



## 1COP029 - Lista de Exercícios 18

1. Exercício Teórico: Traduza o código a seguir para assembly de MIPS. Durante a tradução para assembly, as variáveis locais devem ser mapeadas para registradores \$s0, \$s1, .... A função int value() retorna um valor no registrador \$v0 e não tem o seu código apresentado, bastando apenas ser chamada quando necessário.

```
int value(); //prototipo da funcao value
int func()
{
    int a,b,c;
    int r = -1;
    while(r == -1)
        a = value();
        b = value();
        c = value();
        if((a<b+c) && (b<a+c) && (c<a+b))
        {
            if((a==b)&&(b==c))
            {
                r = 0;
            }
            else
                 if((a==b)||(a==c)||(c==b))
                 {
                     r = 1;
                }
                else
                 {
                     r = 2;
            }
    return r;
}
```





2. Enter Exercício Prático: Considere o seguinte trecho de código C:

```
int potencia(int base, int expoente)
{
    if (expoente == 0)
    {
        return 1;
    }
    else
        return base * potencia(base,expoente-1);
    }
}
int main()
    int base;
    int expoente;
    printf("Entre com a base:");scanf("%d",&base);
    printf("Entre com o expoente:");scanf("%d",&expoente);
    printf("Resultado: %d\n",potencia(base,expoente));
    return 0;
}
```

Traduza o código apresentado para assembly de MIPS e execute o mesmo em um simulador para verificar que o mesmo funciona corretamente. Durante a tradução para assembly, as variáveis locais devem ser mapeadas para os registradores \$s0, \$s1, .... A função int potencia(...) retorna um valor no registrador \$v0. Lembre-se de salvar/restaurar os devidos valores na pilha. Para os comandos printf e scanf utilize as syscalls presentes nos simuladores.