

ASSIGNMENT Nº 1

Breno de Castro Pimenta, RA: 2017114809

06/09/2019

Problem 1: Evens and Odds

Programa que confere se o número é par ou ímpar.

Caso seja Par retorna 0.

Caso seja Ímpar retorna 1.

```
1  .data
2    input:.word 2227
3
4  .text
5  main:
6    la x6, input           #Armazenando endereco do input
7    lw x5, 0(x6)           #Armazenando valor do input
8    andi x10, x5, 1        #Buscando o bit menos significativo
9    addi a1, x10, 0        #Armazenando resposta em a1
10   addi a0, x0, 1         #Transformando ecal na saida
11   ecall                 #Imprimindo resultado
12   jal zero, end          #Voltando para a ordem de execucao
13
14 end:
```

Problem 2: Factorial

O programa retorna o fatorial de um dado número.

```
1  .data
2    input : .word 4
3
4  .text
5  main:
6    addi sp, sp, -16      #Alocando espaco na pilha.
7    sw x5, 0(sp)          # Salvando valores de
8    sw x6, 4(sp)          # registradores que serao utilizados.
9    sw x7, 8(sp)
10   sw x1, 12(sp)
11
12   la x6, input           #Armazenando endereco do input.
13   lw x5, 0(x6)           #Armazenando valor do input.
14   add x10, x5, x0        #Armazenando em x10 o input.
15   addi x7, x0, 1         #Colocando x7 com valor de 1.
16
17   jal x1, FACT          #Inicia-se o fatorial.
18
19   addi a1, x10, 0        #Armazenando resposta em a1.
20   addi a0, x0, 1         #Transformando ecal na saida.
21   ecall                 #Imprimindo resultado.
22
23   lw x5, 0(sp)           #Restaurando os valores dos
24   lw x6, 4(sp)           #registradores utilizados.
25   lw x7, 8(sp)
26   lw x1, 12(sp)
27   addi sp, sp, 16        #Desalocando os espacos na pilha.
28   jal zero, end         #Voltando para a ordem de execucao.
29
30  FACT:
31    mul x10, x10, x7       #Multiplicamos x7 pelo resultado.
32    addi x7, x7, 1        #Aumentamos o x7.
33    blt x7, x5, FACT      #Se x7 >= x5 paramos a conta.
34    jalr x0, 0(x1)        #Volta para a main.
35
36  end:
```

Problem 3: Permutation

O programa verifica se o um vetor é a permutacao de outro.

Caso seja retorna 1, caso contrario 0.

```
1  .data
2  inputVetorOrdenado: .word 1,2,3,4      # Usuario input vetor ordenado
3  inputVetorDesordenado: .word 4,2,3,1    # Usuario input vetor desordenado
4  inputTamanhoVetor: .word 4              # Usuario input tamanho vetor
5
6  .text
7  main:
8      addi sp, sp, -32                    #Alocando espaco na pilha
9      sw x5, 0(sp)                        # Salvando valor de registrador que sera usado
10     sw x6, 4(sp)
11     sw x7, 8(sp)
12     sw x18, 12(sp)
13     sw x28, 16(sp)
14     sw x29, 20(sp)
15     sw x30, 24(sp)
16     sw x31, 28(sp)
17
18     la x5, inputTamanhoVetor            #Armazenando endereco do inputTamanhoVetor
19     lw x5, 0(x5)                         #Armazenando valor do inputTamanhoVetor
20     la x6, inputVetorOrdenado            #Armazenando endereco do inputVetorOrd.
21     la x7, inputVetorDesordenado         #Armazenando endereco do inputVetorDesor.
22
23     addi x28, x0, -1                     #Reinicializando x28 como -1.
24     addi x10, x0, 1                     #Inicializando x10 como 1.
25     addi x6, x6, -4                     #Vetor ordenado -1 posicao
26     beq x0, x0, comparando              #Iniciando a comparacao
27
28
29  comparando:
30     addi x28, x28, 1                     #Iterando de 1 em 1
31     beq x28, x5, final                   #VerificandoSeTerminou
32     addi x6, x6, 4                       #Proxima posicao no vetor
33     lw x29, 0(x6)                        #Valor do vetor ordenado
34     addi x30, x7, -4                     #Reinicializando x30 uma posicao antes
35     add x18, x0, x0                      #Reinicializando x18 em 0.
36     beq x0, x0, comparacaoInterna
37
38  comparacaoInterna:
39     addi x30, x30, 4                     #Somando um na posicao do vetor Desordenado
40     lw x31, 0(x30)                       #Valor do vetor desordenado
41     beq x31, x29, comparando             #Compara se sao iguais
42     addi x18, x18, 1                     #Iterando 1
```

```

43  beq x18, x5, naoOrdenado      #Comparou com todo o vetor e terminou
44  bne x18, x5, comparacaoInterna  #Continua comparando
45
46  naoOrdenado:
47  add x10, x0, x0              #Atribuindo valor zero por nao estar ordenado
48  beq x0, x0, final            #Pulando para o final
49
50  final:
51  addi a1, x10, 0              #Armazenando resposta em a1
52  addi a0, x0, 1               #Transformando ecal na saida
53  ecall                       #Imprimindo resultado
54
55  lw x5, 0(sp)                 #Restaurando valor de registrador utilizado
56  lw x6, 4(sp)
57  lw x7, 8(sp)
58  lw x18, 12(sp)
59  lw x28, 16(sp)
60  lw x29, 20(sp)
61  lw x30, 24(sp)
62  lw x31, 28(sp)
63  addi sp, sp, 32              #Desalocando espaco na pilha
64  jal zero, end                #Voltando para a ordem de execucao
65
66  end:

```