résultat dans la console
<code>print(carre(3))</code>	9

code de la fonction:

- b. Qu'affiche le code - utilisant la fonction `carre` - ci-dessous ? (1 pt)

`print(carre(carre(2)))`

Ex 2.

/ 4 pts

On veut écrire un programme qui demande "nb_final" à l'utilisateur puis crée une liste avec tous les nombres entiers de 1 jusqu'à "nb_final". Ensuite, le programme calcule et affiche la somme de tous ces nombres.

Voici un programme qu'il faut corriger pour qu'il fonctionne correctement:

```
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)
```

Indiquer (sous l'exemple) 4 autres modifications pour corriger ce programme

Ligne	Modification(s)
2	nb final → nb_final

Ex 3.

/ 10 pts

Lire le programme Python présent sur la page suivante puis répondre aux questions:

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

f1 →

f2 →

f3 →

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

entourer le choix: **f1**, **f2** ou **f3** →

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

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

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))
```

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])
```

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
```

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)
Exemple: <code>nombre = 1</code> <code>print(nombre)</code> <code>nombre = 2</code> <code>nombre = 3</code>	1 (2 pts)	nombre : 1 2 3 (2 pts)
<code>nombre = 0</code> <code>somme = 0</code> <code>while nombre < 4:</code> <code> nombre += 1</code> <code> print(nombre)</code> <code> somme += nombre</code> <code>print(somme)</code>		nombre : somme :
<code>nombre = 0</code> <code>somme = 0</code> <code>while nombre < 4:</code> <code> print(nombre)</code> <code> nombre += 1</code> <code> somme += nombre</code> <code>print(somme)</code>		nombre : somme :
<code>somme = 0</code> <code>for nombre in range(4):</code> <code> somme += nombre</code> <code> print(nombre)</code> <code>print(somme)</code>		nombre : somme :

Réseaux (32 pts)

Ex 7.

/ 1 pt

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

Ex 8.

/ 2 pts

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

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

Ex 9.

/ 4 pts

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

a.

b.

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

Ex 10.

/ 2 pts

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

Ex 11.

/ 4 pts

Dans le protocole IPv4:

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

b. Combien d'adresses IP différentes peut-on utiliser dans le système IPv4 ? (1 pt)
(afficher le calcul sans avoir à calculer le résultat)

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

Si oui, expliquer pourquoi / Sinon, expliquer comment on peut contourner ce problème.

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

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)

Ex 15.

/ 1 pt

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

- 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

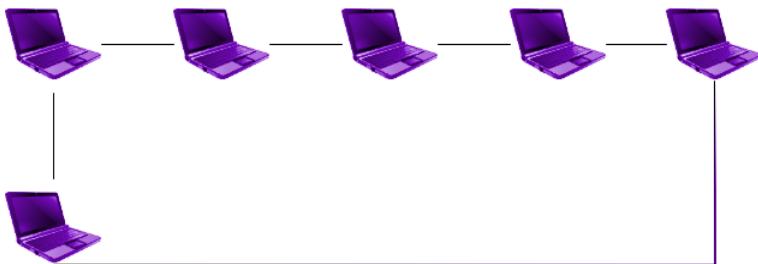
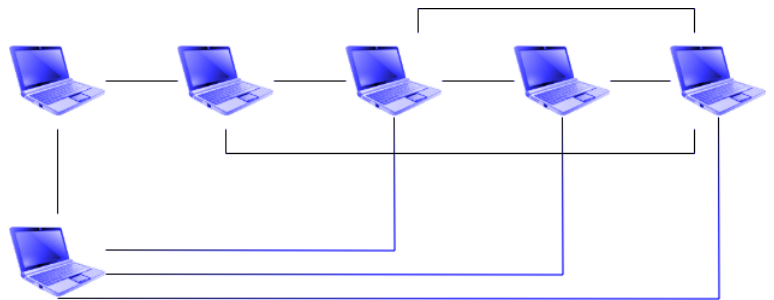
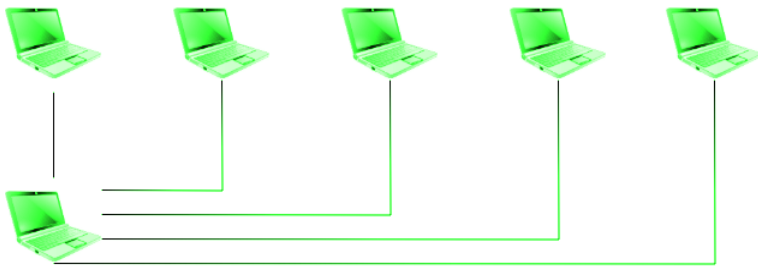
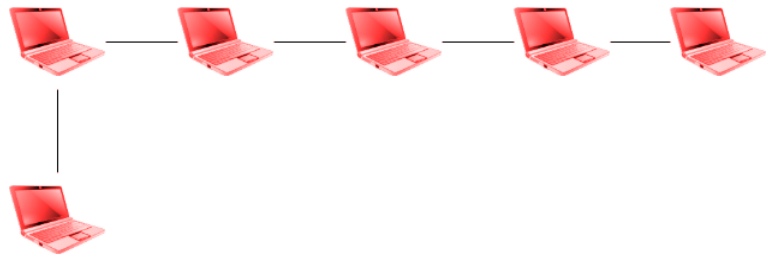
Ecrire sur les pointillés "encapsuler" ou "décapsuler" pour chaque phrase:

- Un navigateur qui demande une page web à un serveur doit 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 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 la trame pour connaître l'adresse du destinataire de ce message.
- Un routeur qui renvoie un paquet (message) doit 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)



b. Quelle topologie est la plus adaptée aux réseaux locaux? (1 pt)

c. Quelle topologie est mise en place dans l'internet? (1 pt)

d. Donner un avantage et un inconvénient de la topologie en anneau: (2 pts)

Sécurité (10 pts)

Ex 18.

/ 7 pts

Relier les notions correspondantes

Application malveillante opérant un chantage en échange de la récupération de données ou d'un service	•	• Malware
Récupération de données personnelles (mot de passe ou autre) par un pirate se faisant passer pour une société connue	•	• DoS
Méthode de décryptage en testant toutes les possibilités une par une	•	• man-in-the-middle
Pirate informatique	•	• Force brute
Terme générique désignant une application malveillante	•	• Hameçonnage (Phishing)
attaque informatique ayant pour but de rendre indisponible un service	•	• Ransomware
Interception de communications entre deux parties, sans que celles-ci s'en rendent compte	•	• Hacker

Ex 19.

/ 3 pts

Relier les enjeux de sécurité et les exemples de situations correspondants

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".	•	• Intégrité
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.	•	• Disponibilité
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	•	• Confidentialité

FIN DE L'ÉPREUVE (l'aide-mémoire suit)

Programmation (Python)

Remarque: Certaines instructions n'ont peut-être pas encore été vues en classe

Calculs :

opérations de base: + - * / **

reste et quotient de division euclidienne: % //

var % div vaut 0 si var est divisible par div

Variables et affectations :

types: int, float, str, bool, list ...

int(), float() pour **convertir** en int ou en float

affectation simple : ma_variable = -5

permutation de variables: a, b = b, a

affectations combinées: +=, -=, *=, etc.

ex: x += 1 est comme x = x + 1

Listes :

liste: L = [-1, 3.2, "abc"]

liste vide: []

multiplication: [0]*3 -> [0, 0, 0]

élément d'une liste: L[1] = 5

doit déjà exister, l'indice commence à 0.

ajouter un élément à une liste :

L.append(-2.5)

supprimer élément à la position i :

del L[i] ou L.pop(i)

supprimer élément de valeur v dans L :

L.remove(v)

longueur d'une liste L :

len(L)

somme des valeurs d'une liste:

sum(liste)

slice(tranche): L[i:j] (i inclus ! j exclu)

élément au hasard d'une séquence non vide:

choice(L)

Chaînes de caractères :

chaîne: txt="bonjour" ou txt='bonjour'

concaténation: txt + " est un texte"

élément, slice d'une chaîne: texte[i]

comme listes mais non modifiable

Entrées/sorties :

sortie: print(valeur,... , end='\n')

valeur de type quelconque, end optionnel

entrée: texte = input("message")

Attention: retourne toujours un str

Commentaires :

sur une ligne : # texte

plusieurs lignes : '''texte''' ou """texte"""

Structures de contrôle :

1^{ère} ligne finit par ':' puis indentation des lignes du bloc

tests simples: ==, !=, <, <=, >, >=, in

tests combinés: and, or, not, a < b < c

conditionnelle

if condition1:

instructions

elif condition2:

elif =contraction de else if

instructions

else:

instructions

boucle while (boucle non bornée) :

while condition:

instructions

boucle for (boucle bornée) :

for i in liste: # boucle parcourt

la liste

instructions

for i in range(5):#pour i de 0 à 4

instructions

range(i,j,k): de i (inclus) à j (exclu) par pas de k

sauter, quitter une boucle : continue, break

Fonctions:

définition de la fonction f à deux paramètres:

def f(x, y):

reponse = x**2 + y**2

return reponse

appel de la fonction: f()

variable = f(3, 5)

print(f(2, 6))

etc.

Modèle OSI

