# Linguagem de montagem 6

**Desenvolvida por**: Raul A. Gonzalez Augusto, RA.: 211023698

# 2 soma

#include<stdio.h>

  short int v[1000];

  int main()  {

  asm(    // atribui o endereço de v para rbx

          "  lea rbx, %[v]      \n"

          // atribui 1234 para ax

          "  mov ax, 0        \n"

          // inicia contador

          "  mov cx, 10       \n"

          "repete:              \n"

          // soma o valor da posicao do vetor no ax

          "  add ax, word ptr [rbx]   \n"

          // incrementa o ponteiro para o próximo elemento do vetor

          "  add rbx, 2         \n"

          // decrementa o contador

          "  dec cx             \n"

          // volta para "repete" enquanto cx > 0

          "  jnz repete         \n"

          : [v] "=m"(v)

          :

          : "ax", "cx", "rbx");

  return 0;

}

# 3 menor

#include <stdio.h>

int main() {

  short int vetor[1000] = {2,4,5,6,8,24,12,4,2,-7,123,-12,13,4,75,-17,-3,9,6,1};

  short int menor=32766,i;

  for(i=20;i<1000;i++)

    vetor[i]=0;

  asm(" LEA RBX,%[vetor] \n"

    " MOV CX, 1000 \n"

    " MOV AX, %[menor] \n"

    " procura: \n"

    " CMP AX,WORD PTR [RBX] \n"

    " JLE menor \n"

    " MOV AX,WORD PTR [RBX] \n"

    " menor: \n"

    " ADD RBX, 2 \n"

    " DEC CX \n"

    " JNZ procura \n"

    " MOV %[menor], AX \n"

    : [menor] "+r"(menor),[vetor] "=m"(vetor)

    : "ax","cx","rbx");

  printf("%d\n", menor);

  return 0;

}

# 4 pilha

#include <stdio.h>

int main() {

  short int vetor[1000] = {2,4,5,6,8,24,12,4,2,-7,123,-12,13,4,75,-17,-3,9,6,1};

  short int menor=32766,i;

  for(i=20;i<1000;i++)

    vetor[i]=0;

  asm(" push ax \n"

    " push cx \n"

    " push rbx \n"

    " LEA RBX,%[vetor] \n"

    " MOV CX, 1000 \n"

    " MOV AX, %[menor] \n"

    " procura: \n"

    " CMP AX,WORD PTR [RBX] \n"

    " JLE menor \n"

    " MOV AX,WORD PTR [RBX] \n"

    " menor: \n"

    " ADD RBX, 2 \n"

    " DEC CX \n"

    " JNZ procura \n"

    " MOV %[menor], AX \n"

    " pop rbx \n"

    " pop cx \n"

    " pop ax \n"

    : [menor] "+r"(menor),[vetor] "=m"(vetor)

    : "ax","cx","rbx");

  printf("%d\n", menor);

  return 0;

}