;1. Se da un sir de dublucuvinte continand date impachetate (4 octeti scrisi ca un singur dublucuvant). Sa se obtina un nou sir ;de dublucuvinte, in care fiecare dublucuvant se va obtine dupa regula: suma octetilor de ordin impar va forma cuvantul de

;ordin impar, iar suma octetilor de ordin par va forma cuvantul de ordin par. Octetii se considera numere cu semn, astfel ca ;extensiile pe cuvant se vor realiza corespunzator aritmeticii cu semn.

; Exemplu:

; pentru sirul initial:

; 127F5678h, 0ABCDABCDh, ...

; Se va obtine:

; 006800F7h, 0FF56FF9Ah

bits 32

global start

extern exit, printf

import exit msvcrt.dll

import printf msvcrt.dll

segment data use32 class=data

sir dd 127F5678h, 0ABCDABCDh

len equ ($-sir)\*4

rez times len db 0

segment code use32 class=code

start:

mov ecx, len

jecxz final

cld

mov esi, sir

mov edi, rez

repeta1:

lodsb

mov bl,al

lodsb

mov cl,al

lodsb

add bl,al

lodsb

add cl,al

mov al,bl

cbw

mov bx,ax

mov al,cl

cbw

mov cx,ax

push bx

push cx

pop eax

stosd

loop repeta1

final:

push dword 0

call [exit]

; 2.Se da un sir de cuvinte. Sa se obtina din acesta un sir de dublucuvinte, in care fiecare dublucuvant va contine nibble-urile despachetate pe octet (fiecare cifra hexa va fi precedata de un 0), aranjate crescator in interiorul dublucuvantului.

; Exemplu:

; pentru sirul initial:

; 1432h, 8675h, 0ADBCh, ...

; Se va obtine:

; 01020304h, 05060708h, 0A0B0C0Dh, ...

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

sir dw 1432h, 8657h, 0ADBCh

len equ ($-sir)/2

aux times 10 db 0

rez times len dd 0

zece db 10

segment code use32 class=code

start:

mov ecx, len

mov esi, sir

mov edi, rez

cld

formsir:

mov edx,2

repeta1:

mov eax,0

mov ebx,0

lodsb

mov bl,al

shr al,4

stosb

shl bl,4

shr bl,4

mov eax,0

mov al,bl

stosb

sub edx,1

cmp edx,0

jne repeta1

loop formsir

mov ecx,len

mov esi,0 ;i=0

mov edi,4

sortaresir:

push ecx

mov ecx,4 ;nr oct ai dd

dec ecx

fori:

mov al,[rez+esi] ;al=rez[i]

mov ebx,esi

inc ebx

forj:

mov dl,[rez+ebx]

cmp al,dl

jb nuinterschimb

mov [rez+esi],dl

mov [rez+ebx],al

mov al,dl

nuinterschimb:

inc ebx

cmp ebx,edi

jb forj

iplusplus:

inc esi

loop fori

inc esi

add edi,4

pop ecx

loop sortaresir

push dword 0

call [exit]

; 3.Se da un sir de 3 dublucuvinte, fiecare dublucuvant continand 2 valori pe cuvant (despachetate, deci fiecare cifra hexa

; e precedata de un 0). Sa se creeze un sir de octeti care sa contina acele valori (impachetate deci pe un singur octet),

; ordonate crescator in memorie, acestea fiind considerate numere cu semn.

; Exemplu:

; pentru sirul initial:

; 0702090Ah, 0B0C0304h, 05060108h

; se va obtine:

; 72h, 9Ah, 0BCh, 34h, 56h, 18h

; care se va ordona crescator:

; 9Ah, 0BCh, 18h, 34h, 56h, 72h

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

sir dd 0702090Ah, 0B0C0304h, 05060108h

lg equ ($-sir)/4

rez times 25 db 0

zece dw 16

segment code use32 class=code

start:

mov ecx,lg

mov esi,sir

mov edi,rez

cld

formsir:

lodsw

mov dx,0

idiv word[zece]

add ax,dx

stosb

lodsw

mov dx,0

idiv word[zece]

add ax,dx

stosb

loop formsir

mov eax,0

mov al,lg

mov bl,2

mul bl

mov ecx,eax

dec ecx

mov edi,eax

mov esi,0

ordonare:

fori:

mov al,[rez+esi]

mov ebx,esi

inc ebx

forj:

mov dl,[rez+ebx]

cmp al,dl

jl nuinterschimb

mov [rez+esi],dl

mov [rez+ebx],al

mov al,dl

nuinterschimb:

inc ebx

cmp ebx,edi

jb forj

next:

inc esi

loop fori

push dword 0

call [exit]

; 4.Se da un sir de octeti s. Sa se construiasca sirul de octeti d, care contine pe fiecare pozitie numarul de biti 1 ai octetului de pe pozitia corespunzatoare din s.

; Exemplu:

; s: 5, 25, 55, 127

; in binary:

; 101, 11001,10111,1111111

; d: 2, 3, 5, 7

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s db 5, 25, 55, 127

lg equ $-s

d times lg db 0

segment code use32 class=code

start:

mov ecx,lg

jecxz final

cld

mov esi,s

mov edi,d

rezolvare:

push ecx

lodsb

mov ecx,8

mov bl,0

numara:

shl al,1

adc bl,0

loop numara

mov al,bl

stosb

pop ecx

loop rezolvare

final:

push dword 0

call [exit]

;5. Se dau doua siruri de octeti s1 si s2. Sa se construiasca sirul de octeti d, care contine pentru fiecare octet din s2 pozitia sa in s1, sau 0 in caz contrar.

; Exemplu:

; pos: 1, 2, 3, 4, 5

; s1: 7, 33, 55, 19, 46

; s2: 33, 21, 7, 13, 27, 19, 55, 1, 46

; d: 2, 0, 1, 0, 0, 4, 3, 0, 5

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s1 db 7, 33, 55, 19, 46

lgs1 equ $-s1

s2 db 33, 21, 7, 13, 27, 19, 55, 1, 46

lgs2 equ $-s1-s2

d times lgs2 db 0

segment code use32 class=code

start:

mov ecx,lgs2

mov esi,0

mov edi,d

cauta:

push ecx

mov ecx,lgs1

mov al,[s2+esi]

parcurgs1:

dec ecx

mov dl,[s1+ecx]

cmp al,dl

je gasit

cmp ecx,0

ja parcurgs1

jmp adaugarezero

gasit:

mov al,cl

inc al

jmp adaugare

adaugare:

stosb

jmp next

adaugarezero:

mov al,0

stosb

next:

inc esi

pop ecx

loop cauta

push dword 0

call [exit]

; 6.Se da un sir de cuvinte s. Sa se construiasca sirul de octeti d, astfel incat d sa contina pentru fiecare pozitie din s:

; - numarul de biti de 0, daca numarul este negativ

; - numarul de biti de 1, daca numarul este pozitiv

; Exemplu:

; s: -22, 145, -48, 127

; in binary:

; 1111111111101010, 10010001, 1111111111010000, 1111111

; d: 3, 3, 5, 7

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s dw -22, 145, -48, 127

lg equ ($-s)/2

d times lg db 0

segment code use32 class=code

start:

mov esi,0

mov edi,d

mov ecx,lg

rezolvare:

push ecx

mov dx,[s+esi]

cmp dx,0

jl nrzero

jg nrunu

nrzero:

mov eax,0

mov ecx,16

nrz:

shr dx,1

jnc adaugaz

inap:

loop nrz

jmp next

adaugaz:

inc al

jmp inap

nrunu:

mov eax,0

mov ecx,16

nru:

shr dx,1

jc adaugau

inapoi:

loop nru

jmp next

adaugau:

inc al

jmp inapoi

next:

stosb

pop ecx

inc esi

inc esi

loop rezolvare

push dword 0

call [exit]

; 7.Se da un sir de dublucuvinte. Sa se obtina sirul format din octetii superiori ai

; cuvitelor superioare din elementele sirului de dublucuvinte care sunt divizibili cu 3.

; Exemplu:

; Se da sirul de dublucuvinte:

; s DD 12345678h, 1A2B3C4Dh, FE98DC76h

; Sa se obtina sirul

; d DB 12h.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s dd 12345678h, 1A2B3C4Dh, 0FE98DC76h

lg equ ($-s)/4

trei db 3

d times lg db 0

segment code use32 class=code

start:

mov ecx, lg

mov esi, s

mov edi, d

rezolvare:

push ecx

mov ecx,4

rep lodsb

mov bl,al

mov ah,0

div byte[trei]

cmp ah,0

je adauga

continua:

pop ecx

loop rezolvare

jmp final

adauga:

mov al,bl

stosb

jmp continua

final:

push dword 0

call [exit]

; 8.Se da un sir de dublucuvinte. Sa se obtina sirul format din octetii inferiori ai

; cuvintelor superioare din elementele sirului de dublucuvinte care sunt palindrom in scrierea in baza 10.

; Exemplu:

; Se da sirul de dublucuvinte:

; s DD 12345678h, 1A2C3C4Dh, 98FCDC76h

; Sa se obtina sirul

; d DB 2Ch, FCh.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s dd 12345678h, 1A2C3C4Dh, 98FCDC76h

lg equ ($-s)/4

zece db 10

d times lg db 0

segment code use32 class=code

start:

mov ecx, lg

mov esi, s

mov edi, d

rezolvare:

push ecx

mov ecx,3

mov eax,0

cld

rep lodsb ;al=2C=44

mov cl,al

mov ebx,0

ogl:

mov ah,0

div byte[zece] ;ah=4

push eax

mov eax,ebx

mul byte[zece] ;ax=0\*10

mov bx,ax ;bx=ax

pop eax

add bl,ah

cmp al,0

jne ogl

compara:

cmp cl,bl

je adauga

jmp final

adauga:

mov al,bl

cld

stosb

final:

pop ecx

inc esi

loop rezolvare

push dword 0

call [exit]

; 9.Se da un sir de dublucuvinte. Sa se obtina, incepand cu partea inferioara a dublucuvantului, dublucuvantul format din octetii

; superiori pari ai cuvintelor inferioare din elementele sirului de dublucuvinte. Daca nu sunt indeajuns octeti

; se va completa cu octetul FFh.

; Exemplu:

; Se da sirul de dublucuvinte:

; s DD 12345678h, 1A2C3C4Dh, 98FCDD76h, 12783A2Bh

; Sa se obtina dublucuvant

; d DD FF3A3C56h.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s dd 12345678h, 1A2C3C4Dh, 98FCDD76h, 12783A2Bh

lg equ ($-s)/4

zece db 10

d times lg db 0

segment code use32 class=code

start:

mov ecx,lg

mov esi,s

mov edi,d

mov ebx,0

cld

rezolvare:

push ecx

mov ecx,2

rep lodsb

test al,01h

jz adauga

jmp sari

adauga:

cld

stosb

add bl,1

sari:

add esi,2

pop ecx

loop rezolvare

cmp bl,4

jb completeaza

jmp final

completeaza:

mov al,0ffh

cld

stosb

add bl,1

cmp bl,4

jb completeaza

final:

push dword 0

call [exit]

; 10.Se da un sir A de cuvinte. Construiti doua siruri de octeti

; - B1: contine ca elemente partea superioara a cuvintelor din A

; - B2: contine ca elemente partea inferioara a cuvintelor din A

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

a dw 1234h, 0A2C0h, 8FC0h, 1278h

lg equ ($-a)/2

b1 times lg db 0

b2 times lg db 0

segment code use32 class=code

start:

mov ecx,lg

jecxz final

mov esi,a

mov edi,b2

mov ebx,b1

rezolvare:

cld

lodsb

stosb

push edi

mov edi,ebx

lodsb

stosb

inc ebx

pop edi

loop rezolvare

final:

push dword 0

call [exit]

; 13.Se da un sir S de dublucuvinte.

; Sa se obtina sirul D format din octetii inferiori ai cuvintelor inferioare din elementele sirului de dublucuvinte,

; care sunt multiplii de 7.

; Exemplu:

; s DD 12345607h, 1A2B3C15h, 13A33412h

; d DB 07h, 15h

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s dd 12345607h, 1A2B3C15h, 13A33412h

lg equ ($-s)/4

d times lg db 0

sapte db 7

segment code use32 class=code

start:

mov ecx,lg

mov esi,s

mov edi,d

rezolvare:

cld

lodsb

mov bl,al

mov ah,0

div byte[sapte]

cmp ah,0

je adauga

jmp next

adauga:

cld

mov al,bl

stosb

next:

add esi,3

loop rezolvare

push dword 0

call [exit]

; 14. Se da un sir S de dublucuvinte.

; Sa se obtina sirul D format din octetii dublucuvintelor din sirul D sortati in ordine crescatoare in interpretarea fara semn.

; Exemplu:

; s DD 12345607h, 1A2B3C15h

; d DB 07h, 12h, 15h, 1Ah, 2Bh, 34h, 3Ch, 56h

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s dd 12345607h, 1A2B3C15h

lg equ $-s

d times lg db 0

segment code use32 class=code

start:

mov ecx,lg

dec ecx

mov esi,0

fori:

push ecx

mov al,[s+esi]

mov ebx,esi

inc ebx

forj:

mov dl,[s+ebx]

cmp al,dl

ja inter

jmp peste

inter:

mov [s+esi],dl

mov [s+ebx],al

mov al,dl

peste:

inc ebx

cmp ebx,lg

jb forj

inc esi

pop ecx

loop fori

push dword 0

call [exit]

; 16.Se dau doua siruri de caractere ordonate alfabetic s1 si s2.

; Sa se construiasca prin interclasare sirul ordonat s3 care sa contina toate

; elementele din s1 si s2.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s1 db 'a','h','r','d'

lgs1 equ $-s1

s2 db 'e','f','b','p'

lgs2 equ $-s1-s2

lgs3 equ $-s1

s3 times lgs3 db 0

segment code use32 class=code

start:

mov ecx,lgs1

dec ecx

mov esi,0

fori:

push ecx

mov al,[s1+esi]

mov ebx,esi

inc ebx

forj:

mov dl,[s1+ebx]

cmp al,dl

ja inter

jmp peste

inter:

mov [s1+esi],dl

mov [s1+ebx],al

mov al,dl

peste:

inc ebx

cmp ebx,lgs1

jb forj

inc esi

pop ecx

loop fori

mov ecx,lgs2

dec ecx

mov esi,0

fori2:

push ecx

mov al,[s2+esi]

mov ebx,esi

inc ebx

forj2:

mov dl,[s2+ebx]

cmp al,dl

ja inter2

jmp peste2

inter2:

mov [s2+esi],dl

mov [s2+ebx],al

mov al,dl

peste2:

inc ebx

cmp ebx,lgs2

jb forj2

inc esi

pop ecx

loop fori2

mov ecx,0

mov ebx,0

mov edi,0

f:

mov al,[s1+ecx]

mov dl,[s2+ebx]

cmp al,dl

jbe ads1

ja ads2

ads1:

mov [s3+edi],al

inc ecx

jmp continua

ads2:

mov [s3+edi],dl

inc ebx

continua:

inc edi

cmp ecx,lgs1

jae cont

cmp ebx,lgs2

jae cont

jmp f

cont:

cmp ecx,lgs1

jb adaugas1

jmp next

adaugas1:

mov al,[s1+ecx]

mov [s3+edi],al

inc ecx

inc edi

cmp ecx,lgs1

jb adaugas1

jmp next

next:

cmp ebx,lgs2

jb adaugas2

jmp final

adaugas2:

mov dl,[s2+ebx]

mov [s3+edi],dl

inc ebx

inc edi

cmp ebx,lgs2

jb adaugas2

final:

push dword 0

call [exit]

; 17.Se da un sir de dublucuvinte. Sa se ordoneze descrescator sirul cuvintelor inferioare ale acestor dublucuvinte.

; Cuvintele superioare raman neschimbate.

; Exemplu:

; dandu-se:

; sir DD 12345678h 1256ABCDh, 12AB4344h

; rezultatul va fi

; 1234ABCDh, 12565678h, 12AB4344h.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s dd 12345678h, 1256ABCDh, 12AB4344h

lg equ ($-s)/4

lgl equ lg\*4

segment code use32 class=code

start:

mov ecx,lg

dec ecx

mov esi,0

rezolvare:

push ecx

mov eax,[s+esi]

mov edi, esi

add edi,4

forj:

mov ebx,[s+edi]

cmp ax,bx

jb interschimba

jmp peste

interschimba:

mov edx,eax

mov ax,bx

mov bx,dx

mov [s+esi],eax

mov [s+edi],ebx

mov eax,ebx

peste:

add edi,4

cmp edi,lgl

jb forj

pop ecx

add esi,4

loop rezolvare

push dword 0

call [exit]

; 19.Dandu-se doua siruri de octeti sa se calculeze toate pozitiile unde al doilea sir apare ca subsir in primul sir.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s1 db 12h, 13h, 14h, 20h, 21h, 12h, 13h, 14h, 18h, 19h, 12h, 13h

lgs1 equ $-s1

s2 db 12h, 13h, 14h

lgs2 equ $-s1-s2

d times 10 db 0

segment code use32 class=code

start:

mov ecx,lgs1

mov esi,0

mov edi,d

rezolvare:

push ecx

mov al,[s1+esi]

mov dl,[s2]

cmp al,dl

je parcurg2

jmp peste

parcurg2:

mov ecx, lgs2

dec ecx

mov ebx,1

inc esi

forj:

mov al,[s1+esi]

mov dl,[s2+ebx]

cmp al,dl

jne peste

cmp esi,lgs1

ja peste

inc ebx

inc esi

loop forj

mov eax,esi

sub eax,ebx

stosb

peste:

pop ecx

inc esi

loop rezolvare

push dword 0

call [exit]

; 21. Dandu-se un sir de cuvinte sa se obtina sirul (de octeti) cifrelor in baza zece ale fiecarui cuvant din acest sir.

; Exemplu:

; daca avem sirul:

; sir DW 12345, 20778, 4596

; obtinem rezultatul

; 1, 2, 3, 4, 5, 2, 0, 7, 7, 8, 4, 5, 9, 6.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s dw 12345, 20778, 4596, 12341

lg equ ($-s)/2

zece dw 10

lgc equ lg\*5

d times 90 db 0

aux times lg\*5 db 0

segment code use32 class=code

reset:

pushad

mov edi,aux

mov al,0

mov ecx,lgc

cld

rep stosb

popad

ret

start:

mov ecx,lg

mov esi,s

mov edi,d

rezolvare:

push ecx

push edi

cld

lodsw

mov edi,aux

mov ebx,0

call reset

parsare:

mov dx,0

div word[zece]

push eax

mov al,dl

cld

stosb

inc ebx

pop eax

cmp ax,0

jne parsare

adaugare:

mov ecx,ebx

dec ecx

add ecx,1

pop edi

ad:

dec ebx

mov al,[aux+ebx]

cld

stosb

loop ad

pop ecx

loop rezolvare

push dword 0

call [exit]

; 22.Se da un sir de octeti 'input' si inca doua siruri de dimensiune N fiecare, 'src' si 'dst'.

; Sa se obtina un nou sir 'output' din sirul 'input' in care se vor inlocui toti

; octetii cu valoarea src[i] cu dst[i], unde i=1..N.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s db 1h,2h,3h,4h,5h

lg equ $-s

n equ 5

src db 1h,5h,4h,3h,2h

dst db 10h,50h,40h,30h,20h

d times lg db 0

segment code use32 class=code

start:

mov ecx,lg

mov esi,0

mov edi,d

rezolvare:

push ecx

mov al,[s+esi]

mov ecx,n

mov ebx,0

cauta:

mov dl,[src+ebx]

cmp al,dl

je schimb

jmp peste

schimb:

mov dl,[dst+ebx]

mov [s+esi],dl

peste:

inc ebx

loop cauta

inc esi

pop ecx

loop rezolvare

push dword 0

call [exit]

; 23.Dandu-se un sir de octeti sa se obtina un sir de cuvinte care sa contina in octetii inferiori multimea

; caracterelor din sirul de octeti, iar octetul superior al unui cuvant sa contina numarul de aparitii al octetului

; inferior din acel cuvant in sirul de octeti dat.

; Exemplu:

; se da sirul

; sir DB 2, 4, 2, 5, 2, 2, 4, 4

; se va obtine sirul

; rez DW 0402h, 0304h, 0105h.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s db 2, 4, 2, 5, 2, 2, 4, 4

lg equ $-s

d times lg dw 0

aux times lg dw 0

nrap db 0

lgaux db 0

segment code use32 class=code

eliminaredubl:

pushad

mov ecx,lg

mov esi,s

mov edi,aux

mov ebx,0

repeta:

push ecx

lodsb

mov ecx,ebx

jecxz p

mov edx,0

repeta1:

mov dl,[aux+edx]

cmp al,dl

je peste

inc edx

loop repeta1

p:

mov [aux+ebx],al

inc ebx

peste:

pop ecx

loop repeta

mov byte[lgaux],bl

popad

ret

cauta:

pushad

mov byte[nrap],0

mov ecx,lg

mov esi,0

fori:

cld

mov bl,[s+esi]

cmp al,bl

je contor

jmp pest

contor:

add byte[nrap],1

pest:

inc esi

loop fori

popad

ret

start:

mov edi,d

call eliminaredubl

mov ecx,[lgaux]

mov ebx,0

rezolvare:

push ecx

mov al,[aux+ebx]

call cauta

mov ah,[nrap]

cld

stosw

inc ebx

pop ecx

loop rezolvare

push dword 0

call [exit]

; 24.Dandu-se un sir de dublucuvinte, sa se obtina un alt sir de dublucuvinte in care se vor pastra doar dublucuvintele din

; primul sir care au un numar par de biti cu valoare 1.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s dd 12h,0Bh, 2Bh

lg equ ($-s)/4

d times lg dw 0

segment code use32 class=code

start:

mov ecx,lg

mov esi,s

mov edi,d

rezolvare:

push ecx

cld

lodsd

mov ecx,32

mov ebx,eax

mov dl,0

numara:

shr ebx,1

jc contor

jmp peste

contor:

add dl,1

peste:

loop numara

add dl,0

jnp ad

jmp sari

ad:

cld

stosd

sari:

pop ecx

loop rezolvare

push dword 0

call [exit]

; 25.Se da un sir de octeti. Sa se obtina sirul oglindit al reprezentarii binare a acestui sir de octeti.

; Exemplu:

; Se da sirul de octeti:

; s DB 01011100b, 10001001b, 11100101b

; Sa se obtina sirul

; d DB 10100111b, 10010001b, 00111010b.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s db 01011100b, 10001001b, 11100101b

lg equ $-s

d times lg db 0

segment code use32 class=code

start:

mov ecx, lg

mov esi,lg

dec esi

mov edi,d

rezolvare:

push ecx

mov al,[s+esi]

mov ebx,0

mov ecx,8

repeta:

shr al,1

rcl bl,1

loop repeta

mov al,bl

cld

stosb

dec esi

pop ecx

loop rezolvare

push dword 0

call [exit]

; 28. Dandu-se un sir de octeti si un subsir al sau,

; sa se elimine din primul sir toate aparitiile subsirului.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s db 3,4,5,1, 2, 3, 4, 5, 6, 7, 8, 3, 4, 5,34

len equ $-s ; numarul de elemente ale sirului s

s1 db 3, 4, 5,6,11

len1 equ $-s1 ; numarul de elemente ale subsirului s1

d times len db 0 ; noul sir obtinut prin eliminare

poz db 0

segment code use32 class=code

verific:

push ecx

push esi

mov ecx,len1

dec ecx

mov ebx,1

mov edi,1

inc esi

repeta1:

mov al,[s+esi]

mov dl,[s1+edi]

cmp al,dl

jne pest

inc edi

inc esi

loop repeta1

jmp sari

pest:

mov ebx,0

sari:

pop esi

pop ecx

ret

sterge:

pushad

mov ecx,len

sub ecx,len1

sub ecx,[poz]

mov ebx,len1

repeta2:

mov al,[s+esi+ebx]

mov [s+esi],al

inc esi

loop repeta2

mov ecx,len1

repeta3:

mov al,0

mov [s+esi],al

inc esi

loop repeta3

popad

ret

start:

mov ecx,len

mov esi,0

rezolvare:

mov al,[s+esi]

mov dl,[s1+0]

cmp al,dl

jne mai\_departe

mov [poz],esi

call verific

cmp ebx,1

jne mai\_departe

call sterge

mov esi,[poz]

jmp maidep

mai\_departe:

inc esi

maidep:

loop rezolvare

push dword 0

call [exit]

; 29.Se dau doua siruri de octeti. Sa se parcurga cel mai scurt sir dintre cele doua siruri si sa se construiasca un al treilea

; sir care va contine cel mai mare element de acelasi rang din cele doua siruri, iar pana la lungimea celui mai lung sir, sirul

; al treilea se va completa alternativ cu valoarea 1 respectiv 0.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s1 db 1,2,3,4,8,9,10,11

lgs1 equ $-s1

s2 db 2,1,5,3

lgs2 equ $-s1-s2

s3 times 100 db 0

poz db 0

segment code use32 class=code

start:

mov edx,1

mov ecx,lgs1

mov byte[poz],lgs2

mov ebx,lgs2

cmp ecx,ebx

jb rezv

mov ecx,lgs2

mov byte[poz],lgs1

mov edx,0

rezv:

mov esi,0

mov ebx,0

mov edi,s3

cmp edx,1

jne rezolv2

rezolv1:

mov al,[s1+esi]

mov dl,[s2+ebx]

cmp al,dl

jb ads2

jmp fin1

ads2:

mov al,[s2+ebx]

fin1:

stosb

inc esi

inc ebx

loop rezolv1

jmp final

rezolv2:

sir2:

mov al,[s2+esi]

mov dl,[s1+ebx]

cmp al,dl

jb ads1

jmp fin2

ads1:

mov al,[s1+esi]

fin2:

stosb

inc esi

inc ebx

loop rezolv2

final:

cmp ebx,[poz]

ja finalt

mov al,0

repeta10:

stosb

cmp al,0

je unu

jne zero

unu:

mov al,1

jmp p

zero:

mov al,0

p:

inc ebx

cmp ebx,[poz]

jb repeta10

finalt:

push dword 0

call [exit]

; 30.Se da un sir de cuvinte. Sa se construiasca doua siruri de octeti, s1 si s2, astfel: pentru fiecare cuvant,

; daca numarul de biti 1 din octetul high al cuvantului este mai mare decat numarul de biti 1 din octetul low,

; atunci s1 va contine octetul high, iar s2 octetul low al cuvantului

; daca numarul de biti 1 din cei doi octeti ai cuvantului sunt egali, atunci s1 va contine numarul de biti 1 din octet,

; iar s2 valoarea 0

; altfel, s1 va contine octetul low, iar s2 octetul high al cuvantului.

bits 32

global start

extern exit

import exit msvcrt.dll

segment data use32 class=data

s dw 1234h,1000h,2000h

len equ ($-s)/2

nrc db 0

s1 times 10 db 0

s2 times 10 db 0

segment code use32 class=code

nroctet:

pushad

mov byte[nrc],0

mov bl,al

mov ecx,8

rep1:

shr bl,1

jnc sari

add byte[nrc],1

sari:

loop rep1

popad

ret

start:

mov ecx,len

mov esi,s

mov edi,0

rezolvare:

push ecx

cld

lodsb

call nroctet

mov bl,[nrc]

cld

lodsb

call nroctet

mov dl,[nrc]

cmp bl,dl

jb maimic

ja maimare

mov [s1+edi],dl

mov cl,0

mov [s2+edi],cl

jmp peste

maimic:

mov [s1+edi],dl

mov [s2+edi],bl

jmp peste

maimare:

mov [s1+edi],bl

mov [s2+edi],dl

peste:

inc edi

pop ecx

loop rezolvare

push dword 0

call [exit]