

1. Elaborar um programa para o montador do Atmega2560 que leia os 10 bytes armazenados a partir da posição de endereço 0x3AA. O programa deve calcular e escrever na posição logo após o último byte a soma do complemento 2 dos números.

Para teste simular os dados apresentados a seguir (em hexa):

End	Byte	End	Byte	End	Byte	End	Byte	End	Byte
3AA	2B	3AC	0F	3AE	02	3B0	A2	3B2	2C
3AB	1A	3AD	11	3AF	01	3B1	1F	3B3	11

; 1. Elaborar um programa para o montador do Atmega2560 que leia os 10 bytes armazenados a partir da posição de endereço 0x3AA.  
 ; O programa deve calcular e escrever na posição logo após o último byte a soma do complemento 2 dos números.

```
.DEF rflag = r16 ; a diretiva def define um nome simbólico para o registrador
.DEF regX0 = r26
.DEF regX1 = r27
.DEF rcont = r17
.DEF aux = r18
.SET flag = 0xB4 ; a diretiva set define um nome simbólico para um dado alterável
.SET end0 = 0xAA
.SET end1 = 0x03
```

; carrega na memória dados de simulação

```
ldi aux,0x2B
sts 0x3AA,aux
ldi aux,0x1A
sts 0x3AB,aux
ldi aux,0x0F
sts 0x3AC,aux
ldi aux,0x11
sts 0x3AD,aux
ldi aux,0x02
sts 0x3AE,aux
ldi aux,0x01
sts 0x3AF,aux
ldi aux,0xA2
sts 0x3B0,aux
ldi aux,0x1F
sts 0x3B1,aux
ldi aux,0x2C
sts 0x3B2,aux
ldi aux,0x11
sts 0x3B3,aux
```

; código

```
ldi regX0,end0 ; inicializa endereço
ldi regX1,end1
ldi rcont,0x00 ;inicializa soma
ldi rflag,flag ; carrega flag no registrador
loop: ld aux,x ; carrega dado da memória em r0
com aux ; determina complemento de 1
inc aux ; incrementa - complemento de 2
add rcont,aux ; carrega resultado no acumulador
```

```
inc regX0
cp regX0,rflag ; compara endereço com endereço flag
brne loop
st x,rcont
break
```

2. Elaborar um programa para o montador do Atmega2560 comparar um valor armazenado na posição de memória 0x200 com o resultado da expressão  $2*k+10$ , onde k é o dado armazenado na posição 0x200. Se os valores forem iguais remeter a continuação do programa para o endereço 0x300; se forem diferentes, remeter para 0x400. Admitir para simulação que o endereço 0x200 armazena 0x07 e o endereço 0x201 armazena 0x1A; e, respectivamente, 0x07 e 0x18. Admitir que o maior valor numérico representado é 255d.

```
; 1. Elaborar um programa para o montador do Atmega2560 que leia os 10 bytes
armazenados a partir da posição de endereço 0x3AA.
; O programa deve calcular e escrever na posição logo após o último byte a soma
do complemento 2 dos números.
```

```
.DEF regX0 = r26 ; a diretiva def define um nome simbólico para o registrador
.DEF regX1 = r27
.DEF reg1 = r16
.DEF reg2 = r17
.DEF aux = r18
.SET end0 = 0x00
.SET end1 = 0x02
```

```
; carrega na memória dados de simulação
```

```
ldi aux,0x07
sts 0x200,aux
ldi aux,0x18
sts 0x201,aux
```

```
; código
```

```
ldi regX0,end0 ; inicializa endereço
ldi regX1,end1
ld reg1,x ; carrega o conteúdo de x em aux1
lsl reg1
ldi aux,0x0A
add reg1,aux
inc regX0
ld reg2,x ; carrega o conteúdo de x em aux2
cp reg1,reg2
breq igual
jmp 0x400
igual: jmp 0x300
```

3. Elaborar um programa para o montador do Atmega2560 que leia 1 byte de um dispositivo de entrada. O programa deve determinar o número de 1s no byte lido.

A paridade esperada é par (0, 2, 4 ou 6 bits). Se o número for par o programa deve remeter a execução para o endereço de memória 0x300.

4. ; 3. Elaborar um programa para o montador do Atmeg2560 que leia 1 byte de um dispositivo de entrada.
5. ; O programa deve determinar o número de 1s no byte lido. A paridade esperada é par (0, 2, 4 ou 6 bits).
6. ; Se o número for par o programa deve remeter a execução para o endereço de memória 0x300.

```
7.  
8. .DEF reg1 = r16  
9. .DEF reg2 = r17  
10. .DEF cont = r18  
11. .DEF aux = r19  
12.  
13. ldi reg1,0x03 ; leitura do byte  
14. ldi cont,0x00 ; inicializa contador  
15. mov reg2,reg1  
16. ldi aux,0x01  
17. and reg2,aux  
18. eor reg2,aux  
19. cpi reg2,0x00  
20. brne p2  
21. inc cont  
22. p2: mov reg2,reg1  
23. ldi aux,0x02  
24. and reg2,aux  
25. eor reg2,aux  
26. cpi reg2,0x00  
27. brne p3  
28. inc cont  
29. p3: break  
...
```

**PROSSEGUIR**