

Exercício de programação 1

Nome: _____ Resolução: _____ Turma: _____

Escrever um fragmento de código *assembly* para somar alguns elementos da sequência SEQ de 32 números inteiros. Os elementos a somar são selecionados pelo valor de MSK (do tipo DWORD): para cada bit igual a 1, o correspondente elemento de SEQ deve ser incluído na soma. Além disso, apenas valores positivos devem ser incluídos.

Exemplo para a sequência $\{-2, 2, -4, 3, 9, 1, \dots, 1\}$ com MSK=25 (decimal):

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1

Neste caso apenas os elementos das posições 3 e 4 da sequência devem ser incluídos na soma ($3+9=12$); o valor da posição 0 não é incluído, por ser negativo.

O fragmento deve guardar o resultado em RES (sem o imprimir). Assumir que o valor da soma pode ser representado em 32 bits.

```
include mpcp.inc
        .data
SEQ      SDWORD  -2, 2, -4, 3, 9, ... ; 32 valores
MSK      DWORD   25
RES      SDWORD  ?

        .code
main:    xor     eax, eax                ; para guardar a soma
        mov     edi, offset SEQ        ; endereço-base da sequência
        mov     ecx, MSK

ciclo:   jecxz   fim                    ; terminar se máscara só tiver zeros
        shr     ecx, 1                  ; bit menos significativo passa para CF
        jnc     cont                    ; não somar se bit == 0
        cmp     sdword ptr [edi], 0
        jle     cont                    ; não somar, se menor ou igual a zero
        add     eax, [edi]
cont:    add     edi, 4                  ; endereço do elemento seguinte
        jmp     ciclo
fim:     mov     RES, eax                ; enunciado pede para guardar resultado em RES
```