

```
1  .text
2  main:
3  addi $s0,$zero,1 # a=$s0
4  addi $s1,$zero,1 # b=$s1
5  addi $s2,$zero,1 # c=$s2
6  addi $s3,$zero,1 # d=$s3
7  addi $s4,$zero,1 # e=$s4
8  addi $s5,$zero,1 # f=$s5
9  #a)
10 sub $s0,$s1,$s2
11 #b)
12 add $s1,$s0,$s2
13 #c)
14 add $t0,$s0,$s1
15 sub $s3,$t0,$s2
16 #d)
17 add $t0,$s0,$s1
18 sub $s5,$t0,$s3
19 #e)
20 add $t0,$s1,$s3
21 sub $s2,$s0,$t0
22 #f)
23 sub $t0,$s1,$s2
24 sub $s4,$s0,$t0
25 #g)
26 sub $t0,$s1,$s2
27 sub $t1,$s0,$t0
28 add $s4,$t1,$s5
29 #h)
30 sub $t0,$s0,$s1
31 sub $t1,$s1,$s2
32 add $t2,$t0,$t1
33 sub $s5,$s4,$t2
34 exit:
35     li $v0, 10      # Código de saída do programa
36     syscall
```

6lista4.asm

7lista4.asm

8lista4.asm

9lista4.asm

```
1  .data
2  a: .word 0 10 20 30 40 50 60 70 80 90 100
3  .text
4  main:
5  la $s3,a           # $s3 = endereço de A
6  li $s2,50          # $s2 => h = 50
7  lw $t0,32($s3)     # carrega A[8] em $t0
8  add $t1,$s2,$t0    # h + A[8] = $t1
9  sw $t1,36($s3)     # carrega do valor de $t1 em A[9]
10 exit:
11     li $v0, 10      # Código de saída do programa
12     syscall
```

6lista4.asm

7lista4.asm

8lista4.asm

9lista4.asm

```
1  .data
2  a: .word 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
3  .text
4  main:
5  la $s3, a           # $s3 = endereço de A
6  li $s1, 0           # $s1 => g = 0
7  li $s2, 50          # $s2 => h = 50
8  li $s4, 2           # $s4 => i = 2
9  mul $t0, $s4, 4      # i*4 = 8
10 add $t1, $s3, $t0    # endereço de A + (i*4) = $t1
11 lw $t2, 0($t1)       # Carrega o valor de A[i] em $t2
12 add $s1, $s2, $t2    # g = h + A[i]
13 exit:
14     li $v0, 10        # Código de saída do programa
15     syscall
```

6lista4.asm

7lista4.asm

8lista4.asm

9lista4.asm

```
1  .data
2  A: .space 44      # Reservando espaço para um array de 11 inteiros (11 * 4 bytes = 44 bytes)
3  i: .word 0        # Variável para armazenar o índice do loop
4  .text
5  main:
6      li $t0, 0      # $t0 = i (índice do loop)
7      li $t1, 11     # $t1 = Tamanho do array A
8      li $t2, 50     # $t2 = Valor a ser adicionado (50)
9  loop:
10     bge $t0, $t1, exit # Verifica se i >= 11 (condição de saída do loop)
11                             # Calcula o endereço do elemento A[i]
12     mul $t3, $t0, 4   # Multiplica i por 4 (tamanho de um inteiro em bytes)
13     la $t4, A         # Carrega o endereço base do array A em $t4
14     add $t4, $t4, $t3  # Adiciona o deslocamento ao endereço base para acessar A[i]
15     lw $t5, 0($t4)    # Carrega o valor atual de A[i] em $t5
16     add $t5, $t5, $t2  # Adiciona 50 ao valor atual de A[i]
17     sw $t5, 0($t4)    # Armazena o novo valor de A[i] de volta em A[i]
18
19     addi $t0, $t0, 1  # Incrementa i
20
21     j loop            # Volta para o início do loop
22 exit:
23     li $v0, 10        # Código de saída do programa
24     syscall
25
```