**Projet TP ARCHI2**

Réalisé par :

Hiouani Lydia 202031060760 G3 Lydiahiouani@gmail.com

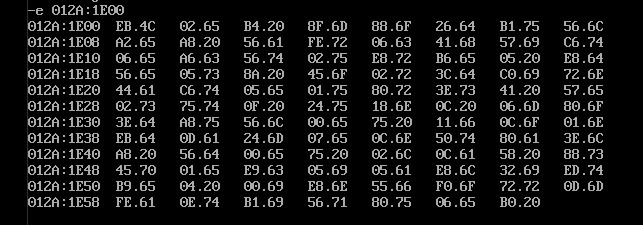
Lamri Mohamed Said Aimen 202031025221 G4 lamrimohamedsaidaimen@gmail.com

***Exercice01 :***

**1. Charger en mémoire la chaine de caractère suivante à partir de l’adresse 12Ah :1E00h**

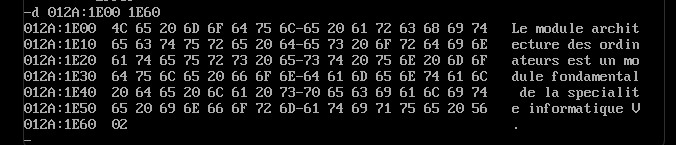
**‘Le module architecture des ordinateurs est un module fondamental de la spécialité informatique ‘**

en utilisant la commande e (enter) on introduit le code ASCII des caractères de la chaine dans la mémoire à partir de l’dresse 012A :1E00

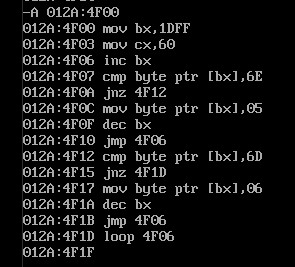


**-Vérifiez que votre commande a été correctement exécutée.**

En utilisant la commande d(dump) on visualise le contenu de la mémoire et on vérifie que la commande précédente a été exécutée correctement

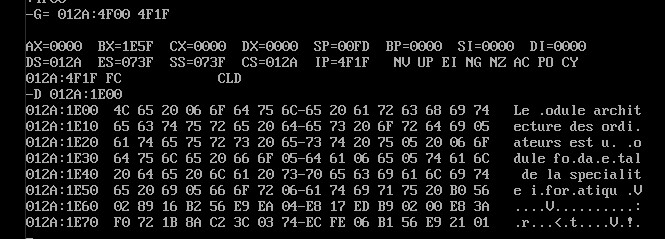


**2. Avec une séquence d’instructions optimale stockée à partir de l’adresse 12Ah: 4F00h , remplacez les caractères ‘n’ et ‘N’ par un trèfle et le ‘m’ et ‘M’ par un pique**

****

**- Vérifiez que le programme s’est bien exécutée.**

On utilise la commande d



*Remarque :*

Dosbox n’affiche pas les caractères spéciaux et donc on remarque des points et pas des trèfles et piques à la place de ‘m’ et ‘n’

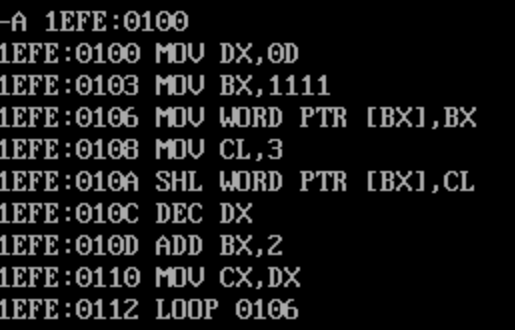
**3. Visualiser le code de la séquence d’instructions écrite dans la question 2**

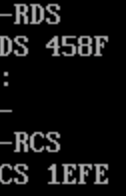
****

**4. Déduire la taille de la mémoire occupée par votre séquence d’instructions.**

Taille\_mem\_occup=@fin-@debut+1=20 octets

***Exercice02 :***

**1. Écrire en mémoire la séquence d’instructions**



**2. Que fait cette séquence ?**

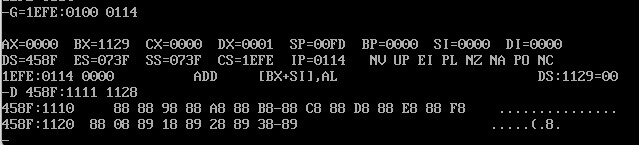
Cette séquence fait des multiplications de bx par 8 et incrémente bx de 2

**3. Visualisez le contenu de la structure ( juste la structure et pas une case**

**mémoire de plus) avant l’exécution de la séquence**



**4. Même question après exécution de la séquence tout en commentant le résultat obtenu**



Le résultat obtenu : êêÿë¿ê©ê╚êÏꝦê◦ê.ë↑ë(ë8ë

**5. Donnez l’état du processeur après la première itération de la boucle tout en commentant les bits d’état (flags)**



L’état des flags :

|  |  |
| --- | --- |
| OF | 0 |
| DF | 0 |
| IF | 1 |
| SF | 0 |
| ZF | 0 |
| AF | 0 |
| PF | 0 |
| CF | 0 |

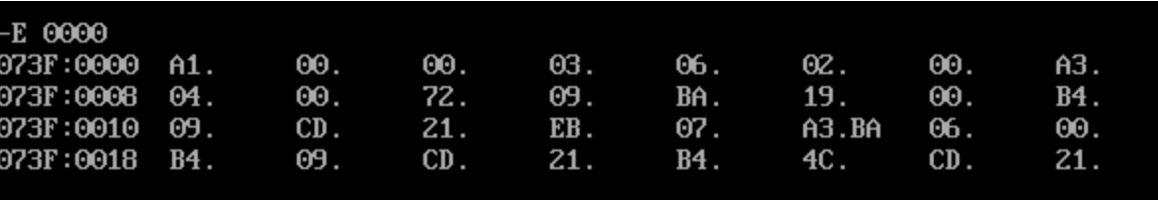
***Exercice03 :***

**1. Générez la séquence d’instructions correspondant au code machine.**

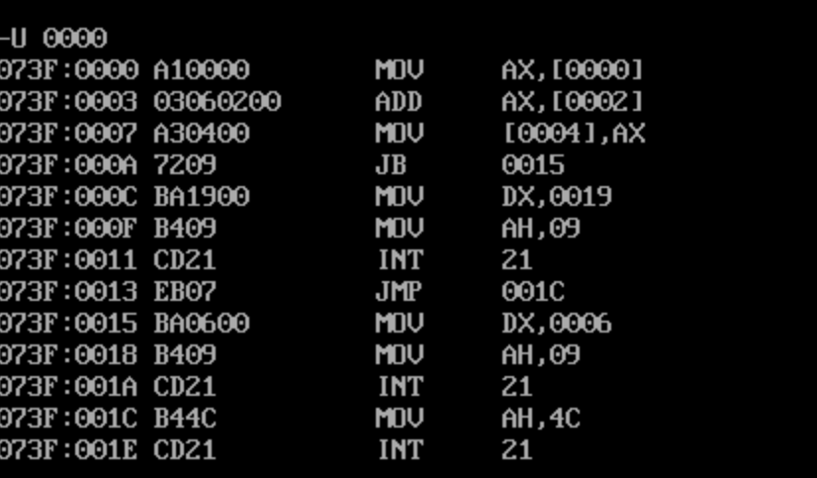
Génération de la séquence d’instructions suivantes :

A1 00 00 03 06 02 00 A3 04 00 72 09 BA 19 00 B4 09 CD 21 EB 07 BA 06 00 B4 09 CD 21 B4 4C CD 21

En utilisant la commande E on fait rentrée la séquence d’instructions précédentes octet par octet (chaque octet est représenté par deux numéros en hexadécimal) on tape à chaque foi espace pour faire entrer l’octet suivant comme suit :



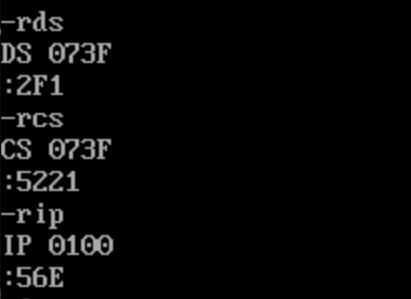
Pour générer la séquence d’instruction on utilise la commande U suivit de l’adresse ou est stocker la première instruction comme suit :



**2. Avant le lancement de l’exécution de la séquence il faut que :**

a. les registres suivants sont initialisés comme suit : DS=2F1h, IP=56EH, CS=5221H :

On initialise les registre DS,IP,CS avec la commande R + « le nom du registre », (R DS, R IP, R CS pour les trois registre) comme suit :

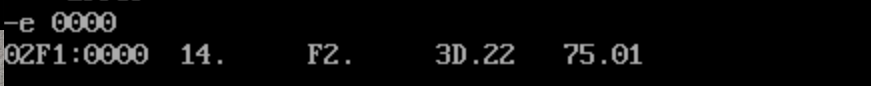


B. Les emplacements mémoire référencées en lecture dans la séquence d’instructions contiennent les valeurs suivantes F214H,0122h

On doit stocker ces valeur dans l’emplacement mémoire qui ont l’adresse 0000 ,0002 car les deux premières instructions du programme font un adressage direct dans le segment de donne vers les deux adresses 0000 et 0002

On utilise la commande E suivit de l’offset 0000 car l’adresse segment est déjà initialise a 02F1H

Comme suit :



**3. Vérifiez que vos commandes de la question 2 sont bien exécutées.**

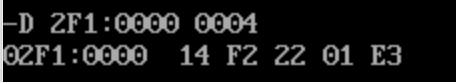
Pour vérifier que les registres ont bien était modifier on utilise la commande r comme suit :



On peut voir clairement que tous les registres sont modifier.

Pour vérifier que le contenu des deux emplacements mémoires ont bien était modifier on utilise la commande d suivit de l’adresse segment de donner et l’offset du début ainsi que la fin puisque on fait entrée 4 octet après l’offset du début 0000 donc l’offset de la fin est

1. me suit :

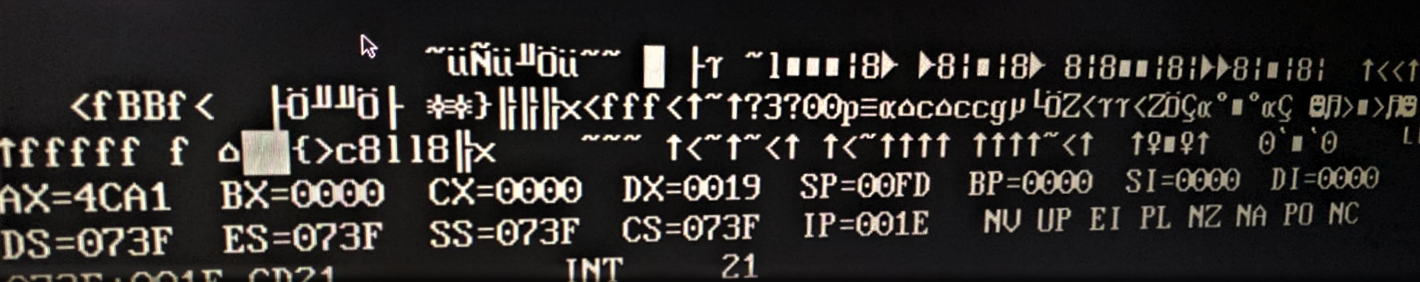


**4. Après exécution de la séquence**

**a. Visualisez les contenus des emplacements mémoires**

On exécute avec la commande G + l’adresse du début et l’offset de la fin du programme (l’offset de la dernière instruction) comme suit :





Remarque : LE PROGRAMME SE BLOQUE APRES L’EXECUTION EST ON PEUT PAS VISUALISER LE CONTENU DES EMPLACEMENTS MEMOIRES

**5. La séquence a-t-elle été exécutée correctement ? justifiez votre réponse**

Non la séquence n’a pas été exécutée correctement