

ÉPREUVE REGROUPEE

Discipline: INFORMATIQUE

Date: 05.06.2024

Cours: 1 IN DP 03 / 04 / 05

Durée: 105 minutes

Maître: H. Garcia

Matériel autorisé:

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

Nom et prénom:

Total: 68 points

RÉSEAU:

1.

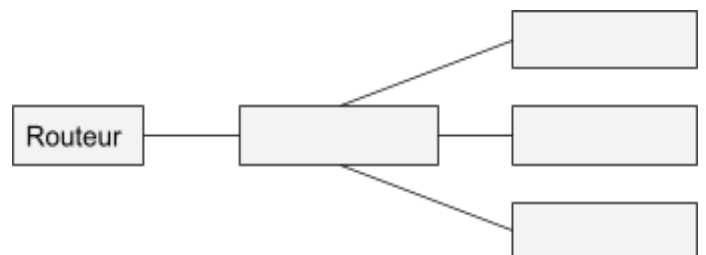
/ 3 pts

Le schéma ci-contre représente un réseau local.

a. Quelle est sa topologie (1 pt) ?

Etoile

b. Expliquer à quoi sert un routeur (1 pt)?



Un routeur est un équipement matériel informatique dont la fonction principale consiste à orienter les données à travers un réseau. Il permet de faire circuler des données entre deux interfaces réseau.

c. Écrire dans le rectangle concerné le nom de l'unique appareil (hormis le routeur) dont on est sûr de l'identité (1 pt).

Switch

2.

/ 5 pts

Entourer chaque bonne réponse

a. Pour envoyer un e-mail à Mamie, il y a seulement deux appareils qui sont sollicités sur le réseau : mon ordinateur et le sien.

Vrai / Faux

b. Pour envoyer un e-mail à Mamie, il n'y a qu'un seul programme sollicité sur le réseau.

Vrai / Faux

c. Pour envoyer un e-mail à Mamie, les données peuvent circuler sous la mer ou dans l'air.

Vrai / Faux

d. Pour envoyer un e-mail à Mamie, on peut écrire son adresse mail dans l'URL.

Vrai / Faux

Pour envoyer un e-mail à Mamie, un seul chemin est utilisé pour acheminer ma lettre sur le réseau comme un envoi postal.

Vrai / Faux

3.

/ 2 pts

Citer deux protocoles utilisés dans la couche **application** du modèle OSI

- a. HTTP, HTTPS, FTP, etc
- b.

4.

/ 3 pts

Expliquer la fonction de DNS en quelques lignes (donner 3 informations pertinentes).

Le Domain Name System (DNS), ou système de nom de domaine.

- 1. Le DNS est essentiel pour le fonctionnement d'Internet.
- 2. Répertoire téléphonique d'Internet : Le DNS agit comme un répertoire téléphonique en traduisant les noms de domaine (comme "nytimes.com" ou "espn.com") en adresses IP spécifiques.
- 3. Conversion des noms de domaine en adresses IP : Lorsqu'un utilisateur saisit un nom de domaine dans son navigateur (par exemple, "example.com"), le DNS traduit ce nom en une adresse IP utilisable par la machine (par exemple, "192.168.1.1"). Cette conversion permet de localiser la page web correspondante. Le processus de résolution DNS se fait en arrière-plan, sans interaction de l'utilisateur

En somme, le DNS est un élément clé pour la connectivité et la navigation sur Internet, permettant aux utilisateurs d'accéder aux sites web sans avoir à mémoriser les adresses IP.

5.

/ 1 pt

Quel est le nom du langage principal utilisé pour écrire des pages web ?
voir l'exemple ci-dessous

HTML

```
<!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head>
<title>
  Exemple de HTML
</title>
</head>
<body>
  Ceci est une phrase avec un <a href="cible.html">hyperlien</a>.
<p>
  Ceci est un paragraphe où il n'y a pas d'hyperlien.
</p>
</body>
</html>
```

- a. Citer deux différences entre les protocoles TCP et UDP. (2 pts)

TCP (protocole de contrôle de transmissions) et UDP (protocole de datagramme utilisateur) :

1. Connexion vs. Sans connexion :

- TCP est un protocole basé sur la **connexion**. Lorsqu'un appareil établit une communication via TCP, il effectue un processus de **poignée de mains** (handshaking) avec l'appareil destinataire pour établir une connexion. Cette approche garantit la fiabilité des données, mais elle est plus lente.
- UDP, en revanche, est un protocole **sans connexion**. Il ne nécessite pas d'établir une connexion préalable. Les données sont simplement envoyées sans vérification de la réception. Cela le rend plus rapide, mais moins fiable que TCP.

2. Fiabilité vs. Vitesse :

- TCP est plus **fiable**. Il s'assure que les données sont correctement reçues en utilisant des mécanismes de retransmission et de contrôle d'erreur. Si un paquet est perdu, TCP le renvoie jusqu'à ce qu'il soit reçu avec succès.
- UDP, en revanche, privilégie la **vitesse**. Il ne gère pas la retransmission des paquets perdus, ce qui le rend plus rapide. Cependant, cela signifie qu'il peut y avoir des pertes de données si un paquet est perdu en cours de route.

En résumé, choisissez TCP lorsque la fiabilité est primordiale (par exemple, pour les transferts de fichiers) et UDP lorsque la vitesse est essentielle (par exemple, pour le streaming vidéo ou les jeux en ligne)

- b. Relier les exemples d'applications utilisant chacun de ces protocoles. (1 pt)

streaming (Youtube, Spotify, Netflix, etc.) 1 • • 1 UDP

web, email, transfert de fichiers 2 • • 2 TCP

- c. Dans quelle couche du modèle OSI ces protocoles sont-ils utilisés? (1 pt)

Couche de transport

Quelle est la différence entre **internet** et **web**?

Donner au moins deux notions importantes

1. Internet :

- **Internet** est un système de réseaux informatiques reliés entre eux. Le nom "Internet" provient de la contraction d'«interconnected network», qui signifie réseau interconnecté en anglais.
- Il permet d'envoyer des **paquets d'informations** d'un ordinateur à un autre grâce à un système complexe d'adresses. À l'origine, ARPANET, l'ancêtre d'Internet, a été inventé par le département américain de la Défense dans

les années 1960. Il a d'abord été utilisé à des fins militaires, puis pour faciliter la transmission d'informations entre des universités et des centres de recherche.

- Internet prend en charge différentes applications telles que le mail ou le pair à pair (utilisé pour le partage de fichiers).

2. Le World Wide Web (ou Web) :

- Le Web est né en 1989 lorsque Tim Berners-Lee, physicien au CERN (Conseil européen pour la recherche nucléaire), a écrit le premier document décrivant ce qui deviendra le World Wide Web.
- Le Web permet de consulter des pages sur des sites à l'aide de navigateurs adaptés, via des machines telles que des ordinateurs ou des smartphones.
- Il s'agit d'une immense bibliothèque de documents reposant sur les fondations complexes d'Internet. Le Web utilise de nombreuses technologies et protocoles pour être simple d'utilisation, comme l'hypertexte (qui permet de naviguer d'un site à un autre en cliquant sur un lien) et les URL (qui identifient simplement l'adresse d'un site ou d'un contenu).

En résumé, Internet est la structure sous-jacente qui relie les ordinateurs, tandis que le Web est l'ensemble des pages et des sites que nous consultons via Internet.

8.

/ 4 pts

Relier chacune des adresse suivantes avec son type:

https://fr.wikipedia.org/wiki/Informatique	1	•	•	1 URL
A7:BF:21:2C:FE:14	2	•	•	4 Adresse (IP)
10.23.567.18	3	•	•	3 aucune...
192.168.1.202	4	•	•	2 Adresse (MAC)

9.

/ 4 pts

Dans le protocole IPv4:

- Combien d'octets composent une adresse (2 pt)?
4
- Combien d'adresses IP différentes peut-on avoir dans le système IPv4 ? (afficher le calcul sans avoir à calculer le résultat) (2 pt)
 $2^8 * 2^8 * 2^8 * 2^8 = 256 * 256 * 256 * 256 = 4294967296$

10.

/ 3 pts

Relier les notions correspondantes

Intégrité 1	•	•	2 Maintenir le bon fonctionnement du système d'information
Disponibilité 2	•	•	3 Rendre les données inutilisables pour les personnes qui ne sont pas les destinataires de ces données
Confidentialité 3	•	•	1 Garantir que les données ne sont pas modifiées pendant le transport

11.

/ 7 pts

Vocabulaire lié à la sécurité : relier les notions correspondantes

Attaque impliquant un chantage en échange de la récupération de données ou d'un service.	•	•	5 Malware
Récupération de données personnelles (mot de passe ou autre) par un service se faisant passer pour un autre (votre banque, la poste, un magasin en ligne, etc.)	•	•	6 Spyware
Méthode pour décrypter un message ou trouver un mot de passe en testant toutes les possibilités une par une.	•	•	7 Man in the middle
Pirate informatique	•	•	3 Force brute
Terme générique désignant une application malveillante	•	•	2 Phishing
Application destinée à intercepter des données confidentielles	•	•	1 Ransomware
Interception des communications entre deux parties, sans que celles-ci s'en rendent compte	•	•	4 Hacker

12.

/ 8 pts

Pour chacun des programmes suivants, indiquer tout ce qu'il affiche à l'écran (rien de plus, rien de moins) et la valeur de la variable **k** à la fin de l'exécution du programme.

PROGRAMMES	Affichage à l'écran	Valeur finale de k
<pre># EXEMPLE k = 10 print(k) k = k + 1 print(k)</pre>	<pre>10 11</pre>	11
<pre># Programme 1 for k in range(3): print(k * 2) print('Bye!')</pre>	<pre>0 2 4 Bye!</pre>	2
<pre># Programme 2 for k in ['Y', 'e', 's']: print(k * 2)</pre>	<pre>YY ee ss</pre>	s
<pre># Programme 3 k = 3 while k > 0: print(k) k = k - 1</pre>	<pre>3 2 1</pre>	0
<pre># Programme 4 k = 3 while k > 0: k = k - 1 print(k)</pre>	<pre>2 1 0</pre>	0
<pre># Programme 5 k = 10 if k * 3 <= 0: print(1) elif k + 2 >= 0: print(2) elif k ** 2 > 0: print(3) else: print(4) print('OK')</pre>	<pre>2</pre>	10

13.

/ 2 pts

m = 'Carlos !'

- Quel est le type de la variable **m** ci-dessus ? (en vocabulaire Python) **str (string)**
- Quel est l'index (position dans la chaîne **str**) du caractère **r** ? **2**

14.

/ 5 pts

Indiquer au bout de chaque flèche ce que les lignes concernées affichent à l'écran.

```

1 S = [ 'a', 'b' , 'b' , 'a' ]
2
3 print( S[2]+S[3] )
4
5 print(len(S))
6
7 S.append( 'm' )
8
9 print(S)
10
11 print(S[1:3])
12
13 S.remove( 'a' )
14 print(S)
15

```

ba
4
['a', 'b', 'b', 'a', 'm']
['b', 'b']
['b', 'b', 'a', 'm']

15.

/ 2 pts

Voici un programme qui ne fait pas ce qui est prévu (commentaire à la ligne 1).

Corriger le programme pour qu'il affiche bien ce qui est prévu..

```

#Afficher 21,23,25,27
n = 20
for i in range (1, 5):
    print(n + i)

```

```

Console ×
Python 3.7.9 (bundled)
>>> %Run test.py
21
22
23
24

```

```

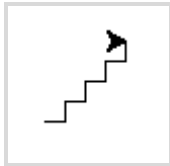
n = 20
for i in range (1, 8/9, 2):
    print(n + i)

```

- a. Tracer le dessin correspondant au programme ci-contre. (2 pts)

La tortue est dirigée vers la droite au départ.

Point de départ: X



```

1  from turtle import *
2
3  forward(10)
4  left(90)
5  forward(10)
6  right(90)
7
8  forward(10)
9  left(90)
10 forward(10)
11 right(90)
12
13 forward(10)
14 left(90)
15 forward(10)
16 right(90)
17
18 forward(10)
19 left(90)
20 forward(10)
21 right(90)

```

- a. Écrire ci-dessous un programme qui donne le même dessin en moins de 10 lignes. (3 pts)

```

from turtle import *
for i in range(4):
    forward(10)
    left(90)
    forward(10)
    right(90)

```

Voici un programme Python:

```
def mini (liste):
    candidat = liste[0]
    for element in liste:
        if element < candidat:
            candidat = element
    return candidat

def trier (liste):
    resultat = []
    for i in range (len(liste)):
        m = mini(liste)
        resultat.append(m)
        liste.remove(m)
    return resultat

liste = [2, 5, 1, 9, 6, 0]
desordonnes
nouv_liste = trier(liste)
print(nouv_liste)
```

mini est une

resultat est une vide
ici, la boucle se répète fois
m correspond au minimum dans la liste
on ajoute a
on m de liste

liste contient des nombres

affiche la liste

a) A quoi servent les symboles # ? (0,5 pt)

commentaires

b) Remplir (dans le programme) les zones pointillées avec le mot, la liste ou le nombre qui manque.
7 réponses sont attendues. (3,5 pts)

fonction

liste vide

len(liste) fois

m a la liste resultat

efface (enleve)

nouv_liste

c) Que fait ce programme Python ? (1 pt)

Le programme trie une liste de nombres par ordre croissant.

d) Proposer des modifications à ce programme pour qu'il affiche la liste dans l'ordre **décroissant**.
Indiquer clairement les modifications, par exemple en mentionnant avec chaque modification la ligne du programme concernée. (3 pts)

```
if element > candidat:
```

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.

