

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACOM - FACULDADE DE COMPUTAÇÃO
SISTEMA DE INFORMAÇÃO

LUCAS ALBINO MARTINS

Matrícula: 12011ECP022

TRABALHO 03: Exercício de programação 2.

Disciplina: Arquitetura e Organização de Computadores.

Uberlândia
2020

Escreva um programa que leia uma palavra de 8 bits da memória, compute seu bit de paridade e então informe o usuário qual o bit de paridade computado.

```
# transformaparidade.asm
```

```
#
```

```
#
```

```
# Aluno: Lucas Albino Martins, Matricula 12011ECP022
```

```
# DESC: Usuário entra com uma letra e o programa devolve o valor binario de paridade.
```

```
#
```

```
#
```

```
#DDA 31.08.2020
```

```
#####  
#####
```

```
.data
```

```
str1:      .ascii "\n Entre com a letra: "
```

```
str2:      .ascii "\n valor de paridade eh: "
```

```
letra:     .space 2
```

```
.text
```

```
.globl entrada
```

```
entrada:
```

```
#      Usuário entra com a letra.
```

```
    la $a0, str1 # imprime a string para o usuário digitar a letra
```

```
    li $v0, 4
```

```
    syscall
```

```
#      programa recebe a letra.
```

```

    la $a0,letra # lê a letra digitada
    li $a1,2
    li $v0,8
    syscall

# Programa imprime a paridade.
    la $a0, str2 # imprime a saída
    li $v0,4
    syscall

    lb $a0,letra # $a0 = recebe a letra
    jal paridade
    li $v0,10 # finaliza o programa
    syscall

# paridade recebe a letra em $a0 e imprime seu valor binario
paridade:
    sub $sp,$sp,4
    sw $ra,0($sp)
    jal contador # salta e retorna a contagem em $v0
    and $t0,$v0,1
    beqz $t0,saiparidade # if even, skip (no action)
    or $a0,$a0,0x80 # se não a paridade = 1
saiparidade:
    jal imprime
    lw $ra,0($sp)
    add $sp,$sp,4
    jr $ra

# Inicia o contador com o a letra em $a0
contador:
    li $v0,0 # $v0 = 0
    li $t1,0x80 # $t1 = 0x80
contagem:

```

```

        and $t0,$a0,$t1
        beqz $t0,saidacontagem
        addu $v0,$v0,1
# sai do loop se a mascara for = 0
saidacontagem:
        srl $t1,$t1,1
        bnez $t1,contagem
        jr $ra

# imprime o valor binario de paridade.

imprime:
        move $t0,$a0
        li $t1,0x80

loop:
        and $t2,$t0,$t1
        beqz $t2,zero
        li $a0,1 # imprime 1
        b sair
zero:
        li $a0,0 # imprime 0

# sai do loop quando a mascara e 0
sair:
        li $v0,1
        syscall
        srl $t1,$t1,1
        bnez $t1,loop
        jr $ra

```