

Lista – Aula prática: Assembly MIPS

Faça cada programa em um arquivo separado, com nome dado por LXQY.asm (Lista/Questão), por exemplo: L1Q1.asm, L1Q2.asm, etc.

Atenção: o código de todas as questões deverá estar claramente comentado, pois caso contrário a correção será muito dificultada. É de seu interesse colaborar com a correção!

Obs: Todos os exercícios deverão ser entregues até a data marcada no calendário.

1. Codifique um programa correspondente ao seguinte pseudo-código:

```
int a = 2;
int b = 1;
int m = 0;
m = a;
if ( b < m )
    m = b;
else
    m = 0;
```

2. Codifique um programa correspondente a:

```
int a = ...;
int b = ...;
x = 0;

if ( a >= 0 && a < b )
    x = 1;
else if( a < 0 && a > b )
    x = 2;
else
    x = 3;
```

3. Faça um programa em linguagem de montagem Mips que receba como entrada uma string com n caracteres e gere como saída uma nova string com a inversão da ordem dos caracteres. Essa nova string também terá a troca das letras maiúsculas por minúsculas e vice-versa. Por exemplo: se a entrada for HArDwArE a saída deverá ser eRaWDRah. A entrada deve ser lida da memória e a saída deve ser escrita na memória. Caso a string possua algum caractere que não seja letra o valor 1 deverá ser armazenado no registrador v1 e o programa deverá ser encerrado.

Lembramos que o fim da string é dado pelo caractere nulo e que para manipular string nessa questão recomendamos que sejam usadas variáveis do tipo ASCII.

Atenção: os caracteres deverão ser armazenados em sequência na memória.

4. Implemente (recursivamente) na linguagem de montagem do MIPS, uma função que receba dois números a e b ($b \geq a$) e retorne a soma de todos os números contidos no intervalo $[a,b]$. Se $a > b$, o valor 1 deverá ser armazenado no registrador $v1$ e o programa encerrado.

5. Escreva um programa em linguagem de montagem do MIPS que receba dois números inteiros armazenados na memória e realiza a multiplicação dos dois números. Considere números positivos e negativos. A instrução `mult` não deverá ser utilizada na implementação dessa questão. O resultado deverá ser armazenado em uma variável na memória.

6. Implemente um programa que recebe um inteiro como parâmetro e retorna número de Fibonacci do mesmo. Use recursão.

Boa Sorte!