Praticas 07a

O objetivo do programa eh ler n elementos dentro de um laço, soma-los, calcular a média simples inteira e encontrar o maior e menor elementos do conjunto. Primeiramente, o programa solicita o número de números a ser lido. Depois, o programa lê o primeiro número e o usa para inicializar as variaúeis maiornum e menornum. Então, o programa permanece em um laço de leitura de n números, usando a instrução loop, a qual trabalha consorciada com o registrador %ecx, o qual é decrementado de 1, automaticamente a cada giro do laço. Quando o %ecx atinge o valor zero (0) o laço é interrompido.

Dentro de cada iteração do laço, cada número lido é adicionado a somatória e depois é comparado com o maior número, podendo substituí-lo, e/ou com o menor número, podendo substituí-lo também.

Para gerar o executavel, gere primeiro o objeto executando o seguinte comando:

```
as praticas_07a.s -o praticas_07a.o
```

e depois link dinamicamente com o seguinte comando:

```
ld praticas_07a.o -l c -dynamic-linker /lib/ld-linux.so.2 -o praticas_07a
```

O executavel se chamara praticas_07a, sem extensão, e para executá-lo digite:

```
./praticas_07a
```

```
.section .data

titulo:   .asciz    "\n*** Programa Trata N Números Inteiros ***\n\n"

pedeN:    .asciz    "\nDigite o Numero de Elementos (N) => "

formato:  .asciz    "%d"

pedenum:  .asciz    "\nEntre com o elemento %d => "

mostra1:  .asciz    "\nElementos Lidos:"

mostra2:  .asciz    " %d"

mostra3:   .asciz      "\nSomatoria = %d    MediaInt = %d    Maior = %d
Menor = %d\n"

maiornum: .int       0

menornum: .int       0

N:    .int   0

num:  .int   0

soma: .int   0
```

```
.section .text


.globl _start
_start:

      pushl  $titulo
      call   printf


leN:
      pushl  $pedeN
      call   printf
      pushl  $N
      pushl  $formato
      call   scanf

      addl   $12, %esp

      movl   N, %ecx
      cmpl   $0, %ecx
      jle    fim

      addl   $16, %esp


lenum1:
      movl   $1, %ebx
      subl   $1, %ecx
      pushl  %ecx
      pushl  %ebx

      pushl      $pedenum
      call   printf
      pushl  $num
      pushl  $formato
      call   scanf

      addl   $12, %esp

      movl   num, %eax
      movl   %eax, soma
      movl   %eax, maiornum
      movl   %eax, menornum
      popl   %ebx
      popl   %ecx

      cmpl   $0, %ecx
      jle    mostratudo

leoutros:
      incl   %ebx
      pushl  %ecx
      pushl  %ebx

      pushl      $pedenum
      call   printf
      pushl  $num
```

```
      pushl  $formato
      call   scanf

      addl   $12, %esp

      popl   %ebx
      popl   %ecx
      movl   num, %eax
      addl   %eax, soma

      cmpl   maiornum, %eax
      jg     mudamaior

      cmpl   menornum, %eax
      JL     mudamenor

      loop   leoutros
      jmp    mostratudo

mudamaior:
      movl   %eax, maiornum
      loop   leoutros
      jmp    mostratudo

mudamenor:
      movl   %eax, menornum
      loop   leoutros

mostratudo:

      movl   menornum, %eax
      pushl  %eax
      movl   maiornum, %eax
      pushl  %eax
      movl    soma, %eax
      divl   N, %eax
      pushl  %eax
      movl    soma, %eax
      pushl  %eax

      pushl  $mostra3
      call   printf

      addl   $20, %esp

fim:
      pushl  $0
      call   exit
```

DESAFIO: Ao invés de ler os N elementos do teclado, obtenha-os a partir de um vetor já pré-definido em memória. Declare e inicialize um vetor de uma quantidade máxima de elementos inteiros (positivos e negativos). Certifique-se de que N seja <= a essa quantidade máxima. Depois que N for conhecido, percorra os N primeiros elementos do vetor e resolva o que está sendo solicitado.