

Os exercícios foram adaptados do livro

Patterson, David A. Hennessy, John L. Organização e Projeto de Computadores. Disponível em: Minha Biblioteca, (5a. edição). Grupo GEN, 2017.

1 <§2.8> Traduza o código de alto nível para código assembly do MIPS. Suponha que os valores dos argumentos estejam em \$a0, \$a1, \$a2 e \$a3. Além disso, utilize os registradores \$s0, ..., \$s7 para armazenar as variáveis internas.

a)

```
int soma(int x, int y) {  
    return x+y;  
}
```

b)

```
int soma(int x, int y) {  
    int sum = x+y;  
    if (sum>0)  
        return sum;  
    else  
        return sum + sum;  
}
```

c)

```
int soma(int x, int y, int vet[]) {  
    int sum = x+y;  
    if (sum>0)  
        return vet[sum];  
    else  
        return vet[0];  
}
```

d)

```
int somavet(int vet[], int n) {  
    int ac = 0;  
    for (int i=0; i<n; i++)  
        ac += vet[i];  
    return ac;  
}
```

e)

```
int maior3(int x, int y, int z){
    int maior = x;
    if( y > maior)
        maior = y;
    if(z > maior)
        maior = z;
    return maior;
}
```

f)

```
int fatorial(int x){
    int y = x;
    int fat = x;
    if(y == 0)
        return 1;
    while(y != 0){
        if(y == 1)
            return fat;
        fat = fat * (y-1);
        y = y-1;
    }
    return fat;
}
```

g)

```
int mediavet(int vet[], int n){
    int ac = 0;
    for (int i=0; i<n; i++)
        ac += vet[i];
    return ac/n;
}
```

h)

```
int subrevers(int x){
    int sub = x;
    int count = x;
    if(x == 0)
        return 0;
    while(count !=0){
        sub = sub - (count-1);
        count--;
    }
    return sub;
}
```