

**Examen Sujet 2 2022-2023 - CYBER1 (1h30)**  
**Architecture des Ordinateurs**

NOM :

PRÉNOM :

Vous devez respecter les consignes suivantes, sous peine de 0 :

- Lisez le sujet en entier avec attention
- Répondez sur le sujet
- Ne détachez pas les agrafes du sujet
- Écrivez lisiblement vos réponses (si nécessaire en majuscules)
- Les appareils électroniques sont tous interdits (calculatrices également)
- Ne trichez pas

## 1 Questions (10 points)

### 1.1 (1 points) Rappelez les 14 premières puissances de 2 :

$2^0$	$2^1$	$2^2$	$2^3$	$2^4$	$2^5$	$2^6$	$2^7$	$2^8$	$2^9$	$2^{10}$	$2^{11}$	$2^{12}$	$2^{13}$
1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192

### 1.2 (3 point) Convertissez ces nombres en décimaux. Vous donnerez leur interprétation non-signée puis signée.

	non-signé	signé
% 1000 1011 0101	2229	-1867
% 1011 1100 1110	3022	-1074
% 1001 1010 0011	2467	-1629
\$ DEF	3567	-529
\$ 86A	2154	-1942
\$ E86	3718	-378

**1.3 (4 point) Convertissez ces nombres décimaux en binaire sur 8 bits ou 12 bits, puis en hexadécimal.**

	binaire	hexadécimal
42	% 0010 1010	\$ 2A
1258	% 0100 1110 1010	\$ 4EA
324	% 0001 0100 0100	\$ 144
−86	% 1010 1010	\$ AA

**1.4 (2 points) Convertissez ces nombres en flottants au format IEEE 754 simple précision :**

	exposant	hexadécimal
67,109375	133 / % 1000 0101	\$ 4286 3800
−82,1015625	133 / % 1000 0101	\$ C2A4 3400

**2 Problème (10 points)**

Un étudiant en cybersécurité faisant ses premières expériences en *forensic* (analyse forensique ou investigation numérique en français) a besoin de vous pour convertir plusieurs valeurs et retrouver des informations. Celui-ci a récupéré un disque dur qui servait dans un RAID 1, et il souhaite retrouver les noms de fichiers, le contenu de ces fichiers, mais également analyser quelques programmes stockés dessus.

**2.1 (2 points) Première étape : lecture d'une structure**

L'étudiant a réussi à extraire un secteur du disque dur formaté en FAT16 qui contenait une liste de fichiers et l'identifiant unique associé. Il a isolé 2 *direntries*, les a affichés en hexadécimal, et vous demande de séparer les champs. Utilisez le modèle de structure d'une *direntry* pour séparer les différentes données et remplir les tableaux suivants avec les valeurs hexadécimales.

```

struct direntry {
    char[11] name;
    char      attributes;
    int       first_cluster;
    long      size;
} __attribute__((packed))

```

Les types de données font :

- char : 1 octet (8 bits)
- int : 2 octets (16 bits)
- long : 4 octets (32 bits)

*Rappel : char[11] correspond à un tableau de 11 cases (de 0 à 10)*

```

direntry 1 :
00 00 00 50 4F 55 ...
name 45 54 54 58 54, 1F attr
1st cluster 00 14, 00 00 00 12, size

direntry 2 :
00 00 00 50 49 45 ...
name 44 53 41 56 49, 1F attr
1st cluster 00 32, 00 00 31 55, size

```

	direntry 1 (f1)	direntry 2 (f2)
name	00 00 00 50 4F 55 45 54 54 58 54	00 00 00 50 49 45 44 53 41 56 49
attributes	1F	1F
first_cluster	00 14	00 32
size	00 00 00 12	00 00 31 55

## 2.2 (2 points) Deuxième étape : conversion des noms

L'étudiant semble dubitatif et vous demande de lui présenter des valeurs lisibles. Pour cela, il vous fournit une toute petite table de conversion ASCII... mais vous vous apercevez qu'il n'a pas du tout recopié la partie la plus essentielle de la table (probablement car il ne prenait pas assez de notes) : sa table ne dispose que de la correspondance entre des caractères et leurs valeurs en base 10. Il vous faut donc convertir les valeurs hexadécimales à la main (zut alors).

Concernant les noms de fichiers, la norme FAT16 précise que sur les caractères, les 8 premiers servent à coder le nom du fichier, et les 3 derniers servent à coder l'extension. Vous devez donc ajouter un point pour séparer l'extension du nom de fichier.

Dec	Char	Dec	Char
48	0	49	1
65	A	78	N
66	B	79	O
67	C	80	P
68	D	81	Q
69	E	82	R
70	F	83	S
71	G	84	T
72	H	85	U
73	I	86	V
74	J	87	W
75	K	88	X
76	L	89	Y
77	M	90	Z

f1				P	O	U	E	T	.	T	X	T
f2				P	I	E	D	S	.	A	V	I

### 2.3 (2 points) Troisième étape : conversion des champs

L'étudiant commence à avoir confiance en vous : vous avez trouvé les bons noms de fichiers (et il reconnaît les noms). Afin de pouvoir vous donner les bons clusters à lire, il vous demande de convertir cette fois la taille contenu dans les champs de chaque direntry, ainsi que le numéro du premier cluster, en nombre décimaux (afin qu'il puisse les mettre en paramètre de son outil d'extraction).

	taille du fichier	numéro du premier cluster
direntry 1 (f1)	18	20
direntry 2 (f2)	12629	50

### 2.4 (2 points) Quatrième étape : lecture d'un fichier

L'étudiant est surexcité, vous êtes sur le point de l'aider à retrouver des fichiers et leurs contenus ! En récupérant la vidéo, l'étudiant vous répond qu'il ne peut pas vous la montrer car celle-ci *serait* corrompue. Néanmoins, tout a l'air correct pour le fichier texte.

Il vous demande de convertir maintenant le contenu du fichier en ASCII, et si possible, de retrouver le numéro codé en binaire dans ce texte.

	texte :	numéro :
<pre> 46 4C 41 47 30 31 30 30 31 31 31 30 </pre>	<pre> FLAG 01001110 </pre>	<pre> 68 </pre>

### 2.5 (2 points) Cinquième étape : désassemblage

L'étudiant n'en croit pas ses yeux : vous avez démontré que vous méritez votre bachelor en cyber-sécurité. Il vous demande d'utiliser vos talents de `h4x0rz` pour désassembler des instructions et les analyser. L'étudiant vous indique qu'il a mis 3 instructions MOV, mais il ne se souvient plus de leur ordre. En vous aidant de la documentation fournie, indiquez l'ordre (par 1, 2, ou 3).

Opcode	Instruction	Description
88	MOV r/m8, r8	Déplace une valeur d'un registre 8 bits vers un registre ou une adresse
8A	MOV r8, r/m8	Déplace une valeur 8 bits vers un registre 8 bits
8B	MOV r32, r/m32	Déplace une valeur 32 bits vers un registre 32 bits

<pre> [1] 401c2d: 88 3c 25 16 00 00 00 [2] 401c3b: 8a 24 25 16 00 00 00 [3] 401c34: 8b 1c 25 99 55 64 16 </pre>	<table> <tr> <td>2</td><td><code>mov 0x16,%ah</code></td></tr> <tr> <td>1</td><td><code>mov %bh,0x16</code></td></tr> <tr> <td>3</td><td><code>mov 0x16645599,%ebx</code></td></tr> </table>	2	<code>mov 0x16,%ah</code>	1	<code>mov %bh,0x16</code>	3	<code>mov 0x16645599,%ebx</code>
2	<code>mov 0x16,%ah</code>						
1	<code>mov %bh,0x16</code>						
3	<code>mov 0x16645599,%ebx</code>						