

ÉPREUVE REGROUPEE

Discipline: INFORMATIQUE

Cours: 1 IN DF 01 / 02 / 03

Date: 15 / 06 / 2022

Durée: 105 minutes

Maître: JC Barros

Matériel autorisé:

- 1 feuille de notes recto / verso manuscrite
A RENDRE AVEC L'ÉPREUVE

Consignes de travail:

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

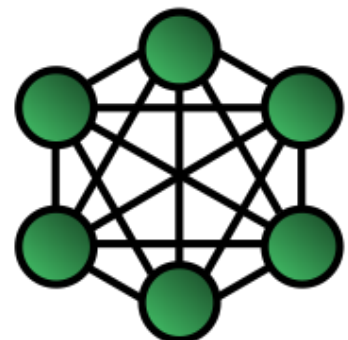
Nom et prénom:

Total: 85 points

RÉSEAU:

Problème 1: (3 pts)

1. Comment s'appelle la topologie du réseau ci-contre? (1 pt)
2. Citer un avantage de cette topologie. (Réponse courte) (1 pt)
3. Citer un inconvénient de cette topologie. (Réponse courte) (1 pt)



Problème 2: (10 pts)

1. Quelle est la différence entre Internet et Web? (Réponse courte) (2 pts)

2. Quelle est la différence entre adresse URL et adresse IP? (Réponse courte) (2 pts)

3. Quelle est la différence entre LAN et WAN? (Réponse courte) (2 pts)

4. Expliquer de manière détaillée l'un des 6 concepts soulignés apparaissant dans les questions ci-dessus (au moins quatre aspects intéressants sont attendus dans l'explication). (4 pts)

SÉCURITÉ:

Problème 3: (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 un email à mon professeur d’informatique mais le contenu est altéré: il manque la pièce jointe.
principe concerné:
2. J’envoie un email à mon professeur d’informatique mais toute la classe reçoit ce message alors qu’il ne leur était pas destiné.
principe concerné:
3. J’essaie de me connecter à Classroom (ou Moodle) mais ça ne fonctionne pas.
principe concerné:

Problème 4: (4 pts)

Citer une fameuse cyberattaque et donner des informations importantes à son sujet (au moins 3).

Problème 5: (6 pts)

Donner 3 techniques pour se protéger contre les cyberattaques et, pour chacune, expliquer brièvement en quoi elle nous protège.

STRUCTURE DE DONNÉES:

Problème 6: (8 pts)

Évaluer les valeurs contenues dans les variables x, y et z à l'issue du programme suivant et indiquer leur type.

Programme :

```
w = 2 ** 3
x = 'z' * 3
y = ['a', 'b'] + ['c', 'd']
z = 1.5 + 0.3
```

Nom de la variable	Valeur affichée si on met cette variable dans un print()	Type de la variable
w		
x		
y		
z		

Problème 7: (8 pts)

Souligner une erreur dans chaque ligne qui empêche l'exécution du programme suivant. Ensuite, écrire la version corrigée juste en dessous.

Remarque: Les quatre lignes forment le programme entier.

ligne 1: (2 pts)

```
mon nom = input("Entrer un nom : ")
```

ligne 2: (2 pts)

```
nombre = Input(" Entrer un nombre : ")
```

ligne 3: (2 pts)

```
resultat = nombre + 2
```

ligne 4: (2 pts)

```
print("Le résultat de ", Nombre, " + 2 est ", resultat)
```

PYTHON:

Problème 8: (10 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 à l'utilisateur de donner la taille du rayon d'un disque
```

```
# Indication 2: (2 pts)
# utiliser le rayon demandé au point précédent pour calculer l'aire
# du disque avec la formule aire = 3,14 • rayon2
```

```
# Indication 3: (2 pts)
# Afficher le message suivant avec les nombres correspondants :
# l'aire du disque de rayon ... est de ...
```

```
# Indication 4: (4 pts)
# Afficher un des mots selon les critères suivants :
# "Petit" si l'aire est inférieure à 10000
# "Moyen" si l'aire est entre 10000 et 20000
# "Grand" si l'aire est supérieure à 20000
```

Problème 9: (7 pts)

1. Qu'affiche le programme suivant ? (2 pts)

```
1 i = 10
2 while i <= 12:
3     print(i)
4     i = i + 1
```

Réponse:

2. Qu'affiche le programme suivant ? (2 pts)

```
1 i = 10
2 while i <= 12:
3     i = i + 1
4 print(i)
```

Réponse:

3. Qu'affiche le programme suivant ? (3 pts)

```
1 pays = ['France', 'Espagne', 'Suisse']
2 print(pays[2])
```

Réponse:

Problème 10: (6 pts)

Relier les instructions avec la catégorie qui leur correspond:

Comparaison	☐	☐ print ("Bonjour!")
Affectation	☐	☐ input("Quelle lettre ?")
Conditionnelle	☐	☐ if
Boucle	☐	☐ var = 10
Sortie	☐	☐ for
Entrée	☐	☐ var == 10

QUESTION À CHOIX:

Problème 11: (8 pts)

Choisir **UNE** des quatre questions (**A à D**) parmi les choix suivants

CHOIX A : Décrire le chargement d'une page Web en détail...

Donner une description détaillée des éléments, des étapes qui entrent en jeu lorsqu'on charge la page eduge.ch dans notre navigateur. (minimum 8 informations pertinentes):

CHOIX B: Modèle de couches (OSI ou TCP/IP)

Développer ce sujet (couches) et expliquer à quoi sert ce modèle, en abordant par exemple les aspects suivants (minimum 8 informations pertinentes):

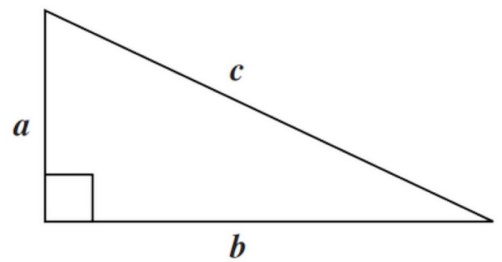
Description du modèle de couches, Noms des couches et leur fonction, Exemples

CHOIX C: Écrire un programme Python:

Écrire un programme Python contenant une fonction **pythagore** qui permet de calculer l'hypoténuse (c) d'un triangle rectangle à partir de ses deux autres côtés (a et b).

La racine carrée peut s'obtenir avec la puissance 0.5

Le programme doit utiliser cette fonction pour afficher l'hypoténuse d'un triangle dont les côtés a et b valent 3 et 4.



$$a^2 + b^2 = c^2$$

CHOIX D: voir page suivante

RÉPONSE (pour les choix **A à C**):

CHOIX D: Comprendre un programme Python (réponses dans l'exercice)

Pour chacune des affirmations au sujet du programme ci-dessous, indiquer si elle est **vraie ou fausse**. Indiquer également le **numéro de la ligne** qui justifie la réponse.

- `guess_the_number(x)` est une fonction
- Lorsque nous exécutons `guess_the_number(10)`, un nombre aléatoire est généré (variable `nombre_aleatoire`) entre 1 et 10.
- L'utilisateur a droit à 10 tentatives pour deviner le numéro `nombre_aleatoire`.
- On sort de la boucle lorsque l'utilisateur trouve le numéro `nombre_aleatoire`.

```
1 import random
2
3 def guess_the_number(x):
4
5     print("+++++")
6     print(" Bienvenu(e) au jeu ")
7     print("+++++")
8     print("Votre objectif est de deviner le nombre aléatoire généré par l'ordinateur.")
9
10    nombre_aleatoire = random.randint(1, x) # Numero aleatoire entre 1 et x.
11
12    prediction = 0 # zéro de sorte qu'il ne coïncide pas avec l'intervalle
13
14    while prediction != nombre_aleatoire:
15        # l'utilisateur saisit un numéro
16        prediction = int(input(f"Devinez un nombre entre 1 et {x}: "))
17
18        if prediction < nombre_aleatoire:
19            print ("Essayez à nouveau. Ce numéro est trop petit.")
20        elif prediction > nombre_aleatoire:
21            print ("Essayez à nouveau. Ce numéro est trop grand.")
22
23        print (f"Félicitations ! Vous avez deviné le numéro {nombre_aleatoire} correctement.")
24
25    guess_the_number(10)
```

Divers :

Problème 12: (4 pts)

Pour assurer l'intégrité de la transmission d'un paquet, on utilise souvent des bits de parité (0 correspond à un nombre pair de bits "1", 1 correspond à un nombre impair de bits "1").

Les 8 octets ci-dessous ont été transmis avec un bit de parité chacun (en gras). On a également transmis un 9^{ème} octet pour assurer une parité verticale.

1. Ce système appelé somme de contrôle (checksum en anglais) permet-il de corriger et pas seulement de détecter une erreur de transmission dans tous les cas? (expliquer) (2 pts)
2. Si oui, corriger celle-ci en entourant le bit à inverser dans le tableau.
Dans tous les cas (si oui ou si non), expliquer pourquoi. (2 pts)

1	0	0	0	0	1	1	1
1	1	0	1	0	0	0	1
1	1	0	0	1	1	1	0
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	1
1	1	1	0	0	1	1	1
1	1	1	0	1	0	1	1
1	1	0	1	1	0	1	1
0	1	0	1	1	0	1	0

Problème 13: (4 pts)

Faire une comparaison entre les protocoles TCP et UDP, en citant au moins un avantage et un inconvénient de chaque protocole.

Problème 14: (4 pts)

Voici les informations trouvées sur une machine : Adresse IPv4 : 12 . 0 . 118 . 218
Masque : 255 . 255 . 0 . 0

1. Quel est le net-id (adresse réseau) ?
2. Quel est le host-id (adresse d'hôte) ?
3. Quelle est l'adresse du broadcast (adresse d'envoi à tous les hôtes du réseau) ?
4. Donner le calcul (pas besoin de donner le résultat) qui permet de déterminer le nombre d'adresses différentes qui peuvent être attribuées aux hôtes (machines) du réseau local :

FIN DE L'ÉPREUVE