713_1 Problema 2 Postescu Răzvan-Ioan

main.asm

```
bits 32

global start

extern exit, scanf, printf, patrate_linie, patrate_coloane
import exit msvcrt.dll
import scanf msvcrt.dll
import print msvcrt.dll


segment data use32 class=data
    n resd 1                          | linie_format db "
    read_format db "%u",              | Bitul de pe. de pe line
    matrice resb 256                  | %u = "%u", 0
    temp_d resd 1                     | coloana_format db "_u_"
    sqrt_n resd 1
segment code use32 class=code
    push dword n
    push dword read_format
    call [scanf] ; citire
    add esp, 4*2 ; golire stivă

    ; verificăm dacă e pătrat perfect calculând
toate pătratele perfecte de la 16 la 1 32
comparându-le cu n

    mov ecx, 16
    pp_loop:
```

1/

```
mov eax, ecx ; EAX = 16, 15, 14..., 1
mul ecx ; 256, 225...
cmp eax, [n] ; comparăm toate pp cu [n]
je este_pp
loop pp_loop
; dacă nu e pp terminăm programul
jmp gata
este_pp:
; citim matricea
mov ecx, [n]
jecx gata
citire_loop:
    push ecx ; salvăm ecx
    push dword temp_d
    push dword read_format
    call [scanf] ; citire elemente matrice
    add esp, 4*2

    mov ebx, [temp_d] ; salvăm elem citit în ebx
    mov eax, [n] ; EAX = [n]
    sub eax, ecx ; metodă de indexare pt elem.
    mov [matrice + eax], bl
    pop ecx
loop citire_loop
```

FI3-1 Problema 2 Postescu Răzvan-Ioan

macro.asm

```
mov ecx, [sqrt_n]
linii_loop:
    push ecx
    mov ebx, [sqrt_n]
    sub ebx, ecx
    mov eax, [sqrt_n]
    mul ebx
    add eax, matrice

    push dword [sqrt_n]
    push dword eax
    call patrate_linie
    add esp, 4*2

    push dword eax
    push dword inc ebx
    push dwor ebx
    push dwor linie_format
    call [printf]
    add esp, 4*3
    pop ecx
loop linii_loop
mov ecx, [sqrt_n]
```

coloane_loop:

; la fel ca pentru linii, modulul
primeste ~~coloane~~ pentru coloane ca param.
adresa de inceput a coloanei si lungimea
ei
; si returneaza in eax bitul de paritate
; va fi afisat la fel ca cel de pe linie
;dar folosind coloana_format

gata:
push dword 0
call [exit]

FTS_1 **Problema 2** Postescu Răzvan-Ioan

```
module2.asm

bits 32

global paritate_linie

segment code use 32 class=data


segment code use 32 class=code
  paritate_linie:
      ; bitul de paritate poate fi calculat
  prin efectuarea operației XOR asupra tuturor
  elementelor de pe o linie
   mov ESI, [ESP+4*1] ; adresa începutului liniei
   mov ECX, [ESP+4*2] ; lungimea liniei
   jecxz done   ; dacă se dau argumente rele funcția
                  se termină

  mov dl, [esi]
  inc esi
  mov eax, 0
  main_loop:
      lodsb
      xor al, dl
  loop main_loop ; în eax va fi 0 sau 1, bitul de paritate
```

+13 - 1 Problema 2 Posteucu Răzvan-Ioan

modul 2 . asm

bits 32

global paritate_coloana

segment data use32 class=data
    lung_col resd 1

segment code use 32 class=code
    ; bitul de par => XOR pe toti elem de
    coloana
    mov esi , [ESP+4*1]; adresa
    mov ECX, [ESP +4*2]; lungime
    mov [lung_col], ecx
    jecxz done;
    mov dl, [esi]
    add esi , ecx
    mov eax, 0
    main_loop:
        mov al, [esi]
        add ex , [lung_col]
        xor al , dl
    loop main_loop.