18 feb 2025 - retake

I, au fost 3 subpuncte:

a) primele 5 chestii care sunt și aici la flag-uri

b) Moduri de adresare a operanzilor (alea cu immediate, register, memory) și ce reprezintă direct și indirect addressing + exemple

c) Prefixele din formatul intern al instrucțiunilor + exemple

II

memory layout dar mult mai puțin tricky și cu 10 instrucțiuni in loc de 15

III:

a) ce s-a dat și în 02.08.2018 dar cu mici modificări:

a1).

```
mov ax,1001h
mov bl, 1000b+10b
div bl
```

a2).

```
mov ah, Obeh
mov al, Odch
add ah,al
```

a3).

```
mov ax, 0020h
mov bx, 1000b
```

imul bl

a4).

mov dh, 62h

mov ch, 200

sub dh,ch

Apoi, conceptul de overflow, cum se aplica pe operațiile aritmetice și exemple unde apare în astea 4 secvențe.

Apoi, modifica "mov ax, 1001h" din a1) astfel încât CF=1 și OF=0

Apoi, modifica "mov ah, 0beh" din a2) astfel încât SF=ZF=1 și să explici la amândoua.

b) Ai secventa:

pop eax

mov [esp-4], eax

Scrie o singura instrucțiune echivalenta cu secventa care să aibă același efect asupra stivei, atât ca valori cat și ca structura

4)

Se citește un număr N de la tastatura, din intervalul [0, 255] și se dă un șir de caractere S in data segment care reprezintă o propoziție in care cuvintele sunt separate doar prin spatii. Daca N-ul citit nu e in interval, programul trebuie sa se oprească. Sa se afișeze toate cuvintele din sir care au checksum-ul mai mare decât N, in ordine descrescătoare a checksum-urilor. Checksum-ul unui cuvânt reprezintă suma caractelor dintr-un cuvânt, fiecare caracter fiind codificat astfel:

'A', 'B', 'C', ..., 'Z' = 11, 12, ..., 36

'a', 'b', 'c', ..., 'z' = 11, 12, ..., 36

'0', '1', ..., '9' = 9, 8, .., 0

Orice alt caracter = 10