

Spécialité NSI Première	DEVOIR SURVEILLE DE	Jeudi 7 avril 2022
Lycée d'Avesnières		Durée : 2 h
Année scolaire 2021-2022		Calculatrice interdite
	NSI	
	N° 7	

L'énoncé complet est à rendre avec la copie. Vous pouvez cependant détacher les feuilles à condition d'écrire votre nom sur toutes les feuilles et de les rendre toutes.

NOM : **Prénom :**

Exercice 1 (5 points)

- 1) Convertir les nombres entiers 1000_2 , 10011_2 et 100101_2 de la base binaire à la base décimale.
- 2) Convertir les nombres entiers 4_{10} , 12_{10} et 18_{10} de la base décimale à la base binaire.
- 3) Réalisation d'un convertisseur binaire vers décimal.

Document 1 : Objectif à atteindre

- On veut créer une page Web comportant du code JavaScript pour obtenir un convertisseur binaire vers décimal similaire à la figure ci-dessous.

Convertisseur binaire vers décimal à 8 bits

Cochez :

2^7
 2^6
 2^5
 2^4
 2^3
 2^2
 2^1
 2^0

Le nombre représenté en binaire ci-dessus est 0.

- Utilisation
 1. L'utilisateur coche certaines cases à cocher.
 2. Dès que l'utilisateur a coché une ou plusieurs cases, la valeur en décimal s'affiche ici.

Exemple

Convertisseur binaire vers décimal à 8 bits

Cochez :

2^7
 2^6
 2^5
 2^4
 2^3
 2^2
 2^1
 2^0

Le nombre représenté en binaire ci-dessus est 24.

Document 2 : Fichiers (incomplets) déjà écrits au départ

- binaire.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Convertisseur binaire</title>
  <link href="....." type="text/css" rel="stylesheet">
</head>
<body>
  <h3>Convertisseur binaire vers d&eacute;ecimal &agrave; 8 bits</h3>
  <label>Cochez :</label>
  <form autocomplete="off"> <!-- Pour effacer les cases lors du rafraichissement -->
  <div class="ligne">
    <input type="checkbox" id="b7" onChange=".....">
    <label for="b7"><var>2<sup>7</sup></var></label>
  </div>
  <div class="ligne">
    <input type="checkbox" id="b6" onChange=".....">
    <label for="b6"><var>2<sup>6</sup></var></label>
  </div>
  <div class="ligne">
    <input type="checkbox" id="b5" onChange=".....">
    <label for="b5"><var>2<sup>5</sup></var></label>
  </div>
  <div class="ligne">
    <input type="checkbox" id="b4" onChange=".....">
    <label for="b4"><var>2<sup>4</sup></var></label>
  </div>
  <div class="ligne">
    <input type="checkbox" id="b3" onChange=".....">
    <label for="b3"><var>2<sup>3</sup></var></label>
  </div>
  <div class="ligne">
    <input type="checkbox" id="b2" onChange=".....">
    <label for="b2"><var>2<sup>2</sup></var></label>
  </div>
  <div class="ligne">
    <input type="checkbox" id="b1" onChange=".....">
    <label for="b1"><var>2<sup>1</sup></var></label>
  </div>
  <div class="ligne">
    <input type="checkbox" id="b0" onChange=".....">
    <label for="b0"><var>2<sup>0</sup></var></label>
  </div>
</form>
<p>Le nombre repr&eacute;sent&eacute; en binaire ci-dessus est
<code id="nombre">0</code>.</p>
<script src="....."></script>
</body>
</html>
```

- script.js

```
function leibniz(){
    var s = .....; // Initialisation de la variable qui cumule la somme des bits.
    if (document.getElementById("b0").checked){s = s + 1}
    if (document.getElementById("b1").checked){s = s + 2}
    if (document.getElementById("b2").checked){.....}
    if (document.getElementById("b3").checked){.....}
    if (document.getElementById("b4").checked){.....}
    if (document.getElementById("b5").checked){.....}
    if (document.getElementById("b6").checked){.....}
    if (document.getElementById("b7").checked){.....}
    document.getElementById(".....").innerHTML = s
}
```

- style.css

```
body {
    margin: 0 20%;
    background-color: #f0f0f0;
}

h3 {
    text-align: center;
    border: 1px solid #aaaaaa;
    border-radius: 10px;
    padding: 20px;
    background-color: #dadada;
}

.ligne {
    margin-left: 24px;
    display: inline-block;
}
```

a) Fichier HTML

Complétez sur cet énoncé à **rendre avec la copie** les zones en pointillés pour que le fichier HTML soit relié au fichier JavaScript et à la feuille de styles.

b) Fichier JavaScript

Compléter sur cet énoncé à **rendre avec la copie** les huit zones en pointillés pour que la fonction `leibniz()` fasse les tâches suivantes :

- Initialiser la variable `s` avec la valeur 0.
- Cumuler dans la variable `s` les puissances de 2 si les cases correspondantes sont cochées.
- Affecter au texte situé dans la balise `<code>` de la page HTML la valeur de `s` ainsi formée.

Exercice 2 (5 points)

1) Recopier sur la copie et compléter l'algorithme suivant en langage naturel

```
Algorithme : tri_insertion(tableau)
    pour ...
```

2) Recopier sur la copie et compléter l'algorithme suivant implémenté en langage Python

```
def tri_insertion(tableau_a_trier):
    for ...
```

3) On s'intéresse dans cette question aux sommets du monde dont l'élévation au-dessus du niveau moyen de la mer est supérieure à 1500 m.

On dispose pour cela d'un fichier csv et d'un fichier Python présentés ci-dessous.

Document 3 : Fichier sommets.csv (le début et la fin)

```
Elevation,Name,Lat_dec,Long_dec
2180,Jabal Asoteriba (new),21.86500,36.50916
2218,Jabal Erba (new),20.74583,36.84083
5468,Cerro Castillo,-33.97583,-69.88166
2987,Scud Peak,57.24111,-131.16750
6407,Mintaka Sar,36.95055,74.83583
2949,Mt. Ovington,54.14305,-120.57416
.....
.....
3190,Mt. Cleveland,48.92472,-113.84777
3588,Arc Dome,38.83277,-117.35305
2871,He Devil,45.32388,-116.54833
2228,Abercrombie Mtn.,48.92833,-117.46000
2791,Mt. Lemmon,32.44277,-110.78833
2975,Chiricahua Peak,31.84638,-109.29111
2751,Mt. Eddy,41.32000,-122.48000
2885,Miller Peak,31.39277,-110.29305
1629,Mt. Marcy,44.11277,-73.92361
3678,Mt. Moriah,39.27333,-114.19861
```

Document 4 : Fichier tri_sommets.py (incomplets) déjà écrit au départ

- tri_sommets.py

```
import csv

def creerListes(nom_fichier):
    """
    Importe un fichier csv existant et le transforme en liste de listes.

    Parametres :
    -----
        nom_fichier : Du type chaine de caracteres.
                     C'est le nom du fichier csv encode en utf-8,
                     par exemple 'fruits.csv'

    Renvoie :
    -----
        Une liste de listes.
    """

    with open(nom_fichier, mode='r', encoding='utf-8-sig') as fichier_ouvert:
        return [ligne for ligne in csv.reader(fichier_ouvert)]

def tri_insertion(liste):
    # Tri par insertion
    tableau_a_trier = list(liste) # Crée une copie de liste

    .....

    .....

    .....

    .....

    .....

    .....

    .....

    return tableau_a_trier

ma_liste = creerListes("sommets.csv")[1:]

ma_liste_triee = tri_insertion(ma_liste)
```

a) La fonction `creerListes(nom_fichier)` permet d'obtenir le tableau `ma_liste` suivant :

```
[['2180', 'Jabal Asoteriba (new)', '21.86500', '36.50916'], ['2218', 'Jabal Erba (new)', '20.74583', '36.84083'], ['5468', 'Cerro Castillo', '-33.97583', '-69.88166'], ['2987', 'Scud Peak', '57.24111', '-131.16750'], ['6407', 'Mintaka Sar', '36.95055', '74.83583'], ['2949', 'Mt. Ovington', '54.14305', '-120.57416'], ...]
```

Complétez sur cet énoncé à **rendre avec la copie** la fonction `tri_insertion(liste)` pour que le tableau `ma_liste_triee` contienne

```
[['8848', 'Mt. Everest', '27.98833', '86.92416'], ['8611', 'K2', '35.88250', '76.51333'], ['8586', 'Kangchenjunga', '27.70333', '88.14750'], ['8485', 'Makalu', '27.89000', '87.08833'], ['8188', 'Cho Oyu', '28.09416', '86.66083'], ['8167', 'Dhaulagiri I', '28.69666', '83.48833'], ...]
```

où les sommets sont triés par **altitudes** dans l'ordre des altitudes **décroissantes**. On utilisera l'algorithme du **tri par insertion**.

b) Écrire **sur la copie** le code Python d'une fonction `plus_hauts1(liste_triee, n)` qui prend en argument la liste de listes Python `ma_liste_triee` et un entier `n` et qui renvoie la liste de listes des `n` plus hauts sommets.

Exemple

```
liste_plus_hauts = plus_hauts1(ma_liste_triee, 3)
```

`liste_plus_hauts` contient

```
[['8848', 'Mt. Everest', '27.98833', '86.92416'], ['8611', 'K2', '35.88250', '76.51333'], ['8586', 'Kangchenjunga', '27.70333', '88.14750']]
```

c) Écrire **sur la copie** le code Python d'une fonction `plus_hauts2(liste_triee, altitude)` qui prend en argument la liste de listes Python `ma_liste_triee` et l'entier `altitude` et qui renvoie la liste de listes des sommets dont l'altitude est supérieure ou égale à `altitude`.

Exemple

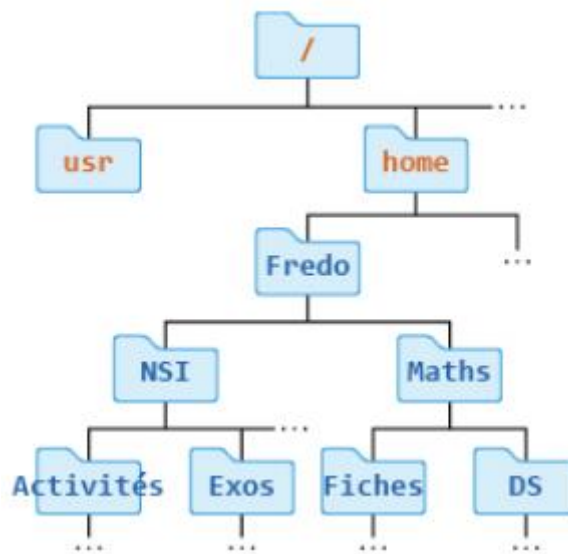
```
liste_plus_hauts2 = plus_hauts2(ma_liste_triee, 8600)
```

`liste_plus_hauts2` contient

```
[['8848', 'Mt. Everest', '27.98833', '86.92416'], ['8611', 'K2', '35.88250', '76.51333']]
```

Exercice 3 (7 points)

Voici un extrait du système de fichiers de l'ordinateur de Fredo, qui a pour système d'exploitation Linux.



1) Fredo se connecte. Il voit l'écran suivant :

```
debian login: Fredo
Password:
Linux debian 4.12.0-rc6-q48ec1f0-dirty #21 Fri Aug 4 21:02:28 CEST 2017 i586

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Fredo@debian:~$
```

- a) Fredo est-il connecté comme simple utilisateur ou comme super utilisateur (root) ? Justifier.
 - b) Quel est le répertoire courant ? Justifier.
- 2) Fredo veut se rendre dans le répertoire NSI. Il saisit la commande `cd /home/Fredo/NSI`
- a) Cette commande est-elle correcte ?
 - b) Si oui, a-t-il indiqué un chemin absolu ou relatif ? Justifier.
- 3) Depuis le répertoire Fredo, déterminer le chemin absolu et le chemin relatif pour aller dans le répertoire Exos.
- 4) Depuis le répertoire Fiches, déterminer le chemin absolu et le chemin relatif pour aller dans le répertoire DS.
- 5) Quel est l'intérêt d'utiliser un chemin relatif par rapport à un chemin absolu ?

- 6) Quel est l'intérêt d'utiliser un chemin absolu par rapport à un chemin relatif ?
- 7) Fredo est dans le répertoire NSI. Il veut créer le répertoire Projets dans le répertoire NSI au même niveau dans l'arborescence que les répertoires Activités et Exos. Écrire la commande.
- 8) Le répertoire courant étant NSI, Fredo veut créer le répertoire Anglais au même niveau que les répertoires NSI et Maths. Écrire la commande utilisée.
- 9) Le répertoire courant étant NSI, Fredo veut se déplacer dans le répertoire Anglais. Écrire la commande utilisée.
- 10) Le répertoire courant étant NSI, Fredo veut déplacer le répertoire Activités dans le répertoire Exos à l'aide de la commande `mv`.
 - a) Écrire la commande utilisée.
 - b) Cela permet-il de déplacer tout le contenu du répertoire Activités dans le répertoire Exos?
- 11) Fredo est à nouveau dans son répertoire personnel Fredo.
Il veut supprimer tout son répertoire Maths. Il teste la commande `rm Maths`.
 - a) Que remarque-t-il ? Justifier.
 - b) Quel conseil pouvez-vous donner à Fredo ?

Exercice 4 (3 points)

Voici un extrait de la documentation des codes d'instructions du 6502. On y donne notamment le code hexadécimal de l'instruction en langage machine, ainsi que le nom de l'instruction et sa signification en assembleur.

Par exemple, le code hexadécimal `A2 00` correspond à l'instruction LDX avec l'opérande `00`.

Code hexadécimal de l'instruction	Nombre d'opérandes de l'instruction	Rôle de l'instruction du langage machine	Nom de l'instruction (et de l'opérande)	Signification du nom de l'instruction
<code>A2</code>	1	Charger l'opérande dans le registre X.	<code>LDX #valeur</code>	<i>LoaD X</i>
<code>A9</code>	1	Charger l'opérande dans le registre A.	<code>LDA #valeur</code>	<i>LoaD A</i>
<code>E8</code>	0	Incrémenter le registre X de 1.	<code>INX</code>	<i>INcrement X</i>
<code>69</code>	1	Ajouter l'opérande au registre A.	<code>ADC #valeur</code>	<i>ADdition with Carry</i>
<code>E0</code>	1	Comparer l'opérande au registre X.	<code>CPX #valeur</code>	<i>ComPare X</i>
<code>D0</code>	1	Si la comparaison précédente est fautive, ajouter l'opérande à la valeur du pointeur d'instructions (attention, la notation est en complément à deux).	<code>BNE #valeur</code>	<i>Branch if Not Equal</i>

- 1) A l'aide de la documentation précédente, complétez sur cet énoncé à rendre avec votre copie le tableau ci-dessous.

Instructions données en langage machine (ici le microprocesseur 6502)	Instructions symboliques en langage assembleur (nom de l'instruction et de l'opérande)	Description de l'action réalisée
<code>A9</code>		
<code>0B</code>		
<code>69</code>		
<code>15</code>		
<code>69</code>		
<code>0A</code>		

- 2) Que réalise ce programme ?
- 3) Quelle valeur contient l'accumulateur à la fin de l'exécution ?