

---

Contrôle Continu  
*Architecture des Ordinateurs*  
2h00 - Aucun document autorisé - Calculatrice interdite

---

**Exercice 1 -(5 pts)** - soit le programme C suivant :

```
int compute(int x[], int y[], int z[], int n) {  
    int i, sum=0;  
    for (i=0;i<n;++i) {  
        z[i] = x[i] * y[i];  
        if (z[i] >= 128) sum += z[i] % 8;  
    }  
    return sum;  
}
```

1. donner l'état de la pile lors de l'appel de la fonction
2. indiquez quels registres vous utilisez pour représenter les variables
3. traduire ce programme en assembleur x86 32 bits

**Exercice 2 -(5 pts)** - Les représentations hexadécimales des nombres flottants suivants sont :

1.5 =	3FC00000	6.0 =	40C00000
3.0 =	40400000	12.0 =	41400000

1. quelle est la représentation hexadécimale de 24.0 ?
2. écrire le code assembleur d'un sous-programme qui multiplie par 2 un nombre flottant passé en paramètre dans le registre EAX, le résultat sera placé dans EAX. On considère qu'il n'y aura pas de dépassement de capacité.
3. faire de même pour la fonction `neg` qui inverse le signe du nombre placé dans EAX

**Exercice 3 -(5 pts)** - On se place dans le cadre de la notation binaire en complément à 2 sur 8 bits.

1. coder la valeur -23
2. coder la valeur -3
3. calculer le produit de ces 2 nombres
4. indiquez si le résultat final est correct ou non et pourquoi.

**Exercice 4 -(5 pts) Questions de cours** - Répondez aux questions au verso de la feuille et n'oubliez pas d'indiquer vos noms et prénoms et de rendre la feuille.

**NOM :**

**Prénom :**

- (1 pt) que fait l'instruction bsr ?

---

- (1 pt) que fait l'instruction popcnt ?

---

- (1 pt) qu'est ce qu'un processeur superscalaire ?

---

---

---

- (1 pt) à quoi sert le pipelining ?

---

---

---

- (2 pts) comment multiplier efficacement le registre EAX par 100 sans utiliser l'instruction MUL ?

---

---

---

---

soit le code suivant qui sera traduit comme vu en cours et sans optimisation

```
int *f(char *t) {  
    int i;  
    for (i=0; t[i]!='0'; ++i) ;  
    return &t[i];  
}  
}
```

- (1 pt) comment obtenir l'ancienne valeur de ebp ?

---

- (1 pt) comment obtenir la valeur de i ?

---

- (1 pt) comment obtenir la valeur de retour de la fonction ?

---

- (1 pt) quel est le code ascii de '0' ?

---