Neacșu-Miclea Liviu-Ștefan
Grupa 215

7. Se dă un șir de dublucuvinte. Să se obțină șirul format din octeții superiori ai cuvintelor superioare din elementele șirului de dublucuvinte care sunt divizibili cu 3.

Metoda 1 (first->last)

Data

```
S DD 12345678h, 1A2B3C4Dh, 0FE98DC76h,
     03111111h, 19010203h, 3FA0A0A0h
S_Len EQU ($-S)/4      ; = 6
D TIMES S_Len RESB 1
```

Output

```
D DB 12h, 03h, 3Fh
```

Code

```
mov ESI, S        ; ESI = S = source
mov EDI, D        ; ESI = D = destination
cld               ; DF = 0 (ESI/EDI+)
mov DL, 3         ; DL = 3

mov ECX, S_Len    ; ECX = S_Len = 6
iterate:
    lodsd         ; EAX = S[ESI] = X1_X2_X3_X4h, ESI+=4
    push CX       ; backup counter reg.
    mov CL, 8     ; CL = 8
    rol EAX, CL   ; EAX = X2_X3_X4_X1h
    pop CX        ; restore counter reg.
    mov BL, AL    ; BL = AL = X1h (backup AL)
    mov AH, 0     ; AH = 0 <=> AX = AL
    div DL        ; AL = AX/3, AH = AX%3

    or AH, AH     ; AH|=AH, ZF=1 if AH == 0; (cmp AH,0)
    loopnz iterate; jp if(!ZF) <=>if(X1%3!=0)continue;
                  ; || ECX==0
    ; (X1h % 3 == 0) :
    mov AL, BL    ; AL = BL = X1h (restore AL)
    stosb         ; D[EDI++] = AL
    loop iterate ; if (ECX>0) goto iterate
```

Neacșu-Miclea Liviu-Ștefan
Grupa 215

Metoda 2 (last->first)

Data

```
S DD 12345678h, 1A2B3C4Dh, 0FE98DC76h,
     03111111h, 19010203h, 3FA0A0A0h
S_Len EQU ($-S)/4     ; = 6
D TIMES S_Len RESB 1
D_Len RESB 1 ; actual dest length
```

Output

```
D DB 12h, 03h, 3Fh
```

Code

```
mov ESI, S+4*(S_Len-1) ; ESI = lst(source)
mov EDI, D             ; ESI = D = destination
mov DL , 3             ; DL = 3

mov ECX, S_Len     ; ECX = S_Len = 6
iterate:
    std            ; DF = 1 (ESI-)
    lodsd          ; EAX= S[ESI] = X1_X2_X3_X4h,ESI-=4
    push CX        ; backup counter
    mov CL, 8      ; CL = 8
    rol EAX, CL    ; EAX = X2_X3_X4_X1h
    pop CX         ; restore counter
    mov BL, AL     ; BL = AL = X1h (backup AL)
    mov AH, 0      ; AH = 0 <=> AX = AL
    div DL         ; AL = AX/3, AH = AX%3

    or AH, AH      ; AH|=AH, set ZF=1 if AH == 0
    loopnz iterate ; if(!ZF) <=>if(X1h%3!=0) continue;
                   ; || ECX==0
    ; (X1h % 3 == 0) :
    mov AL, BL     ; AL = BL = A1h (restore AL)
    cld            ; DF = 0 (EDI+)
    stosb          ; D[EDI++] = AL
    inc BYTE [D_Len] ; D_Len ++
    loop iterate ; if (ECX>0) goto iterate

xor ECX, ECX       ; ECX = 0
mov  CL, [D_Len]   ; CL = (ECX =) D_Len = 3

mov ESI, D         ; ESI = D
mov EDI, ESI       ; EDI = ESI = D
add ESI, ECX       ; ESI = D + D_Len
dec ESI            ; ESI = D + D_Len - 1 = lst(D)
shr ECX, 1         ; ECX = D_Len/2 = 1
cld                ; DF = 0 (ESI/EDI+)

reverseD:
    mov AL, [EDI] ; AL = *EDI := D[i], i=S_Len/2-ECX
    movsb         ; [EDI]=[ESI], EDI++, ESI++
    dec ESI       ; ESI -= 1, crt. elem.
    mov [ESI], AL ; [ESI] := D[D_Len-1-i] = AL
    dec ESI       ; ESI -= 1, prev. elem.
    loop reverseD ; if (ECX>0) goto reverseD
```

Neacșu-Miclea Liviu-Ștefan
Grupa 215