SÉANCE PROGRAMMATION (CODAGE, ALGORITHMIQUE, HTML) DURÉE : 1H

Objectifs :

- connaître la notion de **codage** de l'information et quelques exemples (binaire, ASCII, hexadécimal, Unicode) et de compression de code (exemple Huffman)
- connaître la notion d'**algorithme** (dont les variables et les boucles) et savoir prévoir le résultat d'algorithmes simples
- savoir éditer un **code de balisage HTML** simple (code des pages web)

PIX : compétence 3.4. Programmer

Ecrire des programmes et des algorithmes pour répondre à un besoin (automatiser une tâche répétitive, accomplir des tâches complexes ou chronophages, résoudre un problème logique, etc.) et pour développer un contenu riche (jeu, site web, etc.) (avec des environnements de développement informatique simples, des logiciels de planification de tâches, etc.).

THÉMATIQUES ASSOCIÉES

Algorithme et programme ; Représentation et codage de l'information ; Complexité ; Pensée algorithmique et informatique ; Collecte et exploitation de données massives ; Intelligence artificielle et robots

Rappel : + = pour en savoir plus, non exigible pour l'UE
Système d'exploitation à utiliser pour ce TD : Windows ou Linux *(et Linux pour les +)*
Certains exercices utilisent des fichiers déposés en ligne sur ecampus, zone Pix_programmation_securite, dossier zippé de fichiers supports à télécharger puis à extraire.

Exercice 1 Codage binaire

Combien de valeurs numériques peut-on coder avec 5 bits ?

Exercice 2 Codage ASCII

1) Trouver le mot dont le codage ASCII hexadécimal est le suivant :

43 4F 44 41 47 45 2D 61 73 63 69 69

(S'aider de la table ASCII disponible en annexe du cours et TD et dans les fichiers supports)

2) + Pour afficher **en clair** les **octets** contenus dans un fichier quel qu'il soit, des outils sont disponibles sur la plupart des systèmes d'exploitation, outils à utiliser via une fenêtre de Terminal.
Afficher le contenu du fichier **codage_1.png** à l'aide d'un outil approprié comme **hd** dans une **fenêtre Terminal** *(voir explications en italique page suivante).* Que dire des 4 premiers octets ?
Faire de même avec le fichier **codage_2.pdf** avec les 8 premiers caractères.
Que dire de l'affichage en hexadécimal ?
*L'utilitaire **hd** (hexa dump) peuvent être utilisés comme commande dans une fenêtre « terminal ».*
Syntaxe pour hd (hexa dump) : `hd {fichier} | less`

Explications :
En pratique, le format de la plupart des fichiers (binaire ou non) peut se déduire des premiers octets

d'un fichier. (c'est ce que fait la commande 'file' : lire les premiers octets d'un fichier pour déterminer le format des données qu'il contient)
octal : base 8 (chiffres de 0 à 7)
hexa (ou hexadécimal) : base 16 (chiffres de 0 à 9 et de A à F)
L'octal était prépondérant à l'époque de l'ASCII (qui n'était codé que sur 7 bits au début : de /000 à /177 en octal) où chaque caractère était noté sur 3 digits (3 chiffres)
L'hexadécimal est devenu le standard depuis l'adoption de l'octet (8 bits) puisque toutes les valeurs possibles tiennent sur 2 digits (caractères de 0 à 9 et de A à F)
où les 256 valeurs possibles pour un octet sont comprises entre 0x00 et 0xFF.
Les préfixes standards sont '/' précédant une valeur en octal et 0x ou $ ou H précédant une valeur en hexadécimal
exemple : /200 = 0x80 = 128

Exercice 3 + Codage de Huffman

Comme le Morse avant lui, le codage de Huffman (inventé en 1952) utilise un **codage de taille différente suivant les caractères**, les caractères les plus courants ayant un codage plus court. C'est un algorithme de compression sans perte, très astucieux !

1) Décoder la séquence : « **10 0000 1010 0101 0011 1110** » à l'aide de la table suivante :

Exemple de code			
l	111	e	110
i	00	f	01
d	100	c	101

Noter que ce codage permet de distinguer les caractères sans utiliser de code séparateur : dans notre exemple, les codes à 2 bits commencent par 0 et ceux à 3 bits par 1.

2) Quel est le taux de compression moyen par rapport à l'ASCII ?
Taux de compression : calcul = (avant – après) / avant
(10 % , 50 % , 70 % ou 90%)
avant signifie nombre de bits en ASCII pour le mot que vous avez décodé
après signifie nombre de bits avec Huffman pour ce même mot

Remarque : l'ASCII peut être codé sur 7 ou 8 bits, dans ce cas un bit "inutile" est mis à 0.
Ici on considère 8 bits.

Exercice 4 Conversion UTF

Quelle est la traduction en français du mot suivant dont les caractères Unicode sont :
« 96FB 52D5 706B 8ECA » ?
On pourra s'aider de convertisseurs en ligne, par exemple :
http://hapax.qc.ca/conversion.fr.html
Indiquer en quelle langue le mot est encodé, puis le traduire.
S'aide d'*outils de traduction automatique par exemple :* https://translate.google.fr/

Exercice 5 Algorithme en symbolique

Si la fonction « Ecrire(n) ; » écrit le chiffre n où se trouve le curseur, quel sera le résultat affiché par l'algorithme suivant ?

```
Ecrire(1) ;
Pour i allant de 2 à 20 faire
     Si i est impair et premier, Alors Ecrire(i) ;
```

Exercice 6 +Algorithme en symbolique

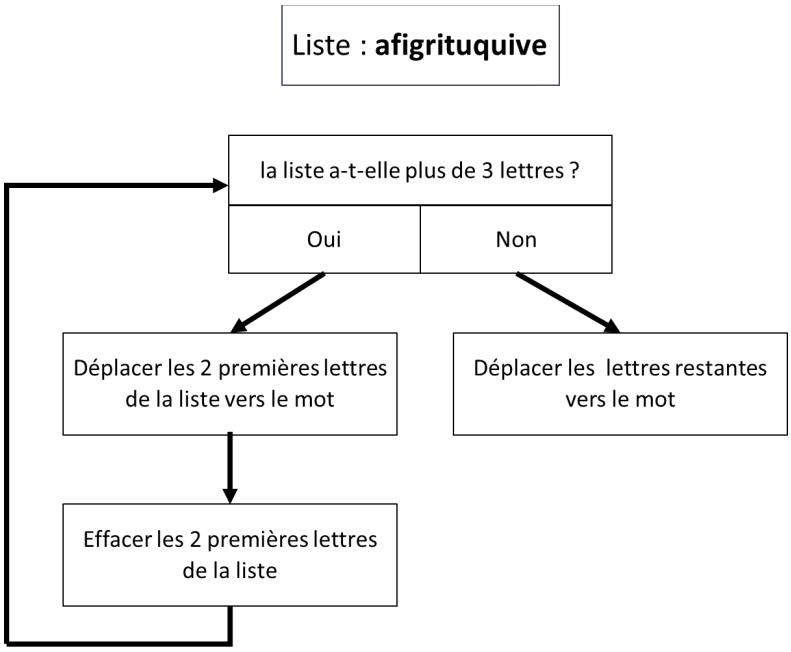
La fonction « Lire() ; » récupère un caractère au clavier et « Ecrire(n) ; » écrit le chiffre n à l'écran. Nous utiliserons l'algorithme ci-après.

```
i ← 0 ;
a ← Lire() ;
Tant que a ≠ '#' faire
     i ← i + 1 ;
     a ← Lire() ;
Ecrire(i) ;
```

Que s'affiche-t-il si l'utilisateur saisit les caractères : **Alésia#** ?

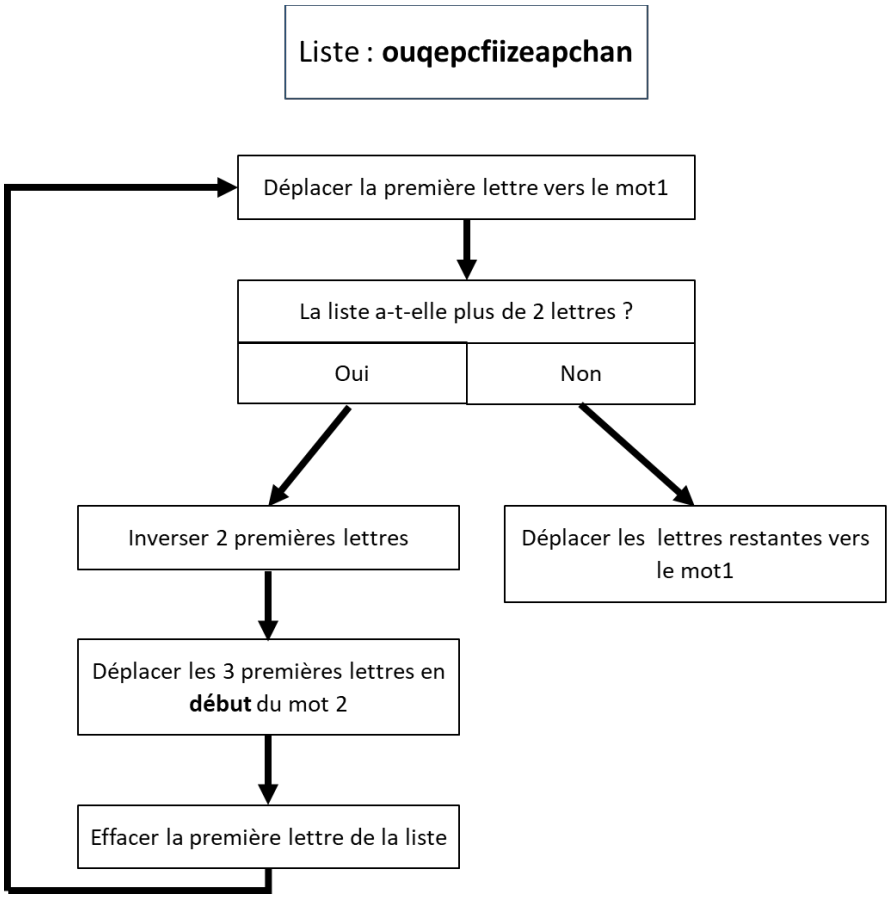
Exercice 7 Algorithme graphique

En suivant l'algorithme suivant, qu'allez-vous écrire ?



Exercice 8 + Algorithme graphique

En suivant l'algorithme suivant, qu'allez-vous écrire ?



Mot 1 : _____

Mot 2 : _____

Exercice 9 Algorithme en script

Soit un Jeu de l'Oie dont le plateau comporte 65 cases numérotées de 1 à 65.
Le joueur commence sur la case no 1. Le but est d'atteindre (ou dépasser) la case 65 en un minimum de coups. Pour chaque tour, le joueur lance un dé.
La fonction Lancer_dé(n) retourne une valeur entière comprise entre 1 et n.
Ecrire(v1, v2, v3, …) écrit la valeur des paramètres v1, v2,…
Enfin, var = expression signifie que la variable « **var** » prend la valeur retournée par *expression*.

```
position = 0
compteur = 0

Tant que position < 65 Faire
Début
     tirage = Lancer_dé(6)
     Ecrire('Le pion avance de ', tirage, 'cases')
     compteur = compteur + 1
     position = ##### + tirage
Fin

Ecrire('La partie a été gagnée en ', €€€€€, ' coups !')
```

Compléter les informations manquantes :

: _____

€€€€€ : _____

Exercice 10 + Scripts comparaison Shell – Python

- Le shell est interprété (et pas particulièrement performant)
- Python est pré-compilé avant l'exécution ce qui le rend plus rapide en exécution (cf. le cours)
- Python est efficace pour certains calculs et possède des librairies optimisées écrites en C

Voici, dans les fichiers nommés eratosthene, une écriture simpliste d'un script réalisant le crible d'Eratosthène (calcul de nombres premiers).
— une version en script Shell classique (fichier **eratosthene.sh**)
— la même version écrite en Python (fichier **eratosthene.py**)

1) Exécuter chaque version en vérifiant que les résultats correspondent.
Pour exécuter un script : (ne pas ouvrir les fichiers !)
- ouvrir un terminal
- taper le nom de l'interpréteur de commande (sh ou python) suivi d'un espace puis faire un glisser-déposer du fichier du script depuis son emplacement sur le disque, puis faire [Return]
(fonctionne normalement dans tous les cas : on lance l'exécuteur de script sh et on lui donne comme paramètre le fichier de script à exécuter)
Ici, pour le script shell :
taper sh puis espace puis faire glisser le fichier eratosthene.sh puis [Return]
(si un message d'erreur est retourné c'est qu'il y a une erreur dans le script)
Il en va de même pour le script en Python, sauf que le nom de l'interpréteur est python : il faudra ***taper python et faire glisser eratosthene.py puis [Return]***

2) A l'aide du programme **time**, mesurer les temps d'exécution de chacun (saisir « man time » pour obtenir de l'aide).
Pour cela, taper : ***time ./eratosthene.sh***
La durée à consulter pour la fonction "time" est la ligne "real".

3) + Donner en paramètre une valeur maximum (ex. 20) à la version en shell pour correspondre à peu près au même temps d'exécution que la version en python.
Pour cela, taper un nombre à la suite du nom du script shell pour limiter les calculs par exemple à la valeur 20 (au lieu de 120 par défaut)
time ./eratosthene.sh 20 [Return]
Quelles conclusions pouvez-vous formuler en comparaison des deux langages ?
(simplicité, rapidité?)

Exercice 11 HTML question cachée

Ouvrir le fichier **html_1.htm** et trouver la réponse demandée.

Réponse : _____

Exercice 12 + HTML mot-clé

Ouvrir le fichier **html_2.htm** et trouver quel mot-clé lui a été affecté.

Réponse : _____

Exercice 13 HTML conversion de couleur

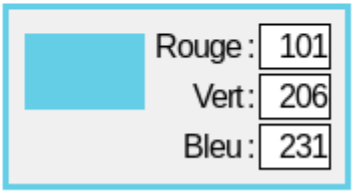
Soit le code source HTML suivant.
Le mot « Bonjour » doit être écrit en couleur rose, comme indiqué par les valeurs RVB fournies.
En syntaxe CSS *(la syntaxe utilisée par le feuilles de style CSS pour les pages web)*, les couleurs peuvent être codées en hexadécimal sur 3 octets (correspondant respectivement au rouge, vert et bleu, précédés du caractère '#').

```
<html>

<head>
<title>Devinette</title>
<meta http-equiv="Content-type" content="text/html" charset="UTF-8">
</head>

<body>
    <Font Color=# ?????? > Bonjour </Font>
</body>

</html>
```



Quelle valeur placer dans le code dans la balise <Font Color=# ??????> ?

?????? : #_____

Exercice 14 + HTML affichage

Quel affichage donne le code source suivant ?

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-type" content="text/html" charset="UTF-8">
<title>Géographie</title>
</head>
<body>
    <h1>Les chaines <u>montagneuses</u></h1>
    <ul>
        <li>Cordillère des Andes</li>
        <li>Himalaya</li>
        <li>Montagnes rocheuses</li>
        <li>Alpes</li>
    </ul>
    <p> Le sommet le plus <i>haut</i> est dans <b>l'Himalaya</b> </p>
</body>
</html>
```

Les chaines montagneuses

- Cordillère des Andes
- Himalaya
- Montagnes rocheuses
- Alpes

Le sommet le plus **haut** est dans l'Himalaya

1

Les chaines montagneuses

- Cordillère des Andes
- Himalaya
- Montagnes rocheuses
- Alpes

Le sommet le plus *haut* est dans **l'Himalaya**

2

Les chaines montagneuses

- Cordillère des Andes
- Himalaya
- Montagnes rocheuses
- Alpes

Le sommet le plus *haut* est dans l'Himalaya

3

Les chaines montagneuses

- Cordillère des Andes
- Himalaya
- Montagnes rocheuses
- Alpes

Le sommet le plus *haut* est dans l'Himalaya

4

Les chaines montagneuses

- Cordillère des Andes
- Himalaya
- Montagnes rocheuses
- Alpes

Le sommet le plus *haut* est dans **l'Himalaya**

5

Les chaines montagneuses

- Cordillère des Andes
- Himalaya
- Montagnes rocheuses
- Alpes

Le sommet le plus haut est dans **l'Himalaya**

6

ANNEXE

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]