

2 _____
ARCHITECTURE DES ORDINATEURS

Devoir Surveillé 1

1 heure et 20 minutes
sans documents

N.B. : - Travaillez calmement. Ne bâclez pas vos réponses : il vaut mieux traiter correctement moins de questions que tout faire de travers.

- Les réponses aux questions doivent être argumentées et aussi concises que possible.
- Le barème est donné à titre indicatif.

Exercice 1 (50 points)

On considère des entiers codés sur 16 bits, interprétés comme des entiers relatifs codés en complément à deux. Chacun de ces entiers peut (et doit) s'écrire avec quatre chiffres hexadécimaux (on omettra le préfixe « 0x »). Soient les cinq entiers suivants :

M	N	P	Q	R
37AD	6FDA	8AC1	82F8	ABCD

(1.1) (10 points)

Parmi ces 5 entiers, quels sont ceux qui sont négatifs ? Expliquez pourquoi.

(1.2) (10 points)

Classez ces entiers du plus petit au plus grand (par exemple : « $M < N < P < Q < R$ », mais cette réponse choisie arbitrairement n'est probablement pas la bonne...). Justifiez la réponse sur votre copie.

(1.3) (10 points)

Calculez $P + Q$, et donnez le résultat en hexadécimal. Pour cela, effectuez l'addition en binaire directement sur la copie — une réponse brute ne rapportera aucun point. Y a-t-il débordement (*overflow*) ?

(1.4) (20 points)

Calculez $-M$ et $N - M$, et donnez les résultats en hexadécimal. Y a-t-il débordement (*overflow*) ? Comme pour la question précédente, le détail des calculs en binaire doit figurer sur la copie.

Exercice 2 (150 points)

La page suivante contient le texte d'un programme écrit en langage y86. Tous les commentaires ayant malencontreusement été effacés, il vous faut partir de zéro pour comprendre ce que fait ce programme, en répondant aux questions ci-dessous.

Attention : les réponses qui, au lieu d'expliquer, paraphrasent simplement le code (telles que « on soustrait edx à ebx ») ne rapporteront aucun point.

(2.1) (20 points)

Quels sont les paramètres de la fonction `mystere`? Expliquez quelle partie du code permet de répondre à cette question.

Si ce code avait été produit par un compilateur C, quel aurait été le prototype de la fonction `mystere`?

(2.2) (10 points)

Quel est le rôle de l'instruction d'adresse `0x077`?

(2.3) (20 points)

Quel est le rôle de l'instruction d'adresse `0x034`? Pourquoi cette instruction prend-elle en paramètre l'entier 20?

(2.4) (10 points)

Quelle est la valeur qui sera stockée à l'adresse d'étiquette `x` à la fin de l'exécution du programme? À quoi correspond-elle? Justifiez votre réponse.

(2.5) (20 points)

De quelle manière les données en mémoire situées entre les adresses `0x07c` et `0x09f` incluses seront-elles modifiées par l'exécution du programme? Justifiez votre réponse.

(2.6) (30 points)

Dans la fonction `mystere`, quels sont les rôles respectifs des différents registres utilisés? Justifiez vos réponses.

(2.7) (10 points)

Dans la fonction `mystere`, quels sont les rôles respectifs des étiquettes `myst1`, `myst2` et `myst3`?

(2.8) (10 points)

Dites, en une phrase, ce que fait la fonction `mystere`, en prenant en compte les connaissances obtenues des questions précédentes.

(2.9) (20 points)

Expliquez le codage binaire des six octets de l'instruction d'adresse `0x02e`, en tirant éventuellement parti des informations obtenues en considérant d'autres instructions.

0x000:		.pos 0
0x000: 308400020000		init: irmovl pile,%esp
0x006: 308090000000		irmovl t2,%eax
0x00c: a008		pushl %eax
0x00e: 308180000000		irmovl t1,%ecx
0x014: a018		pushl %ecx
0x016: 802800000000		call mystere
0x01b: c08408000000		iaddl 8,%esp
0x021: 40087c000000		rmmovl %eax,x
0x027: 10		halt
0x028: a038		mystere: pushl %ebx
0x02a: a068		pushl %esi
0x02c: a078		pushl %edi
0x02e: 507410000000		rrmovl 16(%esp),%edi
0x034: 501414000000		rrmovl 20(%esp),%ecx
0x03a: 6300		xorl %eax,%eax
0x03c: 506700000000		myst1: rrmovl (%edi),%esi
0x042: 6266		andl %esi,%esi
0x044: 737500000000		je myst3
0x049: 503100000000		rrmovl (%ecx),%ebx
0x04f: 2062		rrmovl %esi,%edx
0x051: 6132		subl %ebx,%edx
0x053: 756400000000		jge myst2
0x058: 406100000000		rmmovl %esi,(%ecx)
0x05e: c08001000000		iaddl 1,%eax
0x064: c08704000000		myst2: iaddl 4,%edi
0x06a: c08104000000		iaddl 4,%ecx
0x070: 703c00000000		jmp myst1
0x075: b078		myst3: popl %edi
0x077: b068		popl %esi
0x079: b038		popl %ebx
0x07b: 90		ret
0x07c:		.align 4
0x07c: 00000000		x: .long 0
0x080: 03000000		t1: .long 3
0x084: 04000000		.long 4
0x088: 06000000		.long 6
0x08c: 00000000		.long 0
0x090: 07000000		t2: .long 7
0x094: 01000000		.long 1
0x098: 02000000		.long 2
0x09c: 00000000		.long 0
0x200:		.pos 0x200
0x200:		pile: