INSTUTUT D’ELECTROTECHNIQUE

 Microprocesseur-8086

**Exercice N° 1 :**

Deux nombres N1 et N2 en mémoire, chacun sur quatre octets (N1=57F4F7E2h et N2=60DB513Ch) Ecrire un bout de programme qui réalise leur somme et stock le résultat en mémoire.

**Exercice N° 2 :**

Une table T de douze éléments ; chaque élément est sur un octet. Ecrire un bout de programme qui réalise la somme de tous les éléments et stock le résultat en mémoire.

*Remarque : Nous supposons que la valeur de la somme ne dépasse pas 255 ;*

**Exercice N° 3 :**

Une table T de douze éléments ; chaque élément est sur un octet. Ecrire un bout de programme qui trouve le grand élément et le stock en mémoire.

\*\*\***Exercice N° 4 :**

Une table T de douze éléments ; chaque élément est sur un octet. Ecrire un bout de programme qui trouve le plus petit élément et le stock en mémoire.

**Exercice N°5 :**

Une table T de douze éléments ; chaque élément est sur un octet. Ecrire un bout de programme qui remplace tout élément inferieur à 50h.

\*\*\***Exercice N°6 :**

Une table T de douze éléments ; chaque élément est sur un octet. Ecrire un bout de programme qui remplace tout élément supérieur à 50h.

**Exercice N°7 :**

Une table T de douze éléments organisée dans cet ordre

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 45h | 14h | 27h | 89h | 2h | 56h | 91h | 38h | 68h | 73h | 33h | 34h | 16h |

Ecrire un bout de programme qui la table T dans l’ordre inverse c'est-à-dire comme suit :

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16h | 34h | 33h | 73h | 68h | 38h | 91h | 56h | 2h | 89h | 27h | 14h | 45h |

**EXERCICE N°8:**

Ecrire un programme qui donne une table somme de deux tables T1 et T2

T3(i)= T1(i) + T2(i)

T1 DB 61H,96H,0F7H,43H,0D7H,75H,0B8H,8AH,5DH,1CH

T2 DB 84H,0E6H,29H,34H,98H,0C3H,0F5H,0CCH,4AH,0B2H

T3 DW 0AH PUR(0)

**EXERCICE N°9:**

Soit une table T :45H,14H,27H,89H,2H,56H,91H,38H,68h,73H,33h,34h,35h, de 13 éléments.

Dans un même programme organiser les éléments de cette table dans l’ordre croissant, transférer cette table vers la pile puis de la pile vers 2éme table T1.

|  |
| --- |
| **Exercice N° 1 :** |
| TITLE "Exercice N-1 ".MODEL SMALLSTACK SEGMENT  DB 64H PUR(0) STACK ENDS DATA SEGMENTN10 DW 0F7E2h; Partie basse de N1N11 DW 57F4h; Partie haute de N1N20 DW 513Ch; Partie basse de N2N21 DW 60DBh; Partie haute de N1S0 DW 1 PUR(0); Poids faible sommeS1 DW 1 PUR(0); Poids fort somme R DB 1 PUR(0) ; Retenue si elle existe DATA ENDSCODE SEGMENTASSUME CS:CODE,DS:DATA,SS:STACKSTART: .STARTUP ;MOV AX,DATA ;Initialisation de DS si on utilise pas STARUP. ;MOV DS,AX MOV AX,N10 ; ADD AX,[N20] MOV offset S0,AX  MOV AX,[N11] ADC AX,[N21] ; MOV offset S1,AX MOV DL,0 JC RETENUE MOV offset R,DL JMP FIN RETENUE: INC DL MOV [R],DLFIN:  hlt  CODE ENDS  END START |

|  |
| --- |
| **Exercice N° 2 :** |
| TITLE "Exercice N-2 ".MODEL SMALLSTACK SEGMENT  DB 64H PUR(0) STACK ENDS DATA SEGMENTT DB 45H,14H,27H,89H,2H,56H,91H,38H,68h,73H,33h,34h; S DB 1 PUR(0); La sommeDATA ENDSCODE SEGMENTASSUME CS:CODE,DS:DATA,SS:STACKSTART: .STARTUP mov cx,0Bh ;cx compteur d'éléments lea si,T ;offset de T dans le pointeur SI de la table mov ax ,0h ;par précaution mov al,[si] ;Transfert 1er élémentboucle: inc si ; add al,[si] ; loop boucle mov [S],al ; hlt CODE ENDS  END START |

|  |
| --- |
| **Exercice N° 3 :** |
| TITLE "Exercice N-3 ".MODEL SMALLSTACK SEGMENT  DB 64H PUR(0) STACK ENDS DATA SEGMENTT DB 45H,14H,27H,89H,2H,56H,91H,38H,68h,73H,33h,34h; G DB 1 PUR(0); La sommeDATA ENDSCODE SEGMENTASSUME CS:CODE,DS:DATA,SS:STACKSTART: .STARTUP mov cx,0Bh ;cx compteur d'éléments lea si,T ;offset de T dans le pointeur SI de la table mov ax ,0h ;par précaution mov al,[si] ; boucle: inc si ; cmp al,[si] ; jb max ; si al < [si] sauter à max pour charger le nouveau max loop boucle ; décrémenter cx et sauter vers l’adresse boucle jmp finmax: mov al,[si] loop bouclefin: mov [G],al ;sauvegarde du max dans la variable G  hlt CODE ENDS  END START |

|  |
| --- |
| **Exercice N° 5 :** |
| TITLE "Exercice N-5 ".MODEL SMALLSTACK SEGMENT  DB 64H PUR(0) STACK ENDS DATA SEGMENTT DB 45H,14H,27H,89H,2H,56H,91H,38H,68h,73H,33h,0D8h; DATA ENDSCODE SEGMENTASSUME CS:CODE,DS:DATA,SS:STACKSTART: .STARTUP mov cx,0Ch ;cx compteur d'éléments lea si,T ;offset de T dans le pointeur SI de la table mov ax ,0h ;par précaution mov al,50h ; boucle:  cmp [si],al ; jb zero inc si loop boucle jmp finzero: mov [si],0h inc si loop bouclefin:  hlt CODE ENDS  END START |

|  |
| --- |
| **Exercice N° 7 :** |
| TITLE "Exercice N-7 ".MODEL SMALLSTACK SEGMENT  DB 64H PUR(0) STACK ENDS DATA SEGMENTT DB 45H,14H,27H,89H,2H,56H,91H,38H,68h,73H,33h,34h,16h; DATA ENDSCODE SEGMENTASSUME CS:CODE,DS:DATA,SS:STACKSTART: .STARTUP mov bl,2 ;on divise la longueur de T par 2(cx divisé par 2) mov cx,0Ch ;cx compteur d'éléments lea si,T ;offset de T dans le pointeur SI de la table lea di,T ;DI pointe sur début de T add si,cx ;SI pointe sur la fin de T ; dec si ;// si le nombre de case est pair.  mov ax ,cx ; transférer cx vers ax pour le divisé par 2 div bl ;  mov cx,ax ; résultat de la division par 2 boucle:  mov al,[di] ; xchg al,[si] xchg [di],al dec si inc di loop boucle  hlt CODE ENDS  END STAR |

|  |
| --- |
| **Exercice N° 9:** |
| TITLE "Exercice N-9 "STACK SEGMENT  DB 64H PUR(0) STACK ENDS DATA SEGMENTT DB 45H,14H,27H,89H,2H,56H,91H,38H,68h,73H,33h,34h,35h,N EQU 13T1 DB 5 PUR(0) DATA ENDSCODE SEGMENTASSUME CS:CODE,DS:DATA,SS:STACKSTART:  MOV AX,DATA ;Initialisation de DS. MOV DS,AX  MOV AX,STACK ;Initialisation de SS. MOV SS,AX mov cx,64h  mov ax,0null:  push ax loop null  mov cx,N ;compteur d'élément de la table à mov ax,0 mov dl,0 ; mov si,offset T  mov di,offset T  mov al,[si]  boucle: call min  mov cx ,N;compteur sub cl,dl mov si,offset T  add si,dx mov di,si mov al,[di] mov ah,0  loop boucle inc si mov [si],0;prep transfert vers la pile  mov ax,N;.compteur mov dh,2  div dh mov cl,al add cl,1 mov dh,cl mov si,offset Tpile:  mov ax,[si] push ax inc si inc si loop pile mov di,offset T1 mov cl,dh consT1: pop ax mov [di],ax inc di inc di loop consT1 mov di,offset T1 mov cl,dhorgT1: mov ax,[di] xchg al,ah mov [di],ax  inc di inc di loop orgT1  hltmin:  inc si cmp al,[si] ja saut  loop min inc dl  retsaut: mov al,[si] mov ah,[di] mov [si],ah mov [di],al  loop min  inc dl ret  CODE ENDS ;  END START |