

1. Elaborar um programa para o montador do Atmeg2560 que carregue o valor 0x23 na posição de memória 0x200. O programa deve comparar o valor armazenado na posição com um valor que esteja armazenado no registrador 0 de uso geral. Se o valor for igual, o programa deve armazenar na posição 0x201 o byte 0xAA; se for diferente, deve armazenar na posição o byte 0xBB.

```
; 1.Elaborar um programa para o montador do Atmeg2560 que carregue o valor 0x23
na posição de memória 0x200.
; O programa deve comparar o valor armazenado na posição com um valor que esteja
armazenado no registrador 0 de uso geral.
; Se o valor for igual, o programa deve armazenar na posição 0x201 o byte 0xAA;
se for diferente, deve armazenar na posição o byte 0xBB.
```

```
; guardar o valor em uma posição de memória
ldi r16,0x23 ; carrega em r16 o byte
sts 0x200,r16 ; transfere de r16 para a memória

ldi r16,0x11 ; simula um valor em r0
mov r0,r16

ldi r16,0xAA ; inicializa os valores a serem transferidos para memória
ldi r17,0xBB

lds r3,0x200 ; carrega o dado da posição 0x200 em r3
cp r0,r3 ; compara o valor de r0 com o valor de r3
brne rotulo ; desvia se registradores não são iguais
sts 0x201,r16 ; carrega 0xAA se os valores forem iguais
jmp fim
rotulo: sts 0x201,r17 ; carrega 0xBB se os valores forem diferentes
fim: break
```

2. Elaborar um programa para o montador do Atmeg2560 para armazenar os valores 0x01 a 0x0A a partir das posições de endereço 0x201.

```
; 2. Elaborar um programa para o montador do Atmeg2560 para armazenar os
valores 0x01 a 0x0A a partir das posições de endereço 0x201.
```

```
ldi r16,0x01 ; inicializa r16 com 0x01
ldi r17,0x0B ; determina flag
ldi r18,0x02 ; carrega endereço inicial em X
mov r27,r18
ldi r19,0x01
mov r26,r19
loop: st x,r16 ; armazena dado
inc r26
inc r16
cp r16,r17
```

```
brne loop
break
```

3. Elaborar um programa para o montador do Atmeg2560 que crie uma tabela de dados em memória a partir da posição 0x200, para armazenar 4 bytes, com os valores apresentados abaixo. O programa deve um valor no registrador de I/O no endereço 0x20 e retornar para o registrador de I/O no endereço 0x21 o byte 0x11 se o valor for encontrado. Caso o valor não seja encontrado, retornar o byte 0xFF.

0x200	0x12
0x201	0x11
0x202	0x3A
0x203	0xBB

```
; crie uma tabela de dados em memória a partir da posição 0x200, para armazenar 4
bytes,
; com os valores apresentados abaixo.
; O programa deve um valor no registrador de I/O no endereço 0x20 e
; retornar para o registrador de I/O no endereço 0x21 o byte 0x11 se o valor for
encontrado.
; Caso o valor não seja encontrado, retornar o byte 0xFF.
```

```
; cria a tabela
```

```
ldi r16,0x12 ; carrega em r16 o valor da tabela
sts 0x200,r16 ; transfere de r16 para a memória
ldi r16,0x11
sts 0x201,r16
ldi r16,0x3A
sts 0x202,r16
ldi r16,0xBB
sts 0x203,r16
```

```
in r0,0x00 ; lê o valor no registrador 0 de I/O - endereço 0x20
```

```
ldi r16,0x11 ; inicializa os valores de saída
ldi r17,0xFF
```

```
lds r3,0x200 ; carrega a primeira linha da tabela em r3
cp r0,r3 ; compara o valor de r0 com o valor de r3
brne d2 ; desvia se registradores não são iguais
out 0x01,r16
jmp fim
d2: lds r3,0x201 ; carrega a segunda linha da tabela
cp r0,r3 ; compara o valor de r0 com o valor de r3
brne d3 ; desvia se registradores não são iguais
out 0x01,r16 ; carrega no registrador de I/O 0x11
jmp fim
d3: lds r3,0x202 ; carrega a terceira linha da tabela
cp r0,r3 ; compara o valor de r0 com o valor de r3
brne d4 ; desvia se registradores não são iguais
out 0x01,r16 ; carrega no registrador de I/O 0x11
jmp fim
d4: ldi r16,0xFF ; carrega o valor 0xFF no registrador de I/O
out 0x01,r16 ; carrega no registrador de I/O 0xFF
jmp fim
```

```
d4: lds r3,0x203 ; carrega a quarta linha da tabela
cp r0,r3 ; compara o valor de r0 com o valor de r3
brne d5 ; desvia se registradores não são iguais
out 0x01,r16
jmp fim
d5: out 0x01,r17
fim: break
```