

Durée : 1h30

Calculatrice interdite

NOM :

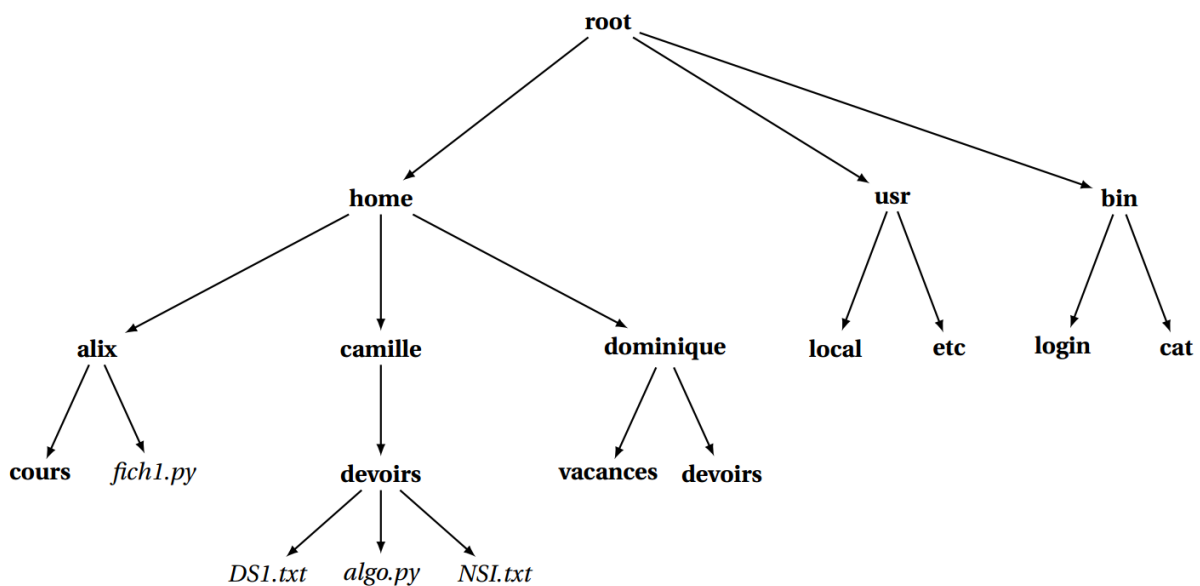
Prénom :

Note : / 40

Commentaire :**Exercice 1. Systèmes d'exploitation**

/ 10 pts

Dans un système d'exploitation de type UNIX, on considère l'arborescence suivante dans laquelle les noms des répertoires sont en gras et ceux des fichiers sont en italique :



On donne ci-après un extrait de quelques commandes de base à utiliser dans un terminal sous linux :

**Extrait des commandes de base linux**

**ls** affiche le contenu d'un répertoire.

**ls repertoire1** affiche le contenu de **repertoire1**

**cd** : cd permet de changer de répertoire courant.

Après la commande **cd repertoire1** , le répertoire courant devient **repertoire1**

**cp** : crée une copie d'un fichier.

**cp fichier1.py fichier2.py** copie fichier1.py sous le nom **fichier2.py**

**mv** : déplace ou renomme un fichier ou un répertoire.

**mv fichier.txt dossier1** déplace **fichier.txt** dans **dossier1**

**rm** : supprime un fichier ou un répertoire.

**rm mon\_fichier.mp3** supprime **mon\_fichier.mp3**

**mkdir** : crée un répertoire.

**mkdir dossier2** crée le répertoire **dossier2**

**cat** : affiche le contenu d'un fichier.

**cat fichier1.txt** affiche le contenu de **fichier1.txt**

**touch** : crée un fichier qui n'existait pas jusqu'alors.

**touch fichier2** crée **fichier2**

Dans l'arborescence décrite dans la figure 1 ci-dessus, alix, camille et dominique sont respectivement les répertoires personnels des utilisateurs Alix, Camille et Dominique.

On souhaite explorer et modifier les répertoires et fichiers grâce à l'utilisation d'un terminal.

1. Dans cette question, la problématique des droits d'accès n'est pas prise en compte.

(a) Dans l'arborescence précédente, on suppose que le répertoire courant est **camille**.

Parmi les quatre propositions suivantes, déterminer la commande qui permet d'afficher son contenu :

- Proposition 1 : **mv**
- Proposition 2 : **cp**
- Proposition 3 : **ls**
- Proposition 4 : **rm**

(b) Dans l'arborescence précédente, on suppose que le répertoire courant est **camille**.

Parmi les quatre propositions suivantes, déterminer toutes celles qui permettent que le répertoire **cours** d'Alix devienne le répertoire courant :

- Proposition 1 : **cd ../alix/cours**
- Proposition 2 : **cd cours**
- Proposition 3 : **cd alix/cours**
- Proposition 4 : **cd /home/alix/cours**

(c) Dans l'arborescence précédente, on suppose que le répertoire courant est **alix**.

Déterminer l'instruction permettant de copier le fichier **fich1.txt** qui se trouve dans le répertoire d'Alix dans le répertoire **devoirs** de Camille.

```
cp fich1.txt ../camille/devoirs/fich1.txt
```

2. Dans cette question, la problématique des droits d'accès est prise en compte. Camille, depuis le répertoire **camille/devoirs** utilise la commande **ls -l** et obtient le résultat suivant :

```
-rw-r--r-- 1 root    root    1096 juil. 22 17:10 DS1.txt
-rw-r--r-- 1 camille eleve   128  juil. 22 16:03 algo.py
-r--r--r-- 1 camille eleve   128  juil. 22 18:21 NSI.txt
```

(a) Expliquer pourquoi l'instruction **rm DS1.txt** est refusée si l'utilisateur est Camille.

Camille ne peut pas supprimer le fichier DS1.txt car elle n'a pas de droit en écriture sur ce fichier (elle juste un droit en lecture "r"). Le fichier appartient à l'utilisateur root et groupe et other ont comme s droits "r" en lecture mais pas w en écriture (modification / suppression)

On donne ci-après un extrait du mode d'emploi de la commande **chmod** à insérer dans un terminal sous linux.

#### La commande **chmod** de linux

**chmod** : modifie les permissions d'un fichier ou d'un dossier.

**chmod** peut prendre en paramètres les caractères suivants et un nom de fichier

- u** pour l'utilisateur (user)
- g** pour le groupe (group)
- o** pour les autres utilisateurs (other)
- +** pour ajouter des droits
- pour supprimer des droits
- r** pour modifier les droits de lecture (read)
- w** pour modifier les droits d'écriture (write)
- x** pour modifier les droits d'exécution

Exemple :

L'instruction **chmod u+rw x o-x exo.py** ajoute sur **exo.py** des droits en lecture, écriture et exécution à l'utilisateur et supprime les droits d'exécution aux autres utilisateurs.

3. Camille souhaite interdire la lecture du contenu du fichier `algo.py` aux autres utilisateurs (même ceux de son groupe). Elle souhaite également se donner la possibilité de l'exécuter.

Écrire l'instruction permettant de modifier les droits sur `algo.py`.

On suppose que Camille bénéficie déjà des droits de super utilisateur.

```
chmod u+x g-r o-r algo.py
```

## Exercice 2. HTML – javascript

/ 6 pts

- 1) En html, à quoi sert la balise `<head>` ?

- A : à indiquer le titre de la page,
- B : à créer une bannière de page,
- C : à insérer le titre de la page dans la barre du navigateur,
- D : à insérer des informations qui n'apparaîtront pas sur la page.

- 2) Quelle est l'utilité de la balise `<h2>` ?

- A : créer un sous-titre,
- B : créer un lien hypertexte,
- C : insérer une image,
- D : insérer du code css.

- 3) Quelle portion de code est correcte ?

- A :
1. `<h1> Le langage html</h1>`
  2. `<h2> définition </h2>`
  3. `<p> Le langage html ... </p>`
- B :
1. `<h1> Le langage html</h1>`
  2. `<h2> définition </h2>`
  3. `<p> Le langage html ... </p>`
- C :
1. `<h1> Le langage html`
  2. `<h2> définition </h2>`
  3. `<p> Le langage html ... </p>`
- D :
1. `<h1> Le langage html`
  2. `<h2> définition </h2>`
  3. `Le langage html ... </p>`

- 4) Quelle instruction permet de créer un lien vers la page d'accueil du site thalesm ?

- A : `<a href = "https://hmalherbe.fr/thalesm/gestclasse"> cliquez ici ! </a>`
- B : `<a src = "https://hmalherbe.fr/thalesm/gestclasse"> cliquez ici ! </a>`
- C : `<a href = "cliquez ici !"> https://hmalherbe.fr/thalesm/gestclasse </a>`
- D : `<a src = "cliquez ici !"> https://hmalherbe.fr/thalesm/gestclasse </a>`

- 5) Entre les balises `<head>` et `</head>` d'un code html, on place l'instruction : `<title>Ceci est un titre</title>`.

Où apparaît le texte "Ceci est un titre" si ce code est interprété par un navigateur ?

- A : En titre principal dans la page.
- C : Dans l'onglet du navigateur.
- B : En titre secondaire dans la page.
- D : Dans le bas de page.

- 6) Quelle est la bonne instruction pour insérer une image nommée image.jpg dans une page web ?  
<title>Ceci est un titre</title>.
- A : <a> image.jpg </a>  
 B : <a src="image.jpg">  
 C :   
 D : <img>image.jpg</img>
- 7) Quel est le rôle de l'attribut "alt" de la balise <img> ?
- A : Donner une image alternative si un lien ne fonctionne pas.  
 B : Donner un lien alternatif si le premier lien ne fonctionne pas.  
 C : Donner un lien alternatif si une image ne s'affiche pas.  
 D : Donner une description d'une image si elle ne s'affiche pas.
- 8) Quelle balise permet d'insérer du code Javascript dans du code HTML ?
- A : <insert>  
 B : <script>  
 C : <code>  
 D : <java>
- 9) Comment écrire en Javascript une condition qui teste si a est égal à 2 ?
- A : if (a = 2)  
 B : if (a == 2)  
 C : if (a != 2)  
 D : if (a <> 2)
- 10) Comment créer en Javascript une fonction nommée "test" qui ne prend pas de paramètre ?
- A : function test()  
 B : def test()  
 C : def test  
 D : function : test
- 11) On veut écrire un programme en Javascript qui affiche "OUI" si la valeur de la variable c est strictement supérieure à 2 et NON sinon. Quel est le bon programme ?
- A : if (c >= 2) {  
    alert("OUI");  
    else { alert("NON") ; }  
}
- B : if (c > 2) {  
    alert("OUI");  
    else { alert("NON") ; }  
}
- C : if (c > 2) then{  
    alert("OUI");  
    else { alert("NON") ; }  
}
- D : while (c > 2) {  
    alert("OUI");  
    else { alert("NON") ; }  
}

12) On considère le code Javascript suivant :

```
function fact(n) {  
  let result = 1;  
  for (i = 1; i <= n; i = i + 1) {  
    result = result * i;  
  }  
  return result;  
}
```

Quel code Python produira le même résultat ?

A :

```
def fact(n)  
{ result = 1  
  for i in range (1,6):  
    result = result * i  
  return result }
```

B :

```
function fact(n):  
  result = 1  
  for i in range (1,6):  
    result = result * i  
  return result
```

C :

```
def fact(n):  
  result = 1  
  for i in range (1, n+1):  
    result = result * i  
  return result
```

D :

```
fact(n):  
  result = 1  
  for i in range (1,6):  
    result = result * i  
  return result
```

## Exercice 3. Python : structure conditionnelle

/ 4 pts

Indiquer la valeur finale de la variable b dans les algorithmes suivants :

<b>Algorithme 1</b>		<b>Algorithme 2</b>	
<pre>a=3 b=6 if a&gt;5 or b!=3:     b=4 else:     b=2</pre>		<pre>a=7 b=12 if a&gt;5:     b=b-4 if b&gt;=10:     b=b+1</pre>	
Valeur de <i>b</i> :	4	Valeur de <i>b</i> :	8
<b>Algorithme 3</b>		<b>Algorithme 4</b>	
<pre>a=2 b=5 if a&gt;8:     b=10 elif a&gt;6:     b=3</pre>		<pre>a=2 b=0 if a&lt;0:     b=1 elif a&gt;0 and a&lt;5:     b=2 else:     b=3</pre>	
Valeur de <i>b</i> :	5	Valeur de <i>b</i> :	2

## Exercice 4. Python : boucles

/ 5 pts

Indiquer pour chaque fragment de code le résultat affiché en sortie.

Code	Résultat en sortie ?
<pre>s = 0 for i in range(1,4):     s = s + 1 print(s)</pre>	3
<pre>n = 0 while n&lt;15 :     n = n + 2 print(n)</pre>	16
<pre>a= 26 b= 3 q = 0 while a &gt; b:     q = q + 1     a = a - b print(q , a)</pre>	8,2
<pre>res = 1 for i in range(3):     res = res * 2 print(res)</pre>	2 4 8
<pre>n = 10 while n&gt;=11 :     n = n + 2 print(n)</pre>	10

**Exercice 5. Python : programme : placement à intérêts composés**

/ 6 pts

- 1) Compléter le programme ci-contre pour qu'il affiche le montant présent sur un compte d'épargne rémunéré avec un taux de 3.5% pendant 15 ans, sachant que la somme initialement versée sur le compte est 1 500 €.

```
s = 1500
for i in range( 15
    s = s*1.035
print(s)
```

- 2) Modifier le programme pour qu'il affiche cette fois le nombre d'années nécessaires pour atteindre au moins 5000 euros sur le compte d'épargne.

```
s = 1500
nb_annees = 0
while s < 5000:
    s = s*1.035
    nb_annees += 1
print(nb_annees)
```

**Exercice 6. Python : fonction somme**

/ 4pts

Ecrire ci-dessous une fonction `somme(n)` qui prend en paramètre un entier naturel, et renvoie en sortie la somme des entiers de 1 à n.

```
def somme(n):
    """
    Calcule et retourne la somme des entiers de 1 à n
    :param n: l'entier n
    :return s: la somme des entiers de 1 à n
    >>> somme(10)
    55
    >>> somme(100)
    5050
    """
    s = 0
    for i in range(1,n+1):
        s = s + i
    return s
```

## Exercice 7. Python : Jeu du BlackJack

/ 5 pts

On veut programmer un jeu qui s'apparente au BlackJack mais se joue avec deux dés. Le joueur est opposé à la banque, **le but est de s'approcher de 21 sans dépasser ce total.**

A chaque tour le joueur choisit de lancer 0, 1 ou 2 dés.

- S'il choisit 0, la partie s'arrête
- Sinon, on calcule le score total des dés. Le banquier lance alors le même nombre de dés que le joueur.
  - Si l'un des deux dépasse 21, il a perdu et l'autre a gagné
  - Si les deux dépassent 21, la partie est déclarée nulle
  - Sinon, le joueur peut continuer à lancer à nouveau un certain nombre de dés pour se rapprocher de 21.

Compléter le script ci-dessous afin que le programme réponde au cahier des charges qui simule ce jeu.

```

from random import *
TotalJoueur=0
TotalBanque = 0
continuer = True
while continuer==True :
    n=int(input("Combien de dés ?"))
    if n > 0 :
        d = 1
        print ("au joueur de lancer !")
        while d <= n :
            res=randint(.....)
            print ("Dé numéro", d, ":", res)
            TotalJoueur = .....
            d = d + 1
        print ("Total du joueur", TotalJoueur)
        print ("Au tour de la banque de lancer :")
        d = 1
        while d <= n :
            res = randint(.....)
            print ("Dé numéro", d, ":", res)
            .....
            d = d + 1
        print ("Total de la banque", TotalBanque)
        if TotalBanque > 21 ..... TotalJoueur > 21 :
            continuer = False
    else :
        continuer = False
# On regarde les résultats :

if TotalJoueur > 21 :
    if TotalBanque ..... :
        print ("Match nul")
    else :
        print ("Vous avez perdu, vous avez dépassé 21.")
else :
    if TotalBanque > 21 :
        print ("Vous avez gagné, la banque a dépassé 21.")
    .....
    .....
    .....
    .....
    .....

```



Indication : Voici l'aide sur la fonction randint() obtenue dans une console Python en saisissant la commande help(randint) :

*randint(a, b) method of random.Random instance*  
*Return random integer in range [a, b], including both end points.*

```
from random import *
TotalJoueur = 0
TotalBanque = 0
continuer = True
while continuer:
    n = int(input("Combien de dés ?"))
    if n > 0:
        d = 1
        print("au joueur de lancer !")
        while d <= n:
            res = randint(1,6)
            print("dé numéro",d,":",res)
            TotalJoueur = TotalJoueur + res
            d = d + 1
        print("Total du joueur",TotalJoueur)
        print("Au tour de la banque de lancer :")
        d = 1
        while d <= n:
            res = randint(1,6)
            print("dé numéro",d,":",res)
            TotalBanque = TotalBanque + res
            d = d + 1
        print("Total de la banque",TotalBanque)
        if TotalBanque > 21 or TotalJoueur > 21:
            continuer = False
    else:
        continuer = False
#
# On regarde les résultats :
#
if TotalJoueur > 21:
    if TotalBanque > 21:
        print("Match nul")
    else:
        print("Vous avez perdu, vous avez dépassé 21.")
else:
    if TotalBanque > 21:
        print("Vous avez gagné, la banque a dépassé 21.")
    else:
        print("Partie non terminée")
        print("Votre score : ",TotalJoueur)
        print("Score de la banque : ",TotalBanque)
```