Instrucțiuni pentru șiruri. Probleme „complexe” pe șiruri

Instructiuni pentru siruri:

* Au operanzii impliciti si sunt folosite astfel:
  + Prelucreaza elementul curent din sir
  + Trece la urmatorul element
* Initializare:
  + Se copiaza offset-ul sirului sursa in registrul ESI
  + Se copiaza offset-ul sirului destinatie in registrul EDI
  + Se configureaza directia de parcurgere: CLD (DF=0, s=>d) sau STD (DF=1, d=>s)
* Clasificare:
  + Instructiuni pentru transferul datelor (B, W, D)
    - Muta datele din memorie in registrii (LODS – load string of ...)

LODSB

- AL <- [DS:ESI]  
- daca DF=0 atunci ESI=ESI+1 altfel ESI=ESI-1

LODSW

- AX <- [DS:ESI]  
- daca DF=0 atunci ESI=ESI+2 altfel ESI=ESI-2

LODSD

- EAX <- [DS:ESI]  
- daca DF=0 atunci ESI=ESI+4 altfel ESI=ESI-4

* + - Muta datele din registrii in memorie (STOS – store string of)

STOSB

- [ES:EDI] <- AL  
- daca DF=0 atunci EDI=EDI+1 altfel EDI=EDI-1

STOSW

- [ES:EDI] <- AX  
- daca DF=0 atunci EDI=EDI+2 altfel EDI=EDI-2

STOSD

- [ES:EDI] <- EAX  
- daca DF=0 atunci EDI=EDI+4 altfel EDI=EDI-4

* + - Muta date din memorie intr-o alta zona de memorie (MOVS – mov string of ...)

MOVSB

- [ES:EDI] <- byte [DS:ESI]  
- daca DF=0 atunci EDI=EDI+1; ESI=ESI+1 altfel EDI=EDI-1; ESI=ESI-1

MOVSW

- [ES:EDI] <- word[DS:ESI]  
- daca DF=0 atunci EDI=EDI+2; ESI=ESI+2 altfel EDI=EDI-2; ESI=ESI-2

MOVSD

- [ES:EDI] <- dword [DS:ESI]  
- daca DF=0 atunci EDI=EDI+4; ESI=ESI+4 altfel EDI=EDI-4; ESI=ESI-4

* + Instructiuni pentru comparararea datelor
    - Compara datele din registrii cu cele din memorie (SCAS – scan string of ...)

SCASB

- CMP AL, [ES:EDI]  
- daca DF=0 atunci EDI=EDI+1 altfel EDI=EDI-1

SCASW

- CMP AX, [ES:EDI]  
- daca DF=0 atunci EDI=EDI+2 altfel EDI=EDI-2

SCASD

- CMP EAX, [ES:EDI]  
- daca DF=0 atunci EDI=EDI+4 altfel EDI=EDI-4

* + - Compara datele din memorie cu cel din alta zona de memorie (CMPS – compare string of ...)

CMPSB

- CMP [DS:ESI], byte [ES:EDI]   
- daca DF=0 atunci EDI=EDI+1; ESI=ESI+1 altfel EDI=EDI-1; ESI=ESI-1

CMPSW

- CMP [DS:ESI], word[ES:EDI]   
- daca DF=0 atunci EDI=EDI+2; ESI=ESI+2 altfel EDI=EDI-2; ESI=ESI-2

CMPSD

- CMP [DS:ESI], dword [ES:EDI]   
- daca DF=0 atunci EDI=EDI+4; ESI=ESI+4 altfel EDI=EDI-4; ESI=ESI-4

Exercitii:

1. Se da un sir de octeti care contine litere mici. Sa se construiasca sirul de octeti care contine literele mari corespunzatoare literelor mici

|  |  |
| --- | --- |
| Data Segment | Code Segment |
| S db "asdf"  LEN equ ($-S)  D times Len db 0 | MOV ESI, S  MOV EDI, D  MOV ECX, LEN  CLD  JECXZ FINAL  REPETA:  LODSB ; AL = a ; ESI = ESI + 1  ADD AL, 'A' - 'a'  STOSB ; ES:EDI = AL EDI = EDI + 1  LOOP REPETA  FINAL: |
|  |  |

‘b’ + (‘A’ - ‘a’) = 1 + ‘A’ = ‘ B’

1. Se da un sir de octeti. Sa se obtina un nou sir de octeti = sirul sursa oglindit

A db 1,2,3,4,5 => b db 5,4,3,2,1

|  |  |
| --- | --- |
| Data Segment | Code Segment |
| S db 1, 2, 3, 4, 5  L equ $-S  D times L db 0 | Mov ecx, L  Jecx sfarsit  Lea esi, [S+L-1]  Lea edi, [D]  Et\_loop:  Std ; DF <- 1  Lodsb ; al <- [DS:ESI], dec ESI  CLD  STOSB  Loop Et\_loop  Sfarsit: |
|  |  |

1. Se dau 2 siruri de cuvinte. Sa se obtina sirul octetilor inferiori din primul sir concatenat cu sirul octetilor superiori din al doilea sir. Sa se sorteze crescator sirul rezultat in interpretarea cu semn

|  |  |
| --- | --- |
| Data Segment | Code Segment |
| S1 DW 4,5,3,8  L1 equ ($-S1)/2  S2 DW 1,2,3,4  L2 equ ($-S2)/2  D times L1+L2 DB 0 | Mov ecx, L1  MOV ESI, S1  MOV EDI, D  JECXZ sfarsit  CLD  REPETA:  LODSB  STOSB  LODSB  Loop REPETA  Sfarsit:  Mov ecx,L2  JEXCZ sfarsit2  MOV ESI, S2  CLD  REPETA2:  LODSB  LODSB  STOSB  Loop REPETA2  Sfarsit2:  Mov DL, 1  REPETA3:  Cmp DL,0  Jz Sfarsit3  Mov Dl, 0  Mov ECX, L1+L2-1  Mov ESI, D  REPETA4:  Mov AL, [ESI]  Mov BL, [ESI+1]  Cmp AL, BL  JLE Next  Mov [ESI], BL  Mov [ESI+1], AL  Mov DL, 1  Next:  Inc ESI  Loop REPETA4  Jump REPETA3  Sfarsit3: |

Ok=1

While(ok==1){

Ok=0;

For(I=1;I<=len-1;I++)

If(u[I+1]<u[I]){

Aux=u[I]

U[I]=u[I+1]

U[I+1]=aux

Ok=1

}

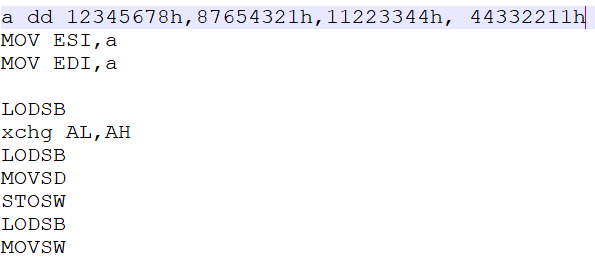
}

4. Se da un sir de numere reprezentate pe dublucuvinte. Sa se construiasca sirul octetilor negativi din reprezentarea fiecarui dublucuvant

|  |  |
| --- | --- |
| Data Segment | Code Segment |
| S DD 1,-4,7,-5,7  L equ $-S  D times L db 0 | Mov ecx, L  Mov esi,S  Mov edi,D  Jecxz sfarsit  CLD  Repeta:  Lodsb; Al=1  Cmp Al,0  Jge sari  Dec ESI  Movsb  Sari:  Loop repeta  Sfarsit: |

5. Se da un sir de dublu cuvinte. Sa se calculeze suma (octetilor low ai cuvintelor high) multiplii de 10.

|  |  |
| --- | --- |
| Data Segment | Code Segment |
| S dd 1, 10, 215, 2415  L EQU ($-S)/4 | MOV ESI, S  MOV EBX, 0 ; EBX = SUMA  CLD ; DF = 0  MOV ECX, L  JECXZ Sfarsit  Repeta:  LODSW  LODSB ; AL = octetul bun  MOV DL, AL ; DL = numar initial  CBW ; AX  IDIV BYTE 10 ; AL = catul, AH restul  CMP AH, 0  JNZ SARI\_PESTE  MOV AL, DL  CBW  CWDE ; EAX = NUMAR  ADD EBX, EAX  SARI\_PESTE:  LODSB  LOOP Repeta  Sfarsit: |
|  |  |



a dd 12345678h,87654321h,11223344h, 44332211h

MOV ESI,a

MOV EDI,a

LODSB; AL=78h; ESI=a+1

xchg AL,AH; AH=78h

LODSB; AL=56h; ESI=a+2 => AX=7856h

MOVSD; 34h|12h|21h|43h|21h|43h|65h|87h|...; ESI=a+6, EDI=a+4

STOSW; 34h|12h|21h|43h|56h|78h|65h|87h|...; EDI=a+6

LODSB; AL=65h; ESI=a+7

MOVSW; 34h|12h|21h|43h|56h|78h|87h|44h|44h|33h|22h|11h|11h|22h|33h|44h

init: 78h|56h|34h|12h|21h|43h|65h|87h|44h|33h|22h|11h|11h|22h|33h|44h