

1) Dado o seguinte programa indique o valor dos registos no fim do programa, e escreva na memória os valores alterados.

```
MOV R5, 150H
MOV R1, 140H
MOV R2, 4H
MOV R3, [R1]
ADD R1, R2
MOVB R4, [R1]
MOV [R1+8], R5
MOV [R1+R2], R5
ADD R5, 1H
ADD R1, 2H
MOVB [R1], R2
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
140H	A1	B2	C3	4F	7A	87	04		01	50			01	50		
150H																

Tab1: Memória de dados

R1=0146H	R2=0004H	R3=A1B2H
R4=007AH	R5=0151H	

2) Dado o seguinte programa indique o valor dos registos quando terminar a sua execução e o valor dos símbolos apresentados na tabela.
seguinte:

```
Place 120H
MAX EQU 110H
A: Word 130H
X: TABLE 3
K: WORD 5678H
Y: STRING "casa"
MIN EQU 100H
Z: STRING "branca"

Place 0h
Inicio: MOV R1, 120H
        MOV R2, [R1]
        MOV R3, [R1+8]
Fim: JMP Fim
```

SIMBOLO	VALOR	REGISTO	VALOR
MAX	0110H	R1	0120H
X	0122H	R2	0130H
K	0128H	R3	5678H
Y	012AH		
MIN	0100H		
Z	012EH		
A	0120H		
Inicio	0000H		
Fim	0006H		

3) Dado o seguinte programa indique o valor dos registos quando terminar a sua execução e preencha a memória com o conteúdo do stack.

```
PLACE 0
INI: MOV SP, 208H
      MOV R1, 06H
      CALL FUN
FIM2: JMP FIM2
FUN: PUSH R1
      MOV R2, 0H
MEIO: SUB R1, 1
      JZ FIM
      ADD R2, R1
      JMP MEIO
FIM: POP R1
      RET
```

Endereço	Dados
200H	
202H	
204H	0006H
206H	0006H
208H	
20AH	
20CH	
20EH	

R1	0006H
R2	000FH
SP	0208H

CALL Eti	SP ← SP-2 M[SP] ← PC PC ← Eti
----------	-------------------------------------

RET	PC ← M[SP] SP ← SP+2
-----	-------------------------