

```

;***** logovanje.asm *****
;** Opis programa: iz ulazne datoteke input.txt učitavaju se podaci o identifikacijama
;***** zaposlenih,
;** kako za uspešne identifikacije tako i za neuspešne (tj. greske).
;** Za racunanje ukupnog vremena koje je zaposleni proveo na poslu potrebno je izracunati
;** razliku izmedju prve i poslednje identifikacije koje imaju isti datum i sifru,
;** dok je za racunanje efektivnog vremena koje je zaposleni proveo na poslu potrebno
;** izracunati zbir svih medjusobnih razlika vremena izmedju odgovarajucih izlazaka i
;***** ulazaka.
;**
;** U input.txt fajlu je pored unete 6-cifrene sifre, zapisan i datum u formatu dd:mm:gg
;***** koji se takodje ispisuje u tabeli,
;** kao i vreme kada je izvršena identifikacija u formatu dd:mm:gg i hh:mm koje služi za
;***** obradu i racunanje ukupnog i efektivnog vremena.
;**
;** Format promenljivih:
;** Svi stringovi su BYTE, a pored njih je to i brojac (jer broji samo do 4).
;** Fajl hendleri su takodje WORD, kao i NewLine (konstanta za prelaz u novi red).
;*****

.model small
.stack 200h
.data
    TextF1 db ' sifra datum ul.vr iz.vr ef.vr uk.vr '
    TextF2 db 'I-----I-----I-----I-----I-----I-----I'
    NewLine dw 0a0dh ;novi red
    PogrBroj db 'Neparan broj logovanja!'
    PogrSifra db 'puta nepostojeca sifra!'
    PogrOtvaranje db 'Greska pri otvaranju ulaznog fajla input.txt!'
    PogrCitanje db 'Greska pri citanju ulaznog fajla!'
    FileIn db 'input.txt',0 ;ulazni fajl
    FileOut db 'output.txt',0 ;izlazni fajl
    FileHn dw ? ;hendlr za fajl
    FileHnI dw ? ;hendlr za input fajl
    FileHnO dw ? ;hendlr za output fajl
    sifra db 'xxxxxx' ; podatak za pogresnu sifru
    dccounter dw ? ; brojac
    petnaest db 15 ; konstante
    deset db 10
    sezdeset dw 60
    sezdeset2 db 60
    tridesetpet dw 35
    pomocni dw ? ; pomocni registar za aritmeticke operacije
    pomocni2 dw ? ; pomocni registar za aritmeticke operacije
    cifra db ? ;za dve tacke u izlaznom fajlu
    razmak db '$' ;za razmak u izlaznom fajlu
    dvetacke db ':$' ;za dve tacke u izlaznom fajlu
    brojstr db '000000 00:00:00 00:00 ' ;string formata 'xxxxxx dd:mm:gg hh:mm '
    TabelaZaposlenih db 1000 dup ('ssssss','dd:mm:yy','ul:vk','ul:vt','iz:vt','00:00','0')

.code
;***** Procedura: NapraviFajl *****
NapraviFajl proc near ;napravi fajl (ime je u dx)
    mov ah,3ch
    xor cx,cx ;setovanje atributa CX na NORMAL
    int 21h
    ret
NapraviFajl endp
;***** Procedura: NapraviFajl - kraj *****

;***** Procedura: OtvoriFajl *****
OtvoriFajl proc near ;otvori fajl za citanje ili upis (ime je u dx)
    mov ah,3dh ;otvori fajl
    mov al,2h ;setovanje atributa AL na READ/WRITE
    int 21h
    mov FileHn,ax ;sacuvaj fajl hendler
    ret
OtvoriFajl endp
;***** Procedura: OtvoriFajl - kraj *****

;***** Procedura: ZatvoriFajl *****
ZatvoriFajl proc near ;zatvori fajl (handle je u bx)
    mov ah,3eh
    int 21h

```

```

        ret
ZatvoriFajl endp
;***** Procedura: ZatvoriFajl - kraj *****

;***** Procedura: FormirajOkvir *****
FormirajOkvir proc near
        ;upisi zaglavlje u izlazni fajl (output.txt)
        mov ah,40h                ;pisanje u fajl
        mov bx,FileHno            ; FileHno se prepisuje u bx

        mov cx,41                 ;duzina teksta se upisuje u cx
        lea dx,textF1             ;pisi prvu liniju tabele
        int 21h

        mov ah,40h
        mov cx,2
        lea dx,NewLine            ;novi red
        int 21h

        mov ah,40h                ;pisanje u fajl
        mov cx,41                 ;duzina teksta se upisuje u cx
        lea dx,textF2             ;drugi red tabele
        int 21h

        mov ah,40h                ;pisanje u fajl
        mov cx,2                 ;duzina teksta
        lea dx,NewLine            ;novi red
        int 21h

        ret
FormirajOkvir endp
;***** Procedura: FormirajOkvir - kraj *****

;***** Procedura: Konverzija1 *****
; ** Konverzija hh:mm u ceo broj minuta nad kojim mogu da se izvrsavaju aritmeticke
; operacije.
; ** U registru SI se nalazi offset stringa hh:mm
;*****
Konverzija1 proc near
        xor dx, dx
        mov al, [si]
        sub al, '0'
        inc si
        xor ah, ah
        mul deset
        add dx, ax
        mov al, [si]
        sub al, '0'
        add si, 2
        xor ah, ah
        add dx, ax ; broj sati se nalazi u DX
        mov ax, dx
        mul sezdeset2
        mov dx, ax ; broj sati pretvoren u minute se nalazi u DX, sada treba dodati i
minute
        mov al, [si]
        sub al, '0'
        inc si
        xor ah, ah
        mul deset
        add dx, ax
        mov al, [si]
        sub al, '0'
        inc si
        xor ah, ah
        add dx, ax ; e sad je u DX ukupan broj minuta, izracunato iz hh:mm
        ret
Konverzija1 endp
;***** Procedura: Konverzija1 - kraj *****

;***** Procedura: Konverzija2 *****
; ** Konverzija celog broja minuta u hh:mm i upis u efektivno vreme u memoriji.
; ** U registru SI se nalazi offset podatka u kome se nalazi broj minuta.
;*****
Konverzija2 proc near

```

```

    xor     dx,dx
    mov     ax,[si]
    div     sezdeset      ; u AX se nalazi broj sati, u DX broj minuta
    div     deset         ; u AL se nalazi visa, u AH visa cifra

    mov     TabelaZaposlenih[bx+29],al
    add     TabelaZaposlenih[bx+29],48
    mov     TabelaZaposlenih[bx+30],ah
    add     TabelaZaposlenih[bx+30],48
    mov     ax, dx
    div     deset

    mov     TabelaZaposlenih[bx+32],al
    add     TabelaZaposlenih[bx+32],48
    mov     TabelaZaposlenih[bx+33],ah
    add     TabelaZaposlenih[bx+33],48
    ret

Konverzija2 endp
;***** Procedura: Konverzija2 - kraj *****
;***** Procedura: Konverzija3 *****
;** Konverzija celog broja minuta u hh:mm i upis u ukupno vreme u izlaznom fajlu
;** U registru SI se nalazi offset podatka u kome se nalazi broj minuta.
;*****
Konverzija3 proc     near

    lea     si, TabelaZaposlenih[di+24]                                ; konvertuje trenutno hh:mm ↩
    izlazno vreme u minute i stavlja u pomocni2
    call    Konverzija1
    mov     pomocni2,dx

    lea     si, TabelaZaposlenih[di+14]                                ; konvertuje trenutno hh:mm ↩
    ulazno vreme u minute i oduzima to od pomocni2
    call    Konverzija1
    sub     pomocni2, dx

    push    bx
    mov     bx, FileHnO

    xor     dx, dx
    mov     ax, pomocni2

    div     sezdeset      ; u AX se nalazi broj sati, u DX broj minuta
    div     deset         ; u AL se nalazi visa, u AH niza cifra sati
    push    dx
    push    ax

    mov     cifra,al
    add     cifra,48

    mov     ah, 40h
    lea     dx, cifra
    mov     cx,1
    int     21h

    pop     ax
    mov     cifra,ah
    add     cifra,48

    mov     ah, 40h
    lea     dx, cifra
    mov     cx,1
    int     21h

    mov     ah, 40h
    lea     dx, dvetacke
    mov     cx,1
    int     21h

    pop     dx
    mov     ax, dx          ; u DX se nalazio ukupan broj minuta
    div     deset          ; u AL se nalazi visa, u AH niza cifra minuta

    push    ax
    mov     cifra,al

```

```

    add cifra,48
    mov ah, 40h
    lea dx, cifra
    mov cx,1
    int 21h

    pop ax
    mov cifra,ah
    add cifra,48
    mov ah, 40h
    lea dx, cifra
    mov cx,1
    int 21h

    pop bx
    ret
Konverzija3 endp
;***** Procedura: Konverzija3 - kraj *****

;***** Procedura: Racunanje *****
; Obrada podataka iz brojstr. Ako se kombinacija sifre i datuma poklapa sa sifrom i datumom
; nekog vec obradjenog podatka, treba azurirati red u memoriji.
; Ako nije prepoznat, treba formirati novi red sa novom sifrom i datumom.
;*****
Racunanje    proc    near

    mov ax, dccounter    ; broj redova u memoriji se mnozi sa 35 da bi se uporedio sa
    pointerom na bajtove
    mul tridesetpet
    mov pomocni, ax
    mov bx, 0            ; bx sluzi kao pointer na bajtove, tj. bx=0 znaci da smo u
    prvom redu, bx=35 u drugom itd.

    nalazenjes:lea di, brojstr    ; proverava se da li sifra vec postoji u
    rekordima
    lea si, TabelaZaposlenih[bx]
    mov cx, 6
    cld
    repe cmpsb
    je nalazenjed    ;ako ima sifre, proverava se datum uz tu sifru
    cmp bx, pomocni
    je novi    ; ako ne postoji red koji ima identicnu sifru a
    stiglo se do kraja tabele pravimo novi red
    add bx, tridesetpet    ; pomeranje na sledeci red u tabeli
    jmp nalazenjes
    nalazenjed: lea di, brojstr[7]
    lea si, TabelaZaposlenih[bx+6]    ; proverava se da li kombinacija sifre i datuma
    vec postoji u redovima
    mov cx, 8
    cld
    repe cmpsb
    je postojeci    ; ako vec postoji red koji ima identican datum i
    sifru, azuriraju se polja u tom redu
    cmp bx, pomocni
    je novi    ; ako ne postoji red koji ima identicnu sifru a
    stiglo se do kraja tabele pravimo novi red
    add bx, tridesetpet    ; pomeranje na sledeci red u tabeli
    jmp nalazenjes

    postojeci: test TabelaZaposlenih[bx+34],1 ; da li je poslednje bilo ulaz ili izlaz?
    jnz izlaz

    ulaz:
    cld
    mov cx, 5
    lea si, brojstr[16]    ; upisuje trenutno ulazno vreme vreme u
    novi red tabele i inkrementira broj ulazaka/izlazaka
    lea di, TabelaZaposlenih[bx+19]
    rep movsb
    inc TabelaZaposlenih[bx+34]
    jmp kraj2

    izlaz:cld
    mov cx, 5
    lea si, brojstr[16]    ; upisuje trenutno izlazno vreme vreme u

```

```

novi red tabele i inkrementira broj ulazaka/izlazaka
    lea di, TabelaZaposlenih[bx+24]
    rep movsb
    inc TabelaZaposlenih[bx+34]

    lea si, TabelaZaposlenih[bx+24]           ; konvertuje trenutno hh:mm izlazno vreme u
    minute i stavlja u pomocni
    call Konverzija1
    mov pomocni,dx

    lea si, TabelaZaposlenih[bx+19]         ; konvertuje trenutno hh:mm ulazno vreme u
    minute i oduzima to od pomocnog
    call Konverzija1
    sub pomocni, dx

    lea si, TabelaZaposlenih[bx+29]         ; konvertuje trenutno hh:mm efektivno vreme u
    minute i sabira sa pomocnim
    call Konverzija1
    add pomocni,dx

    lea si,pomocni                          ; konvertuje novo efektivno vreme u minutima u
    format hh:mm
    call Konverzija2

    jmp kraj2

novi: inc dccounter
    cld
    mov cx, 6
    lea si, brojstr                         ; upisuje sifru u novi red tabele
    lea di, TabelaZaposlenih[bx]
    rep movsb

    cld
    mov cx, 8
    lea si, brojstr[7]                     ; upisuje datum u novi red tabele
    lea di, TabelaZaposlenih[bx+6]
    rep movsb

    cld
    mov cx, 5
    lea si, brojstr[16]                     ; upisuje konacno ulazno vreme vreme u
    novi red tabele
    lea di, TabelaZaposlenih[bx+14]
    rep movsb

    cld
    mov cx, 5
    lea si, brojstr[16]                     ; upisuje trenutno ulazno vreme vreme u
    novi red tabele
    lea di, TabelaZaposlenih[bx+19]
    rep movsb

    ; izlazno vreme i efektivno vreme ostaju
    prazni, tj default vrednosti
    inc TabelaZaposlenih[bx+34]             ; inkrementira broj ulazaka/izlazaka

kraj2: ret

Racunanje    endp
;*****      Procedura: Racunanje - kraj *****

;*****      Procedura: IspisFajla      *****
; ** Uzima redove i ispisuje podatke u fajl
;*****
IspisFajla proc    near    ; upisi rezultate u izlazni fajl (output.txt)
    mov ax, dccounter    ; broj elemenata u nizu se mnozi sa 15 da bi se uporedio sa
    internim brojacem
    sub ax,1
    mul tridesetpet
    mov pomocni, ax
    mov di,0
ispis:
    mov bx, FileHn0

```

```
    lea     dx,razmak      ;razmak
    mov     ah,40h
    mov     cx,1
    int     21h

    lea     dx,TabelaZaposlenih[di] ;sifra
    mov     ah,40h
    mov     cx,6
    int     21h

    lea     dx,razmak      ;razmak
    mov     ah,40h
    mov     cx,1
    int     21h

    lea     dx,TabelaZaposlenih[di+6]      ;datum
    mov     ah,40h
    mov     cx,8
    int     21h

    lea     dx,razmak      ;razmak
    mov     ah,40h
    mov     cx,1
    int     21h
    push    di
    lea     di,TabelaZaposlenih[di]
    lea     si,sifra
    mov     cx,6
    cld
    repe    cmpsb
    pop     di
    je      losa_sifra

    test    TabelaZaposlenih[di+34],1
    jnz     greska1
    lea     dx,TabelaZaposlenih[di+14]      ;ul vreme konacno
    mov     ah,40h
    mov     cx,5
    int     21h

    lea     dx,razmak      ;razmak
    mov     ah,40h
    mov     cx,1
    int     21h

    lea     dx,TabelaZaposlenih[di+24]      ;izl vreme trenutno
    mov     ah,40h
    mov     cx,5
    int     21h

    lea     dx,razmak      ;razmak
    mov     ah,40h
    mov     cx,1
    int     21h

    lea     dx,TabelaZaposlenih[di+29]      ; ef.vreme
    mov     ah,40h
    mov     cx,5
    int     21h

    lea     dx,razmak      ;razmak
    mov     ah,40h
    mov     cx,1
    int     21h

    push    di
    call    Konverzija3      ; ukupno vreme
    pop     di

    mov     ah,40h      ;novi red
    mov     cx,2
```

```

        lea dx,NewLine
        int 21h

        cmp di, pomocni
        je povratak                ; ako se stiglo do kraja tabele, tj nema vise
redova
        add di, tridesetpet        ; pomeranje na sledeci red u tabeli
        jmp ispis

losa_sifra: mov ah, 40h
lea dx, TabelaZaposlenih[di+34]
mov cx,1
int 21h
mov ah,40h
        lea dx,PogrSifra          ; poruka o gresci
        mov cx,23
        int 21h
        jmp novired
greskal:                                ;ispisivanje poruke ukoliko imamo neparan broj logovanja
                                        ; Moze biti do toga da je zaposleni jos uvek u zgradi
        u vreme obrade podataka,      ; ili radi nocu pa je usao jednog dana a izasao drugog
        ,                             ; ili prosto nije doslo do upisa neki put kada je usao

        ili izasao
        mov ah,40h
        lea dx,PogrBroj          ; poruka o gresci
        mov cx,23
        int 21h
novired:      mov ah,40h
        mov cx,2
        lea dx,NewLine ;novi red
        int 21h

        cmp di, pomocni          ; da li se stiglo do kraja tabele
        je povratak
        add di, tridesetpet      ; pomeranje na sledeci red u tabeli
        jmp ispis
povratak:  ret
IspisFajla endp
;***** Procedura: IspisFajla - kraj *****

;***** Procedura: CitajFajl *****
;** Cita kompleksne podatke (svaki podatak ima 23 znaka) iz fajla ciji je hendler u
promenljivoj FileHnI sve dok ne dodje
;** do kraja fajla. Svaki string od 23 znakova (6 za sifru, 1 blanko,
;** 8 za datum dd:mm:gg, 1 blanko, 5 za vreme hh:mm, i 2 za prelazak u novi red) se
prosledjuje proceduri
;** Racunanje koja vrsi obradu stringova i azuriranje rekorda.
;*****
CitajFajl proc near
        mov dccounter,0 ; iniciranje brojaca za niz u koji se smestaju stringovi
read: mov ah,3fh ;citaj fajl
        mov bx,FileHnI
        mov cx,23 ;citaj 23 znaka (6 cifara za sifru, 1 blanko, 8 za datum, 1
blanko, 5 za vreme i 2 za novi red)
        lea dx,brojstr ; bafer za smestaj podatka
        int 21h ; procitani sadrzaj od 23 znaka se nalazi u promenljivoj
brojstr

        cmp brojstr[22], 0Ah ;da li je poslednji bajt u brojstr kod za novi red? ozbiljna
greska jer remeti strukturu ulaznih podataka
        jz ok2 ; moglo i da se desi da neki bajt fali pa bi odatle nadalje
pogresno interpretirao sve
        MOV BX, FileHnO ;ispisuje poruku o gresci
        MOV AH, 40h
        MOV DX, offset PogrCitanje
        MOV CX,33
        INT 21h
        mov ah,40h
        mov cx,2
        lea dx,NewLine ;novi red
        int 21h

```

```

MOV AX, 4c00h
INT 21h

ok2:    cmp ax,0           ;da li je stigao do kraja fajla pa je procitao novi red?
        jz     eof         ;ako ne, kraj citanja

        call  Racunanje    ;obradi tekuci podatak od 23 znaka i azuriraj rekorde
        jmp   read

eof:     ret               ; povratak u main, gde se ispisuje u fajl iz rekorda
CitajFajl endp
;*****
;***** Main *****

Main:
    mov ax,@data          ;inicijalizacija data segmenta
    mov ds,ax
    mov es,ax

    lea si,brojstr
    lea dx,FileOut

    call NapraviFajl      ;napravi output.txt fajl
    call OtvoriFajl       ;otvori ga
    mov FileHnO,ax        ; u FileHnO cuva se fajl hendler izlaznog fajla.
    lea dx,FileIn
    call OtvoriFajl       ;otvara input.txt
    JNC ok

    MOV BX, FileHnO        ; ukoliko nema fajla input.txt, u output.txt se ispisuje
    poruka o gresci!
    MOV AH, 40h
    MOV DX, offset PogrOtvaranje
    MOV CX,45
    INT 21h
    MOV AX, 4c00h
    INT 21h

ok: mov FileHnI,ax        ; u FileHnI cuva se fajl hendler ulaznog fajla.
    call CitajFajl        ;cita fajl red po red i smesta podatke u memoriju
    call FormirajOkvir    ;crta zaglavlje tabele u output.txt
    call IspisFajla       ;ispisuje red po red podatke u output.txt
    mov bx,FileHnO
    call ZatvoriFajl      ;zatvara output.txt
    mov bx,FileHnI
    call ZatvoriFajl      ;zatvara input.txt

Kraj:  mov ax,4c00h       ;kraj programa
        int 21h

end Main

```