

Nom: _____ Groupe: _____

Prénom: _____ N° Etud: _____

 université angers	FACULTÉ DES SCIENCES <i>Unité de formation et de recherche</i>	Département Informatique
		L3 Informatique Jean-Michel Richer Architecture des Ordinateurs 2024/2025 ¹
jean-michel.richer@univ-angers.fr		

Contrôle Continu Janvier 2025



Aucun document (excepté **une** feuille A4 recto/verso), ni outil électronique n'est autorisé. Vous devez soigner votre écriture (la copie ne doit pas être un brouillon) et expliquer clairement votre raisonnement. L'absence de raisonnement ou des étapes de calcul conduit à l'absence de point. De même, la propreté de la copie est prise en compte ainsi que le fait que vous ne rendez pas votre copie quand cela vous est demandé.

Exercice 1 - 4 pts, 20 min - Soit le code suivant pour lequel les registres contiennent les valeurs indiquées.

Instruction	eax	edx	edi	esi
	102	0	13	5
add eax, 3 ; i0	105			
div edi ; i1				
lea eax, [esi + edx*4] ; i2				
add eax, 63 ; i3				
shr eax, 4 ; i4				
add eax, edi ; i5				
lea edi, [edx + eax*4] ; i6				
popcnt eax, edi ; i7				
xor eax, 7 ; i8				

Indiquez ce que contiennent chacun des registres **eax**, **edx**, **esi**, **edi** en utilisant un tableau où les colonnes représentent les registres et les lignes les instructions i1 à i8. On n'indiquera que les modifications pour le registre concerné à l'exécution de chaque instruction.

Exercice 2 - 5 pts, 30 min - On considère la série qui, à partir d'un entier x positif non nul, génère le terme suivant en respectant les règles :

- ◇ si x est pair, alors le terme suivant est $x = x/2$
- ◇ si x est impair, alors le terme suivant est $x = 3 \times x + 1$

1. Donnez le code assembleur x86 32 bits d'une **fonction récursive** qui affiche chacun des x générés. Dès qu'on entre dans la fonction, on affiche la valeur de x : `printf("%d\n",x);` puis on calcule le x suivant. On arrête la récursion dès que $x = 1$. On pourra s'aider en écrivant au préalable le code C de la fonction.

Par exemple, à partir de 19, on trouve 58, 29, 88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, puis finalement 1.

Exercice 3 - 5 pts, 30 min - On considère la fonction C suivante qui crypte une chaîne de caractères s qui contient des lettres majuscules et d'autres caractères de ponctuation en n'encodant que les lettres (les autres caractères sont ignorés) :

```
1 void cesar(char *s, int decalage) {
2     char *t = s;
3     while (*s != '\0') {
4         char c = *s;
5         if ((c >= 'A') && (c <= 'Z')) {
6             c = ((c - 65 + decalage) % 26) + 65;
7             *t = c;
8             ++t;
9         }
10        ++s;
11    }
12    *t = *s;
13 }
```

Par exemple, la chaîne "BONJOUR ! HELLO" encodée avec un décalage de 3 donne : "ERQMRXUKHOOR".

1. Donnez un tableau permettant de faire le lien entre variables et registres qui stockeront les données, expliquez vos choix
2. Traduire la fonction en assembleur x86 32 bits

Exercice 4 - 5 pts, 30 min - Répondre aux questions de cours en expliquant et détaillant (si nécessaire) votre réponse : maximum 5 lignes par réponse.

1. comment fonctionne l'instruction `mul` avec des opérandes 32 bits ?
2. comment additionner deux valeurs 64 bits avec des registres 32 bits ?
3. de combien de registres généraux dispose t-on en architecture 32 bits ? quels sont-ils ?
4. en quoi consiste l'alignement mémoire ?
5. à quel dilemme est confronté le développeur dans le cadre de son activité de développement ?