



Nom :		<h1>/ 20</h1>
Prénom :		
Groupe :		
Matricule :		

Exercice 1 : (12 points)

Complétez, à droite, la traduction vers le langage C du code MIPS illustré à gauche.

<pre>foo: sll \$t6 \$a0 2 sub \$sp \$sp \$t6 move \$t4 \$sp sw \$zero 0(\$t4) addi \$t1 \$zero 1 L1: bge \$t1 \$a0 Next andi \$t2 \$t1 1 sll \$t3 \$t1 2 add \$t3 \$t3 \$t4 sw \$t2 0(\$t3) addi \$t1 \$t1 1 j L1 Next: move \$t1 \$zero move \$t2 \$zero sll \$a0 \$a0 2 L2: bge \$t1 \$a0 End add \$t3 \$t4 \$t1 lw \$t3 0(\$t3) add \$t2 \$t2 \$t3 addi \$t1 \$t1 4 j L2 End: move \$v0 \$t2 add \$sp \$sp \$t6 jr \$ra</pre>	<pre>unsigned int foo(unsigned int n) { unsigned int arr[n]; unsigned int total = 0; unsigned int *ptr = arr; ptr[0] = 0; for (int i = 1; i < n; i++) { ptr[i] = i & 1; } for (int i = 0; i < n; i++) { total += ptr[i]; } return total; }</pre>	<p>3 points</p> <p>1 point</p> <p>3 points</p> <p>1 point</p> <p>3 points</p> <p>1 point</p>
---	--	--

Exercice 2 : (8 points)

Soit un système équipé d'un cache mémoire direct de 1Ko où chaque ligne de cache est de 16 octets. Le système possède 4 Go de mémoire RAM. Soit aussi le code C illustré ci-dessous, vous pouvez supposer que le tableau **ARRAY** est alloué à une adresse multiple de la taille d'un bloc mémoire.

```
#define LEN 2048
int ARRAY[LEN];
int main() {
    for (int i = 0; i < LEN - 256; i += 256) {
        ARRAY[i] = ARRAY[i] + ARRAY[i+1] + ARRAY[i+2] + ARRAY[i+256];
        ARRAY[i] += 10;
    }
}
```

a) Donnez les nombres de bits pour les champs étiquette (T), indice(I), et offset (O) de notre cache

T : **22** I : **6** O : **4**

3 x 0.25 point

Justifiez vos réponses

2 points

Offset (O) : $\log_2(\text{taille du bloc}) = \log_2(16) = 4$

Indice (I) : $\log_2(\text{taille du cache} / \text{taille du bloc}) = \log_2(1\text{Ko} / 16) = \log_2(1024 / 16) = \log_2(64) = 6$

Étiquette (T) : taille de l'adresse mémoire en bits – tailles des champs « Offset » et « Indice »
 $= \log_2(4\text{Go}) - 6 - 4 = \log_2(4 \times 2^{30}) - 10 = \log_2(2^{32}) - 10 = 32 - 10 = 22$

b) Donnez le taux de succès pour le code ci-dessus ? Vous pouvez considérer que les expressions sont évaluées de gauche à droite dans le langage C.

57.14%

0.25 point

Justifiez vos réponses

3 points

Chaque itération de la boucle **for** réalise :

Lecture de ARRAY[i] (Echec de première référence)
Lecture de ARRAY[i+1] (Succès)
Lecture de ARRAY[i+2] (Succès)
Lecture de ARRAY[i+256] (Echec de conflit avec ARRAY[i])
Ecriture dans ARRAY[i] (Echec de conflit avec ARRAY[i+256])
Lecture de ARRAY[i] (Succès)
Ecriture dans ARRAY[i] (Succès)

⇒ 3 échecs, 4 succès, taux de succès = $100 \times 4 / 7 = 57,14\%$

c) Vous décidez d'ajouter un cache de niveau 2 à votre système. En ce sens, la taille du cache L1 est réduite de façon à avoir un temps de réponse de 3 cycles d'horloge pour ce cache. En outre, comme le nouveau cache L2 possède une taille plus grande, le temps de réponse de ce cache est de 50 cycles d'horloge. Enfin, supposez un temps d'accès à la mémoire RAM de 500 cycles d'horloge.

Si le taux de succès du cache L1 est de 25% et si le taux de succès du cache L2 est de 90%, quel est le Temps Moyen d'Accès à la Mémoire (AMAT) en cycles d'horloge ?

78

0.25 point

Justifiez vos réponses

1.75 points

AMAT = temps d'accès succès + taux d'échec × pénalité d'échec
 $= 3 + \frac{3}{4} (50 + \frac{1}{10} \times 500)$
 $= 3 + \frac{3}{4} (50 + 50)$
 $= 3 + 75$
 $= 78$