

BASES DES SYSTÈMES INFORMATIQUES

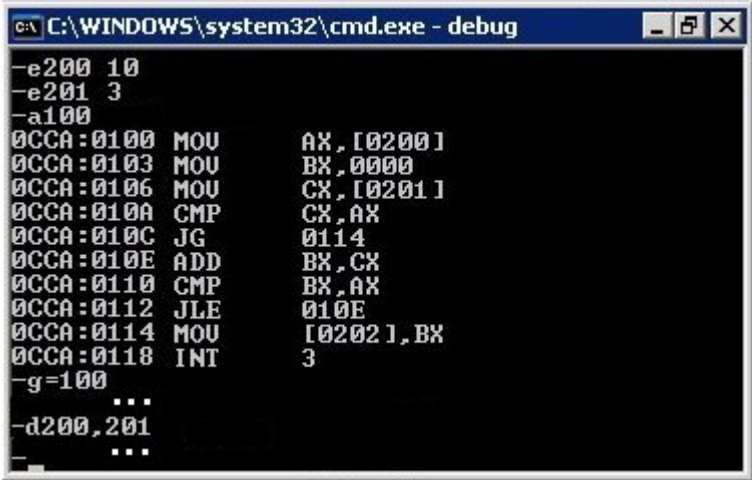
Jeudi 7 janvier 2010

Contrôle Assembleur

-
- i. Tous les documents sont autorisés
 - ii. Aucun matériel électronique (téléphone, calculatrice, ...) n'est autorisé
 - iii. Bon courage à tous
-

Exercice 1

1. Commenter le code suivant.



```
C:\WINDOWS\system32\cmd.exe - debug
-e200 10
-e201 3
-a100
0CCA:0100 MOV     AX,[0200]
0CCA:0103 MOV     BX,0000
0CCA:0106 MOV     CX,[0201]
0CCA:010A CMP     CX,AX
0CCA:010C JG     0114
0CCA:010E ADD     BX,CX
0CCA:0110 CMP     BX,AX
0CCA:0112 JLE    010E
0CCA:0114 MOV     [0202],BX
0CCA:0118 INT     3
-g=100
...
-d200,201
...
```

2. Que fait le programme et quelles sont les valeurs affichées ?

Le code écrit le plus petit multiple de la valeur stockée en 201 supérieur ou égal à la valeur stockée en 200 et écrit le résultat en 202. Les valeurs affichées à l'exécution du programme sont les valeurs des registres (AX=10, BX=12, CX=3). La commande suivante affiche les paramètres d'appel de la fonction ([200]=10, [201]=3).

Exercice 2

On va écrire un programme qui va trier cinq nombres entiers. On suppose que les nombres à trier sont stockés en mémoire de l'adresse 200 à l'adresse 204 dans un ordre aléatoire au début de l'exécution. On souhaite qu'ils soient rangés, en ordre croissant, aux mêmes adresses à la fin.

1. Écrire un algorithme effectuant l'opération en pseudo-code, ou en C.

```
For A in ([200];[201];[202];[203];[204])
{
    For B in ([200];[201];[202];[203];[204])
    {
        If (A>B)
        {
            T=A;
            A=B;
            B=T;
        }
    }
}
```

2. Traduire l'algorithme de la question 1 en assembleur.

...

Exercice 3

On souhaite écrire un programme effectuant le calcul de A modulo M en ne servant que de l'addition. On supposera, à nouveau, que A est stocké à l'adresse 200 et M à l'adresse 201.

1. Comme dans l'exercice précédent, écrire un algorithme calculant le résultat.

```
A=[200];
M=[201];
V=0;
i=0;
If (A>=M)
{
    While (V<=A)
    {
        V+=M;
        i++;
    }
    V=0;
    For (j=1;j<i;j++)
    {
        V+=M;
    }
    For (R=0;V+R<A;R++)
    {
    }
}
Else
{
    R=A;
}
[202]=R;
```

2. Transcrire l'algorithme ci-dessus en assembleur et terminer le programme en écrivant le résultat de l'opération à l'adresse 202.

```
MOV AX,[200]
MOV BX,[201]
CMP AX,BX
JL L4
MOV CX,0
MOV DX,0
L1: ADD CX,BX
    INC DX
    CMP CX,AX
    JLE L1
    MOV AX,0
    MOV CX,0
L2: ADD CX,BX
    INC AX
    CMP AX,DX
    JL L2
MOV AX,[200]
MOV BX,0
```

```
        CMP CX,AX
        JGE L5
L3:     INC BX
        MOV DX,CX
        ADD DX,BX
        CMP DX,AX
        JLE L3
        JMP L5
L4:     MOV BX,AX
L5:     MOV [202],BX
        INT 3
```