1. \_somaVetor:

2. push ebp

3. mov ebp, esp

4. sub esp, 16

5. mov DWORD PTR [ebp-8], 0

6. mov DWORD PTR [ebp-4], 0

7. jmp L2

\_somaVetor:

**2 .push ebp =**

empilha o registrador de base da pilha antes de ser alterado

**3. mov ebp, esp =**

criação do stack frame

Stack frame = o "quadro da pilha" reservado para uma função.

esp = registrador do topo

ebp = registrador de base da pilha

ou seja quando fizemos mov ebp,esp fizemos os dois registradores apontarem pro mesmo lugar na memória.

**4. sub esp, 16 =**

O desloca 16 **bytes** o registrador de topo da pilha com o objetivo de reservar espaço para variáveis locais.

**5. mov DWORD PTR [ebp-8], 0 =**

DWORD é abreviação para double word, uma palavra de 4 bytes sendo uma WORD uma palavra de 2 bytes, então [ebp-8],0 estamos “zerando” a variável na posição [ebp-8] que corresponde ao acum=0 no main.c.

**6. mov DWORD PTR [ebp-4], 0 =**

A mesma ideia anterior porém com deslocamento 4 bytes acima [ebp-4], corresponde a j=0 no main.c.

**7. jmp L2 =**

Realiza um salto incondicional para a linha com rótulo L2, que corresponde ao for (j = 0; j < elementos; j++).