

ÉPREUVE REGROUPEE - Correction

Discipline: INFORMATIQUE

Cours: 1 IN DP (tous les cours)

Date: 06 / 06 / 2023

Durée: 105 minutes

Maître: Barros / Chiou / Garcia / Schmidt

Matériel autorisé:

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

Consignes de travail:

- Ecriture lisible et propre demandée
- Mettre en évidence les résultats

Nom et prénom:

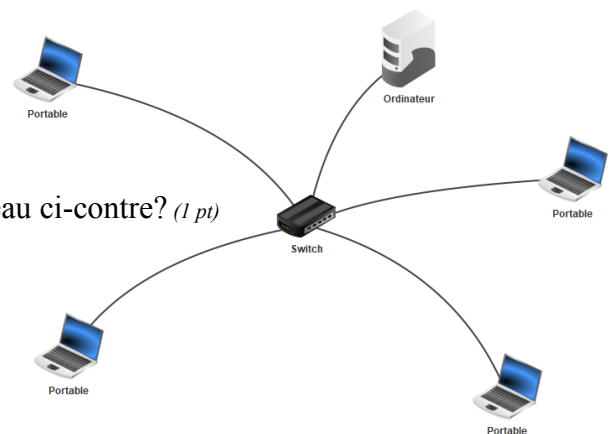
Total: 65 points

RÉSEAUX:

Problème 1. (5 pts)

- a. Comment s'appelle la topologie du réseau ci-contre? (1 pt)

la topologie en étoile



- b. Pourquoi n'est-elle pas utilisée dans les WAN comme par exemple l'internet? (2 pt)
Donner deux arguments précis.

1. ce réseau est entièrement dépendant du noeud central s'il casse, tout est HS
2. il y aurait une surcharge du noeud central
3. Le noeud central verrait tout passer et pourrait espionner tous les échanges etc.

- c. Indiquer pour chaque réseau ci-dessous si c'est un LAN ou un WAN: (2 pt)

Réseau à la maison LAN.....

Ensemble de routeurs reliés
par des fibres optiques transatlantiques WAN.....

Réseau d'une salle de classe LAN.....

Internet WAN.....

Problème 2. (8 pts)

a. Relier les mots de gauche avec ce qui leur correspond à droite. (2 pts)

- | | | |
|----------------|---|---|
| internet (I) | • | est une infrastructure matérielle (I) |
| | | est un réseau de contenus (W) |
| web (W) | • | ses noeuds sont des appareils (I) |
| | | ses noeuds sont des pages (W) |

b. Quel nom (sigle de 3 lettres) donne-t-on à l'adresse qu'on saisit dans un navigateur web (Chrome, Firefox, etc.) ? (1 pts)

URL

c. Quel système (sigle de 3 lettres) permet de trouver l'adresse IP de **eduge.ch** ? (1 pts)

DNS

d. Expliquer ce qui se passe **au niveau réseau** depuis que j'entre l'adresse d'un site (comme par exemple eduge.ch) dans un navigateur jusqu'à ce que la page du site s'affiche. (donner quatre étapes importantes). (4 pts)

1. le protocole http ou https prend en charge l'opération (connection et échanges)
2. l'adresse IP est donnée par DNS
3. la requête parvient au serveur
4. le serveur accepte la requête et envoie la page du site mais aussi...

Les données qui quittent le terminal (1) sont transférées à travers le wifi ou un câble Ethernet (2), en passant par des noeuds tels que switch, hub ou routeurs (3) pour arriver au serveur qui envoie les données de la page web par le chemin inverse (4).
etc.

Problème 3. (3 pts)

Citer 3 protocoles utilisés dans les réseaux et indiquer à quelle couche ils correspondent. Attention, chaque protocole doit être associé à une couche distincte du modèle OSI!

protocoles	couches
http, https, ftp, pop, smtp, etc.	7. Application
TCP, UDP	4 Transport
IP / Wifi, Ethernet	3. Réseau / 2. Liaison

Problème 4. (3 pts)

Relier chaque adresse (à gauche) avec sa description correcte (à droite): (3 pts)

192.168.1.45 (IP)	☐	☐	adresse MAC
CA:22:08:D1:E2:11 (MAC)	☐	☐	adresse URL
fr.wikipedia.org (URL)	☐	☐	adresse IP

Problème 5. (4 pts)

Choisir **UNE** des deux questions (entourer **A ou B**) et en donner 4 aspects importants ci-dessous:

CHOIX A : Encapsulation

CHOIX B: Topologie maillée (mesh)

Réponse:

A : Avant de quitter l'appareil, les données passent à travers 7 couches (1), au travers desquelles des informations s'ajoutent à chaque étape (2) à cause des différents protocoles (3). Arrivées au destinataire, les données parcourent les couches dans le sens inverse (4) (décapsulation)

B : C'est une topologie où tous les nœuds sont connectés à ses voisins (1), de ce fait, elle est très résistante aux coupures (2). Cette topologie est utilisée par l'Internet (3), mais le matériel pour relier les nœuds rend cette topologie plus chère.

SÉCURITÉ:

Problème 6. (3 pts)

Indiquer pour chaque problème ci-dessous quel est le principe de sécurité informatique concerné parmi les trois que nous avons vus:

Exemple: “J’essaie de me connecter à ma boîte eduge; la page de connexion ne fonctionne pas.”

Principe concerné: disponibilité

1. J’envoie le message suivant: “Virer 10 CHF à Eve” à ma banque mais elle reçoit le message suivant: “Virer 90 CHF à Eve”.

Principe concerné: intégrité

2. Les ingénieurs d’un système de messagerie ont accès aux messages que j’envoie à mes amis.

Principe concerné: confidentialité

3. J’essaie d’envoyer un message avec Whatsapp mais l’application me dit que le système est en maintenance et qu’il faut réessayer plus tard.

Principe concerné: disponibilité

Problème 7. (4 pts)

Citer deux types de cyberattaques et donner au moins une information importante pour chaque.

Problème 8. (3 pts)

Donner trois précautions différentes (une par notion) qui peuvent aider à assurer les trois notions de sécurité suivantes :

- disponibilité: sauvegarde et/ou mise à jour fréquentes
- intégrité: antivirus et/ou faire des mises à jour. (Les sauvegardes sont insuffisantes [-0,5])
- confidentialité: chiffrement des données, mot de passe robuste etc. (Les sauvegardes sont insuffisantes [-0,5])

PROGRAMMATION (Python):

Problème 9. (5 pts)

Compléter le code Python ci-dessous pour que les deux fonctions (moyenne et affiche) fonctionnent correctement. Indication: il y a 5 zones (.....) où on attend quelque chose.

```
# définitions des fonctions:
def moyenne():
    print("Quelle note à l'évaluation 1?")
    premiere = int(input())
    print("Quelle note à l'évaluation 2?")
    deuxieme = int(input())
    print("Quelle note à l'évaluation 3?")
    troisieme = int(input())
    moyenne = (premiere + deuxieme + troisieme) / 3

    return moyenne .....

def affiche( ...moy....., ...nom.....):
    print("L'élève:", nom)
    print("obtient la moyenne:", moy)

# programme principal:
mLuc = moyenne() # mLuc stocke la moyenne de Luc
mEve = moyenne() # mEve stocke la moyenne d'Eve

affiche(mLuc, ..."Luc"... )

affiche(..mEve....., "Eve")
```

Problème 10. (1 pts)

Qu'affiche le programme suivant ?

```
texte = 'abcdef'
print(texte[3:5])
```

Réponse: de

Problème 11. (2 pts)

Quel est l'ordre logique de ces 4 lignes pour former un programme cohérent?

```
# partie A:
derniere_lettre = mon_nom[longueur_nom-1]

# partie B:
print("Le dernière lettre de votre nom est", derniere_lettre)

# partie C:
mon_nom = input('Entrer un nom :' )

# partie D:
longueur_nom = len(mon_nom)
```

Réponse : C - D - A - B

Problème 12. (9 pts)

Lire les quatre indications ci-dessous puis écrire les instructions en Python correspondant à chacune.

L'ensemble du code doit former un programme Python cohérent :

```
# Indication 1: (2 pts)
# Demander le prénom de l'utilisateur et le stocker
# dans une variable.

autre_prenom = input("Donnez votre prénom : ")

# Indication 2: (1 pts)
# affecter votre propre prénom à la variable "mon_prenom"

mon_prenom = 'Gwendoline'

# Indication 3: (2 pts)
# Afficher un message qui présente
# le prénom de l'utilisateur et le vôtre.
# (utiliser deux variables) Par exemple:
# "Tu t'appelles Robert, je m'appelle Gwendoline."

print("Tu t'appelles", autre_prenom, ", je m'appelle",
mon_prenom, ".")

# Indication 4: (4 pts)
# En utilisant les deux variables, afficher selon ce qui convient:
# "Nos noms sont identiques!" OU "Nos noms sont différents."

if autre_prenom == mon_prenom :
    print("Nos noms sont identiques!")
else:
    print("Nos noms sont différents.")
```

Problème 13. (6 pts)

Pour chacun des programmes suivants (à gauche), indiquez ce qui est affiché à l'écran (rien de plus, rien de moins!) et la valeur de la variable `i` à la fin de l'exécution du programme.

PROGRAMMES	Affichage à l'écran	Valeur finale de <code>i</code>
EXEMPLE <pre>i = 10 while i <= 12: print(i) i = i + 1</pre> FIN DE L'EXEMPLE	10 11 12	13
1. programme 1 (2 pts) <pre>i = 0 while i < 3: print(i) i = i + 1</pre>	0 1 2	3
2. programme 2 (2 pts) <pre>i = 0 while i < 3: print(i) i = i + 1</pre>	0 0 0 etc.	0
3. programme 3 (2 pts) <pre>i = 0 while i < 3: i = i + 1 print(i)</pre>	1 2 3	3

Problème 14. (3 pts)

Écrire un programme utilisant une boucle `for` pour afficher tous les nombres de 27 à 64 compris.

for `i` in range(27 , 65) :

`print(i)`

Problème 15. Choisir **UNE** des deux questions (entourer **A ou B**) (6 pts)

CHOIX A : Fonction verlan

Écrire une fonction “verlan” en Python.

Cette fonction reçoit un texte et retourne le texte à l’envers.

Par exemple,

```
print(verlan(“Bonjour!"))
```

doit afficher “!ruojnoB”.

CHOIX B : Fonction factorielle

Écrire une fonction “factorielle” en Python.

Cette fonction reçoit un nombre entier et retourne la factorielle de ce nombre.

Attention! On n’a pas le droit d’utiliser l’opérateur factorielle (!). Il faut faire le calcul autrement.

Rappel: $5! = 5 * 4 * 3 * 2 * 1$.

Par exemple,

```
print(factorielle(5))
```

doit afficher 120.

Fonction Python (proposition de corrigé):

```
# A :
def verlan(texte):
    envers=' '
    for car in texte:
        envers=car+envers
    return envers

# B :
def factorielle(n):
    resultat=1
    for i in range(1,n+1):
        resultat=resultat*i
    return resultat

# **(facultatif)*****
#programme principal
A_inverser=input('texte à inverser?')
print("A l'envers, ça donne", verlan(A_inverser))
Nombre=int(input('nombre à "factoriser"?'))
print('La factorielle de ce nombre est', factorielle(Nombre))
```

FIN DE L'ÉPREUVE
CI-DESSOUS: AIDE-MÉMOIRE

Programmation (Python)

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

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

