

## Laboratório 11

### Assembly: Chamada de Função e Estrutura

1. Crie o arquivo “calc.c” a seguir:

```
int sum(int x, int y, int z, int w);

int calc(int a, int b, int c, int d) {
    int i, j, k;
    i = sum(a+1, b, c+2, d);
    j = sum(b, d+1, a, c+2);
    k = sum(a-1, b, c-2, d);
    return i + j + k;
}
```

Crie o “main.c” para testar a função acima:

```
#include <stdio.h>

int calc(int a, int b, int c, int d);

int sum(int x, int y, int z, int w) {
    return x+y+z+w;
}

int main (void) {
    printf("%d\n", calc(1, 2, 3, 4));
    return 0;
}
```

Compile os dois arquivos juntos e depois execute “calc” para ver o resultado:

```
gcc -Wall -o calc calc.c main.c
```

2. Traduza a função *calc()* acima para Assembly. Salve a função no arquivo “calc.s”.  
Compile os dois arquivos juntos e depois execute “main\_calc” para ver o resultado:

```
gcc -Wall -o calc calc.s main.c
```

### 3. Traduza o programa de C para Assembly:

```
#include <stdio.h>

#define BLUE 1
#define RED  2

struct Circle {
    int  x;
    int  y;
    int  r;
    char filled;
    long color;
};

struct Circle c1;

int main() {
    c1.x = 10;
    c1.y = 20;
    c1.r = 2 * c1.y;
    c1.filled = 1;
    if (c1.r > 30)
        c1.color = RED;
    else
        c1.color = BLUE;
    printf("color: %ld\n", c1.color);
    return 0;
}
```

4. Traduza o programa de C para Assembly:

```
#include <stdio.h>

struct Placar {
    int  pos;
    int  neg;
};

struct Placar placar;
int nums[5] = {4,-9,3,-7,1};

int is_positivo(int a) {
    return (a > 0) ? 1 : 0;
}

int main() {
    int i;
    for (i = 0; i < 5; i++) {
        if (is_positivo(nums[i]))
            placar.pos++;
        else
            placar.neg++;
    }
    printf("Pos: %d\n", placar.pos);
    printf("Neg: %d\n", placar.neg);
    return 0;
}
```