1. Sa se scrie un program in limbaj de asamblare care sa numere de cate ori apare caracterul ‘c’ in urmatorul sir de caractere aflat in memorie.” *Poate veti fi surprinsi sa aflati ca lista instructiunilor în cod masina este destul de scurta.”*.

2. Se dau doua numere pozitive a si b reprezentate pe 16 biti. Sa se scrie un program in limbaj de asamblare care sa adune cele doua numere daca primul este mai mare decât al doilea sa se adune, iar rezultatul sa se plaseze in memorie la adresa F0h. Daca nu sa se calculeze diferenta celor doua numere, iar rezultatul sa se plaseze la aceeasi adresa.
Caz particular a= F312h, b=5678h.

3. Se dau doua numere a si b intregi pozitive reprezentate pe 8 biti. Sa se scrie un program in limbaj de asamblare care daca primii trei biti ai acestor numere sunt identici cele doua numere sa se deplaseze la stânga cu trei biti, iar la dreapta numarului sa se introduca 1 si sa se scada numarul mai mare din numarul mai mic.
Caz particular a= F3h, b=56h

4. Se dau trei numere a,b,c pe intregi pozitive reprezentate pe 8 biti. Sa se scrie un program in limbaj de asamblare care sa determine daca cele trei numere formeaza laturile unui triunghi(a+b>c, a+c>b, b+c>a). Daca nu sa se afiseze un mesaj de eroare.
Caz particular a=9,b=4,c=16.

5. Se da un numar intreg pozitiv pe 8 biti. Sa se scrie un program in limbaj de asamblare care daca prima jumatate este mai mica decât a doua jumatate cele doua jumatati sa îsi schimbe locurile. Sa se scada din numarul intreg pozitiv format din prima jumatate numarul intreg pozitiv format din a doua jumatate, iar rezultatul se se afiseze în format decimal(în fereastra de afisare sa apara echivalentul decimal al numarului). Caz particular AB

6. Se da urmatorul program scris in limbaj de asamblare. Sa se scrie in limbaj masina.

```
        load    R1,Text     ;the start of the string
        load    R2,1        ;increase step
        load    R0,0        ;string-terminator
NextChar:load   RF,[R1]     ;get character and print it on screen
        addi    R1,R1,R2    ;increase address
        jmpEQ   RF=R0,Ready ;when string-terminator, then ready
```

```
        jmp     NextChar    ;next character
Ready:  halt

Text:   db      10
        db      "Hello world !!",10
        db      "    from the",10
        db      "  Simple Simulator",10
        db      0
```

Tabela codurilor masina:

OPCODE	OPERANZI	INSTRUCTIUNE
2	RXY	Load R,XY
1	RXY	Load R,[XY]
3	RXY	Store R,[XY]
D	0RS	Load R,[S]
E	0RS	Store R,[S]
4	0RS	Move R,[S]
5	RST	Addi R,S,T
6	RST	Addf R,S,T
7	RST	Or R,S,T
8	RST	And R,S,T
9	RST	Xor R,S,T
A	R0X	Ror R,X
B	RXY 0XY	JmpEQ R=R0,XY Jmp XY
F	RXY	JmpLE R<=R0,XY
C	000	Halt

7. Se dau doua numere intregi pozitive pe 8 biti. jumatate Sa se scrie un program in limbaj de asamablare care daca bitii din prima jumatate formeaza doua numere intregi pozitive egale sa se înmulteasca numerele intregi pozitive formate din bitii jumatatilor cele mai putin semnificative.
Caz particular: FA si EF

8. Se da un numar interg pozitiv pe 8 biti . Sa se scri un program in limbaj de asamblare care sa inmulteasca numarul intreg pozitiv format de jumatatea cea mai semnificativa cu numartul intreg pozitiv format de jumatatea cea mai putin semnificativa.
Caz particular: CD

9. Se dau doua numere in format BCD necompactat. Sa se scrie un program in limbaj de asamblare care sa le adune Caz particular: 34 si 35

10. Se dau doua numere in format EXCESS 3 compactat. Sa se scrie un program in limbaj de asamblare care sa le treaca in format necompact si sa le adune.
Caz particular: 34 si 35 in format decimal

11. Se da un numar de doua cifre in format BCD compactat. Sa se scrie un program in limbaj de asamblare care sa le converteasca in cod binar.
Caz particular 67

12. Se dau doua numere intregi binare pozitive a,b. Sa se scrie un program in limbaj de asamblare care daca cele doua numere au acelasi numar de biti 1, sa se adune printr-o metoda care permite detectarea transportului. Daca nu sa scada din numarul mai mare numarul mai mic.
Caz Particular 81h si A0h

13. Se da urmatorul program scris in limbaj de asamblare. Sa se scrie in limbaj masina.

```
                load R1,[inmultit]
                load R2,[inmultitor]
                load R3,0x01
                load R4,0x01
nextbit:  and R0,R2,R3
                jmpEQ R3=R0,shiftAndAdd
                jmp shift
shiftAndAdd:addi R5,R5,R1
shift:          ror R1,7
                ror R3,7
                addi RC,RC,R4
                load R0,4
                jmpEQ RC=R0,stop
                jmp nextbit
stop:           halt
inmultit:       db 0x02
inmultitor:     db 0x0F
```

Tabela codurilor masina:

OPCODE	OPERANZI	INSTRUCTIUNE
2	RXY	Load R,XY
1	RXY	Load R,[XY]
3	RXY	Store R,[XY]
D	0RS	Load R,[S]
E	0RS	Store R,[S]
4	0RS	Move R,[S]
5	RST	Addi R,S,T
6	RST	Addf R,S,T
7	RST	Or R,S,T
8	RST	And R,S,T
9	RST	Xor R,S,T

A	R0X	Ror R,X
B	RXY 0XY	JmpEQ R=R0,XY Jmp XY
F	RXY	JmpLE R<=R0,XY
C	000	Halt