9.

Se da un sir de octeti S de lungime l. Sa se construiasca sirul D de lungime l-1 astfel incat elementele din D sa reprezinte diferenta dintre fiecare 2 elemente consecutive din S.  
**Exemplu:**

S: 1, 2, 4, 6, 10, 20, 25

D: 1, 2, 2, 4, 10, 5

bits 32 ; assembling for the 32 bits architecture

; declare the EntryPoint (a label defining the very first instruction of the program)

global start

; declare external functions needed by our program

extern exit ; tell nasm that exit exists even if we won't be defining it

import exit msvcrt.dll ; exit is a function that ends the calling process. It is defined in msvcrt.dll

; msvcrt.dll contains exit, printf and all the other important C-runtime specific functions

; our data is declared here (the variables needed by our program)

segment data use32 class=data

; ...

s db 1, 2, 4, 6, 10, 20, 25

l equ $-s

d times l-1 db 0

; our code starts here

segment code use32 class=code

start:

;mov ecx, l ;punem lungimea in ECX pentru a putea realiza bucla loop de ecx ori

mov ESI, 1; esi=1

Repeta:

mov AL, [s+ESI] ; al ia valorile al 2lea element pana la ultimul element din s

mov BL, [s+ESI-1]; bl ia valorile de la primul la penultimul element din s

sub AL, BL ; al=al-bl

mov [d+ESI], AL ; mov al on the corresponding byte o the result

inc ESI ; increment esi by 1

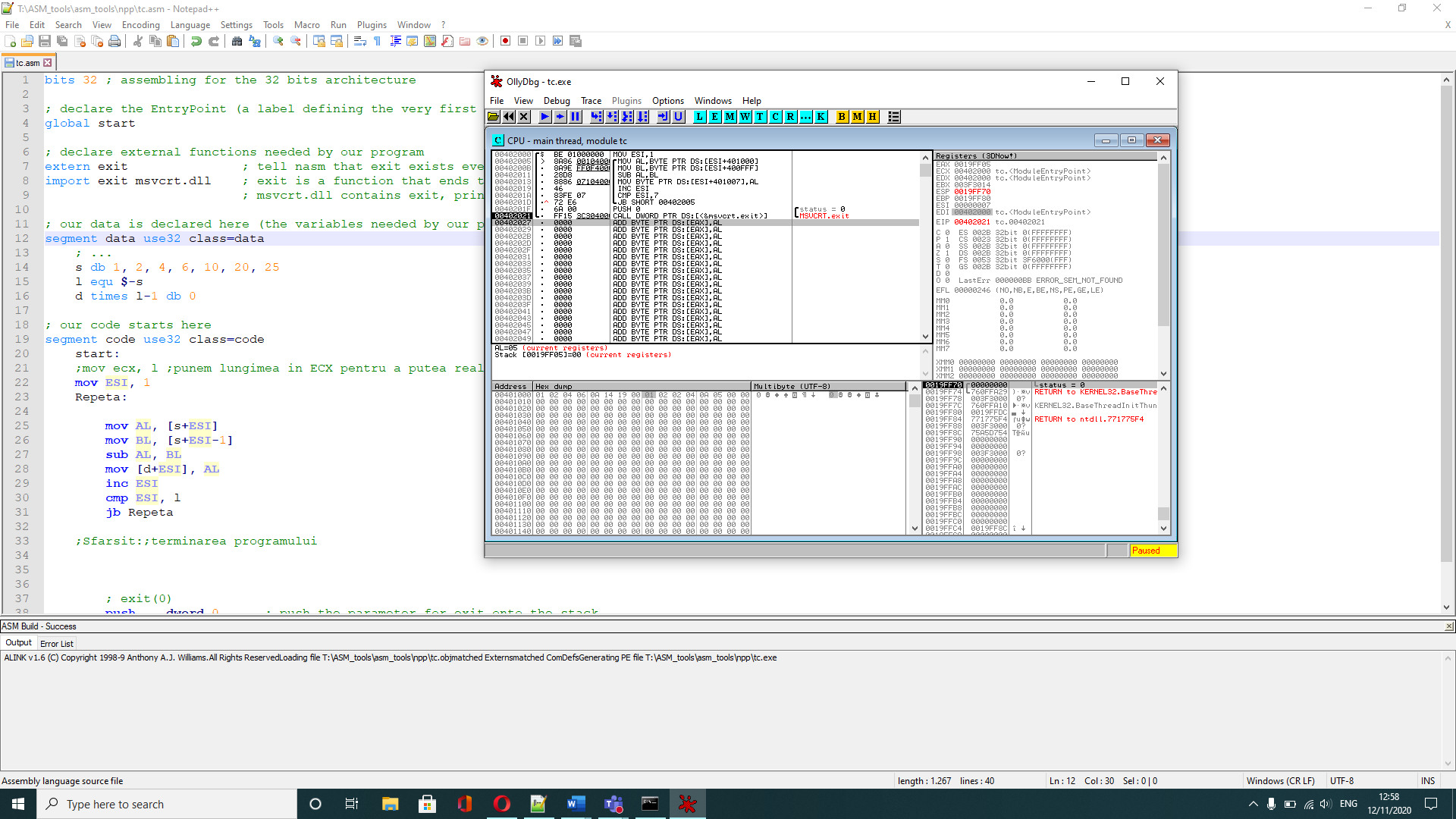
cmp ESI, l ; compare esi to l in order to know when to finish the loop

jb Repeta ;repeat the instructions until esi=l

; exit(0)

push dword 0 ; push the parameter for exit onto the stack

call [exit] ; call exit to terminate the program



8.

Se da un sir de caractere S. Sa se construiasca sirul D care sa contina toate literele mari din sirul S.  
**Exemplu:**

S: 'a', 'A', 'b', 'B', '2', '%', 'x', 'M'

D: 'A', 'B', 'M'

bits 32 ; assembling for the 32 bits architecture

; declare the EntryPoint (a label defining the very first instruction of the program)

global start

; declare external functions needed by our program

extern exit ; tell nasm that exit exists even if we won't be defining it

import exit msvcrt.dll ; exit is a function that ends the calling process. It is defined in msvcrt.dll

; msvcrt.dll contains exit, printf and all the other important C-runtime specific functions

; our data is declared here (the variables needed by our program)

segment data use32 class=data

; ...

s db 'a', 'A', 'b', 'B', '2', '%', 'x', 'M'

l equ $-s

d times l db 0

; our code starts here

segment code use32 class=code

start:

mov esi, 0 ;esi=0

mov edi, 0 ;edi=0

Repeta:

mov al, [s+esi] ;al gets the value of each element of the string

cmp al, 'Z' ; compare al to Z

ja not\_capital; if al is grater than Z jumps to label

cmp al, 'A; compare al to A

jb not\_capital ; if al is less than A jumps to label

mov [d+edi], al ;moves al to the result on the corresponding position

inc edi ;increment edi by 1

not\_capital:

inc esi; increment esi by 1

cmp esi, l; compare esi to l

jb Repeta ;if esi is less than l repeats the loop

Push dword 0

Call [exit]

; exit(0)

push dword 0 ; push the parameter for exit onto the stack

call [exit] ; call exit to terminate the program

