Probleme recapitulative

1. Se citesc de la tastatura nume de fisiere pana se va introduce cuvantul “END”. Sa se gaseasca fisierul cu numar maxim de litere mari si sa se afiseze pe ecran numele si continutul lui

OBS:

* Fisierele pot contine oricate caractere
* Sa se scrie o procedura care calculeaza numarul de litere mari dintr-un fisier dat ca parametru
* Sa se scrie o procedura care primeste ca parametru un nume de fisier si afiseaza pe ecran numele si apoi continul fisierului pe linie noua.

|  |
| --- |
| Main.asm |
| Bits 32    Global start        Extern exit, scanf, nr\_litere\_mari, afisare\_continut    Import exit msvcrt.dll    Import scanf msvcrt.dll        Segment data use32 class=data public    nume\_fisier times 101 db 0    citire db "%s", 0    final db "END", 0    maxim\_litere dd 0    maxim\_continut times 101 db 0        Segment code use32 class=code public    start:    cld    repeta:    Push dword nume\_fisier    Push dword citire    Call [scanf]    Add esp, 4 \* 2        Mov esi, nume\_fisier    Mov edi, final        Verificare\_final:    Cmp byte [esi], 0    Je sfarsit\_sir1    Cmpsb    Jne mai\_departe    Je Verificare\_final        sfarsit\_sir1:    Cmp byte [edi], 0    Je iesire        mai\_departe:        Push dword nume\_fisier    Call nr\_litere\_mari    Add esp, 4 \* 1        Cmp dword [maxim\_litere], eax    Ja repeta        Mov [maxim\_litere], eax    Mov esi, nume\_fisier    Mov edi, maxim\_continut        copiere:        Movsb    Cmp byte [esi], 0    Jne copiere    Movsb        Jmp repeta        iesire:        Push dword maxim\_continut    Call afisare\_continut    Add esp, 4 \* 1        Push dword 0    Call [exit] |
| Functie1.asm |
| Bits 32    Global nr\_litere\_mari    Extern fopen, fclose , fread    Import fopen msvcrt.dll  Import fread msvcrt.dll    Import fclose msccrt.dll        Segment data use32 class=data public    mod\_acces db "r", 0    descriptor dd 0    cch times 100 db 0        Segment code use32 class=code public    nr\_litere\_mari:    ; [esp] - adresa de revenire    ; [esp+4] - adresa nume\_fisier    Mov bx, 0  mov eax,[esp+4]    Push dword mod\_acces    Push dword eax    Call [fopen]  add esp, 4\*2        Cmp eax, 0    Je final        Mov [descriptor], eax            Citire:    Push dword [descriptor]    Push dword 1    Push dword 1    Push dword cch    Call [fread]    Add esp, 4\*4        Cmp eax, 0    Je gata            Cmp byte [cch], 'A'    Jl next    Cmp byte [cch], 'Z'    Jg next        Inc bx            next:    Jmp Citire        gata:    Push dword [descriptor]    Call [fclose]    Add esp, 4        final:    Mov eax,0    Mov ax, bx        ret |
| Functie2.asm |
| ; afisare\_continut  Bits 32  Global afisare\_continut  Extern printf, fopen, fclose, fread  Import printf msvcrt.dll  Import fopen msvcrt.dll  Import fclose msvcrt.dll  Import fread msvcrt.dll  Segment data use32 public data  format db "%s", 10, 0  mod\_acces db "w", 0  descriptor dd –1  format\_afsiare db "%c", 0  ch db 0  Segment code use32 public code  afisare\_continut:  Mov eax, [esp+4]  Push dword eax  Push format  Call [printf]  Add esp, 4 \* 2  Mov eax, [esp+4]  Push mod\_acces  Push eax  Call [fopen]  Add esp, 4 \* 2  Mov [descriptor], eax  Cmp eax, 0  Je final:  Mov eax, [esp + 4]  repeta:  Push dword [descriptor]  Push dword 1  Push dword 1  Push dword ch  Call [fread]  Add esp, 4 \* 4  Cmp eax ,0  Je inchidere\_fisier  Mov ebx, 0  Mov bl, [ch]  Push ebx  Push format\_afisare  Call [printf]  Add esp, 4 \* 2  Jmp repeta  Inchidere\_fisier:  Push dword [descriptor]  Call [fclose]  Add esp, 4  Final:  ret |
|  |

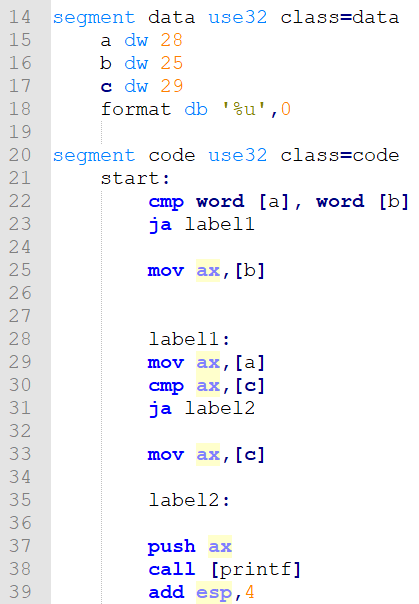
1. Se citeste de la tastatura numele unui fisier si apoi se citesc din fisier toate numerele existente reprezentate pe dublucuvant. Sa se afiseze pe ecran numerele ai caror octeti low din cuvintele high sunt primi.

|  |
| --- |
| Main.asm |
| Bits 32  Global start  Extern exit ,scanf,prim ,fopen,fscanf,fclose  Import exit msvcrt.dll  Import scanf msvcrt.dll  Import fopen msvcrt.dll  Import fscanf msvcrt.dll  Import fclose msvcrt.dll  Segment data use32 class=data  nume\_fisier times 20 db 0  format db “%d”,0  Format3 db “%s”,0  Format2 db “%d”,10,13,0  mod\_acces db “r”,0  descriptor dd 0  Numar dd 0  Segment code use32 class=code  start:  Push dword [nume\_fisier]  Push dword format3  Call [scanf]  Add esp,4\*2  Push dword mod\_acces  Push dword nume\_fisier  Call [fopen]  Add esp,4\*2  Cmp eax,0  Je final  Mov [descriptor],eax  Bucla:  Push dword Numar  Push dword format  Push dword [descriptor]  Call [fscanf]  Add esp,4\*3  Cmp eax,0  Jle inchidere\_fisier  Mov ebx,[Numar]  Shr ebx,16  Mov bh,0  Push dword ebx  Call prim  Add esp,4  Cmp eax,0  Jne Sari  Push dword [Numar]  Push dword format2  Call [printf]  Add esp,4\*2  Sari:  Jmp bucla  Inchidere\_fisier:  Push dword [descriptor]  Call [fclose]  Add esp,4  Final:  Push dword 0  Call [exit] |
| Modul.asm |
|  |

int fscanf( FILE \*stream, const char \*format [, argument ]... );

<https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/crt-alphabetical-function-reference?view=msvc-160&viewFallbackFrom=vs-2017>

Ce modificari trebuie facute programului astfel incat acesta sa afiseze corect pe ecran cea mai mare valoare fara semn dintre a,b,c indiferent de valorile din a,b,c?



A. linia 18 trebuie inlocuita cu: format db '%d',0

B. linia 29 trebuie stearsa

C. pe linie 37 trebuie push eax.

D. linia 22 trebuie inlocuita cu:

mov eax, [a]

cmp ax, [b]

E. linia 22 trebuie inlocuita cu:

mov eax, 0

mov ax, [a]

cmp ax, [b]

F. intre liniile 37 si 38 trebuie adaugat codul: push dword format