



Graduação Presencial Sincrona ...

[Página inicial](#)

[Teams](#)

[Tarefas](#)

[Fóruns](#)

[Notas](#)

[Biblioteca PUC Minas](#)

[Arquivos](#)

[Programa](#)

[Módulos](#)

[Colaborações](#)

[Office 365](#)

[Medalhas](#)

[Lucid \(Quadro branco\)](#)

[Pesquisa inteligente](#)

[Avaliação CPA](#)

[PUC Carreiras](#)



Exercício 03

Vencimento

6 de out de 2024 por 23:59

Pontos

10

Enviando

um upload de arquivo

Tipos de arquivo

pdf

Disponível

1 de ago de 2024 em 23:59 - 10 de out de 2024 em 23:59

Esta tarefa foi travada 10 de out de 2024 em 23:59.

Implemente os seguintes programas, e faça a entrega de um PDF como de costume:

Programa 10

```
# considere x mapeado em s1
.text
.globl _start
_start:
addi s1, zero, 1 # x = 1
.data
x1: .word 15
x2: .word 25
x3: .word 13
x4: .word 17
```

Observe o que acontece no MARS (ou outro simulador) na memória. Verifique por exemplo, em quais endereços o valor será armazenado. Coloque um print da tela que apresenta os dados no simulador.

Programa 11

Complete o programa anterior de maneira a ler os valores armazenados em x1, x2, x3 e x4 em registradores. Utilize as instruções de LOAD aprendidas em sala. Utilize também registradores para armazenar os endereços identificados no exercício anterior (caso o valor seja muito grande, utilize shifts para lhe auxiliar).

Programa 12

Considere os dados a seguir:

```
.data
x1: .word 15
x2: .word 25
x3: .word 13
x4: .word 17
soma: .word -1
```

Escreva um programa que leia todos os números acima, e em seguida, calcule e armazene a soma desses números em um registrador. Ao final, o valor desse registrador deverá ser gravado na posição de memória representado por "soma" nos dados acima.

Programa 13

Considere a seguinte expressão: $y = 127x - 65z + 1$

Faça um programa que calcule o valor de y, conhecendo os valores de x e z. Os valores de x e z estão armazenados na memória e na posição imediatamente posterior desses valores, deverá ser escrito o valor de y.

Ou seja:

```
.data
x: .word 5
z: .word 7
y: .word 0 # esse valor deverá ser sobrescrito após a execução do programa
```

Programa 14

Considere o trecho de código a seguir, e compile esse código em assembly:

```
int A[] = {1,3,5,7,9};
int B[] = {2,4,6,8,10};

A[0] = B[0] * 1 + A[0];
A[1] = B[1] * 2 + A[1];
A[2] = B[2] * 3 + A[2];
A[3] = B[3] * 4 + A[3];
A[4] = B[4] * 5 + A[4];
```

- Regra: Usar apenas um registrador para o endereço de A e outro para o endereço de B.

Programa 15

Substitua o código abaixo em assembly. As variáveis x, y e m devem ser armazenadas nos registradores s0, s1 e s2 respectivamente.

```
int x = um_valor_inteiro_qualquer;
int y = um_valor_inteiro_qualquer;
int m = 0;
if (x > y)
    m = x;
else
    m = y;
```

Programa 16

Compile o código abaixo em assembly. Use qualquer mapeamento de variáveis em registradores, seguindo a convenção.

```
int a = um_valor_inteiro_qualquer;
int b = um_valor_inteiro_qualquer;
int x = 0;
if ( a >= 0 && b <= 50 )
    x = 1;
```

Programa 17

Estude, com auxílio da internet e livros, como funciona um trecho de código do tipo switch/case. Verifique também como pode ser feito para este trecho de código ser compilado em assembly. Após os estudos, apresente um trecho de código usando switch/case em C, e seu correspondente em assembly.

Programa 18

Substitua o código abaixo em assembly. As variáveis temp e x devem ser armazenadas nos registradores s0 e s1 respectivamente.

```
switch(temp){
    case 10:
        x = 10;
        break;
    case 25:
        x = 25;
        break;
    default:
        x = 0;
}
```

Programa 19

Substitua o código abaixo em assembly. As variáveis i e x devem ser armazenadas nos registradores s0 e s1 respectivamente.

```
while(i == 8){
    x = i++;
}
```

Programa 20

Compile para o assembly do RISC-V o código a seguir:

```
int i;
int A[10];
for (i=0; i<10; i++) {
    A[i]=A[i]+1;
}
```

Programa 21

Compile para o assembly do RISC-V o código a seguir:

```
int i;
int A[10];
for (i=0; i<10; i++) {
    if (i%2==0)
        A[i]=A[i]+A[i+1];
    else
        A[i]=A[i]*2;
}
```

Programa 22

A série de Fibonacci é definida assim: os dois primeiros termos têm o valor 1 e cada termo seguinte é igual à soma dos dois anteriores.

1 1 2 3 5 8 13 21 ...

Escreva um programa em linguagem de montagem, que calcule o valor do 9º termo.

Envio

✓ Enviado!

6 de out de 2024 em 22:29

[Detalhes do envio](#)

[Baixar Atividade 3.pdf](#)

Nota: 10 (10 pts possíveis)

Avaliado anonimamente: não

Comentários:
Nenhum comentário