POP MIHAI - DANIEL

EXAMEN - PARTEA 2

a) Procedura "changeChar" primește la intrare cu ajutorul stivei 3 date: un număr natural "n" la adresa [ESP+4], un caracter "op" la adresa [ESP+8] și un caracter "c" la adresa [ESP+12]. Scoate de pe stivă aceste valori și le pune în regiștrii, cu ajutorul cărora putem ~~opera~~ efectua operații de comparație, adunare și scădere.

```
procedură.asm
bits 32
global changeChar
segment data use32 class=data public
    c db 0
    op db 0
    n db 0

segment code use32 class=code public
 changeChar:
    mov ebx, [ESP+4]
    mov ecx, [ESP+8]
    mov edx, [ESP+12]
    mov byte [n], bl
    mov byte [op], cl
    mov byte [c], dl
    cmp cl, '-'
    jne next
```

b) Pentru cele 2 șiruri date în segmentul de date, le parcurgem simultan și formăm un al treilea conform cerinței date, byte cu byte. În timpul parcurgerii, comparăm octetul curent să fie diferit de caracterul '$', astfel, dacă e diferit apelăm procedura "changeChar". La final, după ce a fost creat al treilea șir, îl mai parcurgem încă o dată și cu ajutorul unui maxim aflăm nr. de apariții maxime al unui caracter. După se afișează pe ecran acel caracter și de câte ori apare.

```
main.asm
bits 32
global start
extern exit, printf, changeChar
import exit msvcrt.dll
import printf msvcrt.dll

segment data use32 class=data public
    s1 db 'ana,are,mere,eu,am,mai,am', 0
    s2 db '---,$--,$---,--,$-,$$$,--', 0
                          ← len equ $-s1
    format_c db "%c", 0
    format_d db "%d", 0
    maxim db 0
    ~~s3 db~~
    s3 times len db 0
```

```
segment code use 32 class=code public
  start:
    mov esi, s1
    mov edi, s3
    mov ecx, len
    mov ebx, 0
    jecxz final
    repetă:
      lodsb
      cmp al, ','
      je virgulă
      ~~push~~
      mov dl, al
      mov eax, 0
      mov al, dl
      push eax
      mov al, [s2+EBX]
      push eax
      ~~mov al,~~
      push ebx
      call changeChar          ; din procedură revine cu
      add esp, 4*3             rezultatul salvat în eax
      virgulă:
      stosb
    loop repetă

    :
    :

push dword [maxim]
push dword format d
call [printf]
add esp, 8
final:
push dword 0
call [exit]
```

3