

# Exercícios de *Assembly*

*Prof. Iaçanã Ianiski Weber*



## 1. Encontrar um elemento no vetor *save*

- ✓ Assumir: *i* em *\$s3*, *k* em *\$s5*, *\*save* em *\$s6*
- ✓ Assumir a área de dados abaixo e completar com o que se pede
- ✓ Código C:

```
main() {
    int save[] = {0x901, 0x345, 0x879, 0x100, 0x900};
    int i = 0, k = 0x100, pos = 0;
    while(save[i] != k)
        i++;
    pos = i;
}
```

- ✓ Código MIPS:

```
.text
.globl main

main:
    <completar com zeramento de 'i', carga de k, e carga de *save >

Loop: sll    $t1,$s3,2    # $t1 = i * 4 : para endereçar save
      add    $t1,$t1,$s6  # $t1 = address of save[i]
      lw     $t0,0($t1)   # $t0 = save[i]
      beq    $t0,$s5,Exit # go to Exit if save[i] = k
      addi   $s3,$s3,1    # i = i + 1
      j      Loop        # go to Loop

Exit: <grave 'i' na posição pos>

fim: j      fim

.data

k:    .word 0x100
pos:  .word 0
save: .word 0x901 0x345 0x879 0x100 0x900
```

2. Implementar o código que segue para calcular **o maior valor em um vetor de naturais**, utilizando o *assembly* do MIPSs (usar apenas o subconjunto de instruções apresentado em aula) :

✓ Código C:

```
main() {
    int a[] = {0x123, 0x345, 0x879, 0x100, 0x090};
    int max = 0, n = 5;
    for(int i = 0; i < n; i++) {
        if (a[i] > max)
            max = a[i];
    }
}
```

- ✓ No final da execução, deve-se gravar o valor **max** no label **max**
- ✓ Utilizar a seguinte área de dados:

```
.data
max:    .word    0
n:      .word    5
a:      .word    0x123 0x345 0x879 0x100 0x090
```

3. Implementar o código que segue para calcular **divisão por subtrações sucessivas**, utilizando o *assembly* do MIPS (usar apenas o subconjunto de instruções apresentado em aula)

✓ Exemplo de divisão de 10 por 3 usando subtrações sucessivas:

Dividendo	Divisor	Dividendo - Divisor	Quociente
10	3	7	1
7	3	4	2
4	3	1	3
d1	d2	resto	q

✓ Utilizar a seguinte área de dados:

**.data**

**d1:**    **.word**   **0xFAAA**   # 64170   resposta: 489 (1E9)   resto 111 (6F)

**d2:**    **.word**   **0x83**     #131

**q:**     **.word**   **0**

**resto:** **.word**   **0**

4. Implementar o código que segue para calcular **a série de Fibonacci**, utilizando o *assembly* do MIPSs (usar apenas o subconjunto de instruções apresentado em aula):

✓ Assumir: **N** em **\$s0**, **\*vet** em **\$s1**, **cnt** em **\$s2**, **a** em **\$s3**, **b** em **\$s4**, **aux** em **\$s5**

✓ Assumir a área de dados abaixo e completar com o que se pede

✓ Código C:

```
int main(){
    int N = 15;
    int vet[15];
    int cnt = 0, a = 0, b = 1, aux;

    if (cnt==N) return(0);
    vet[cnt] = a;
    cnt++;

    if (cnt==N) return(0);
    vet[cnt] = b;
    cnt++;

    while(cnt != N) {
        aux = a + b;
        a = b;
        b = aux;

        vet[cnt] = b;
        cnt++;
    }
    return 0;
}
```

✓ Código MIPS:

```
.text
.globl main

main:
    <completar>

end: j    end

.data

N:    .word 15
vet:  .word 0x0
```

✓ Exemplo para N = 15:

```
0  1  1  2  3  5  8  13  21  34  55  89  144  233  377
0, 1, 1, 2, 3, 5, 8,  D, 15, 22, 37, 59,  90,  E9, 179
```