

Lista de Exercícios I (Resoluções) - Monitoria

Arquitetura de Computadores

Resoluções dos exercícios da Lista de Exercícios I utilizando o simulador SimuS.

Exercício 1

```
1  ORG 0
2  INICIO:
3      LDA #15          ; carga imediata do primeiro número
4      STA NUM1         ; armazena NUM1
5      LDA #27          ; carga imediata do segundo número
6      STA NUM2         ; armazena NUM2
7      LDA NUM1         ; Carrega primeiro número (modo direto)
8      ADD NUM2         ; Soma com segundo número (modo direto)
9      STA RESULTADO    ; Armazena resultado (modo direto)
10     HLT              ; Para o programa
11
12 ; Dados
13 NUM1:   DS 1         ; Primeiro número (inicializado no inicio)
14 NUM2:   DS 1         ; Segundo número (inicializado no inicio)
15 RESULTADO: DS 1      ; Resultado da soma
16
17 END 0
```

Exercício 2

```
1  ORG 0
2  INICIO:
3      LDA #255         ; Carrega 255 (modo imediato - 8 bits)
4      ADD #1           ; Soma 1 para gerar carry (overflow)
5
6      LDA #100         ; carga imediata do primeiro número
7      STA NUM1         ; armazena NUM1
8      LDA #20          ; carga imediata do segundo número
9      STA NUM2         ; armazena NUM2
10     LDA NUM1         ; Carrega primeiro número (modo direto)
11     SBC NUM2         ; Subtrai NUM2 com carry (modo direto)
12     STA RESULTADO    ; Armazena resultado (modo direto)
13     HLT              ; Para o programa
14
15 ; Dados
16 NUM1:   DS 1         ; Primeiro número
17 NUM2:   DS 1         ; Número a ser subtraído
18 RESULTADO: DS 1      ; Resultado da subtração com carry
19
20 END 0
```

Exercício 3

```
1  ORG 0
2  INICIO:
3      LDA #42          ; carga imediata do número a ser testado
4      STA NUMERO       ; armazena o número em NUMERO
5      LDA NUMERO       ; carrega o número (modo direto)
6      AND #1          ; Testa o bit menos significativo (modo imediato)
7      JZ PAR          ; Se zero, é par (desvio condicional)
8
9      ; É ímpar
10     LDA #0           ; Carrega 0 para ímpar (modo imediato)
11     STA RESULTADO    ; Armazena resultado (modo direto)
12     JMP FIM         ; Pula para o fim (desvio incondicional)
13
14 PAR:
15     LDA #1           ; Carrega 1 para par (modo imediato)
```

```

16     STA RESULTADO    ; Armazena resultado (modo direto)
17
18 FIM:
19     HLT              ; Para o programa
20
21 ; Dados
22 NUMERO:    DS 1
23 RESULTADO: DS 1      ; 1=par, 0=ímpar
24
25 END 0

```

Exercício 4

```

1  ORG 0
2  INICIO:
3      LDA #1          ; Inicializa contador com 1 (modo imediato)
4      STA CONTADOR    ; Armazena contador (modo direto)
5
6  LOOP:
7      LDA CONTADOR    ; Carrega contador atual (modo direto)
8      SUB #5          ; Subtrai 5 (modo imediato)
9      JN CONTINUA     ; Se negativo, continua (contador < 5)
10     JZ FIM          ; Se zero, termina (contador = 5)
11     JMP FIM         ; Se positivo, termina (contador > 5)
12
13 CONTINUA:
14     LDA CONTADOR    ; Carrega contador atual (modo direto)
15     ADD #1          ; Incrementa contador (modo imediato)
16     STA CONTADOR    ; Armazena novo valor (modo direto)
17     JMP LOOP        ; Volta para o loop (desvio incondicional)
18
19 FIM:
20     HLT              ; Para o programa
21
22 ; Dados
23 CONTADOR: DS 1      ; Contador atual
24
25 END 0

```

Exercício 5

```

1  ORG 0
2  INICIO:
3      LDA #123        ; carga imediata do valor a ser copiado
4      STA VALOR_ORIGEM ; armazena em VALOR_ORIGEM
5      LDA #VALOR_ORIGEM ; carrega o endereço de VALOR_ORIGEM
6      STA PONTEIRO     ; armazena o ponteiro
7      LDA @PONTEIRO    ; carrega indiretamente o valor apontado
8      STA DESTINO      ; armazena no destino (modo direto)
9      HLT              ; Para o programa
10
11 ; Dados
12 VALOR_ORIGEM: DS 1
13 PONTEIRO:     DS 1
14 DESTINO:      DS 1      ; Destino da cópia
15
16 END 0
17

```

Exercício 6

```

1  ORG 0
2  INICIO:
3      LDA #25          ; carga imediata do número a ser multiplicado
4      STA NUMERO
5      LDA NUMERO       ; Carrega o número (modo direto)
6      SHL              ; Desloca 1 bit à esquerda (multiplica por 2)
7      STA RESULTADO    ; Armazena resultado (modo direto)
8      HLT              ; Para o programa
9
10 ; Dados
11 NUMERO:    DS 1      ; Número a ser multiplicado por 2
12 RESULTADO: DS 1      ; Resultado (deveria ser 50)
13

```

14 | END 0

Exercício 7

```
1  ORG 0
2  INICIO:
3      LDA #170      ; carga imediata do número original (10101010b)
4      STA NUMERO
5      LDA #15       ; carga imediata da máscara (00001111b)
6      STA MASCARA
7      LDA NUMERO    ; Carrega o número (modo direto)
8      AND MASCARA   ; Aplica máscara bit a bit (modo direto)
9      STA RESULTADO ; Armazena resultado (modo direto)
10     HLT           ; Para o programa
11
12 ; Dados
13 NUMERO:    DS 1
14 MASCARA:   DS 1
15 RESULTADO: DS 1 ; Resultado da operação AND (10101010 AND 00001111 = 00001010 ou 10 em decimal)
16
17 END 0
```

Exercício 8

```
1  ORG 0
2  INICIO:
3      LDA #85       ; carga imediata do número original (01010101)
4      STA NUMERO
5      LDA #170      ; carga imediata da máscara XOR (10101010)
6      STA MASCARA_XOR
7      LDA NUMERO    ; Carrega o número (modo direto)
8      XOR MASCARA_XOR ; Aplica XOR com a máscara (modo direto)
9      STA RESULTADO ; Armazena resultado (modo direto)
10     HLT           ; Para o programa
11
12 ; Dados
13 NUMERO:    DS 1 ; Número original
14 MASCARA_XOR: DS 1 ; Máscara XOR
15 RESULTADO: DS 1 ; Resultado do XOR (deve ser 11111111 ou 255 em decimal)
16
17 END 0
```

Exercício 9

```
1  ORG 0
2  INICIO:
3      LDA #45       ; carga imediata do primeiro valor
4      STA VALOR_A
5      LDA VALOR_A    ; Carrega VALOR_A (modo direto)
6      PUSH          ; Empilha VALOR_A
7
8      LDA #35       ; carga imediata do segundo valor
9      STA VALOR_B
10     LDA VALOR_B    ; Carrega VALOR_B (modo direto)
11     STA TEMP       ; Salva VALOR_B temporariamente
12
13     POP            ; Desempilha VALOR_A para o acumulador
14     ADD TEMP       ; Soma com VALOR_B
15     STA RESULTADO  ; Armazena resultado (modo direto)
16     HLT           ; Para o programa
17
18 ; Dados
19 VALOR_A:    DS 1 ; Primeiro valor
20 VALOR_B:    DS 1 ; Segundo valor
21 TEMP:       DS 1 ; Variável temporária
22 RESULTADO: DS 1 ; Resultado da soma
23
24 END 0
```