

Trabalho Prático 1 - Simulador NEANDER

"[The Euclidean algorithm] is the granddaddy of all algorithms, because it is the oldest nontrivial algorithm that has survived to the present day." Donald Knuth, The Art of Computer Programming, Vol. 2: Seminumerical Algorithms, 2nd edition (1981), p. 318.

Escrever um programa para o simulador Neander que calcule o maior divisor comum (mdc) para números inteiros em 8 bits. O cálculo do maior divisor comum pode ser feito pelo Algoritmo de Euclides, descrito a seguir em uma pseudo linguagem:

```
function mdc(a, b)
  while b ≠ 0
    t := b
    b := a mod b
    a := t
  return a
```

O maior dos dois números é sucessivamente dividido pelo menor e substituído pelo resto inteiro da divisão. Por exemplo, para calcular o maior divisor comum de 33 e 27, tem-se:

a	b
33	27
27	6
6	3
3	0

O resultado, quando **b** atinge o valor zero, está na variável **a**. Assim, pelos passos calculados acima, o maior divisor comum de (33,27) é 3. Por outro lado, calculando o maior divisor comum de 16 e 15, tem-se:

a	b
16	15
15	1
1	0

Ou seja, neste caso o maior divisor comum é 1. Observe-se que o maior divisor comum também é definido para números negativos, ou $\text{mdc}(33,27) = \text{mdc}(-33,27) = \text{mdc}(33,-27) = \text{mdc}(-33,-27) = 3$. De uma forma geral, $\text{mdc}(a,b) = \text{mdc}(|a|,|b|)$. Observe-se também que $\text{mdc}(a,0) = a$.

O programa Neander deve receber como entrada dois números inteiros de 8 bits, em complemento de dois, e calcular o maior divisor comum. Para o cálculo devem ser obrigatoriamente utilizadas as seguintes posições de memória:

- Posição 128 – entrada **a** (número inteiro em complemento de 2)
- Posição 129 – entrada **b** (número inteiro em complemento de 2)
- Posição 130 – $\text{mdc}(a,b)$

Para os valores de entrada podem ser assumido que serão diferentes de -128, ou seja, tanto **a** quanto **b** estarão sempre no intervalo [-127,+127].

Dicas e Observações:

- O Neander não possui instrução de subtração. Entretanto, uma subtração pode ser transformada em uma soma trocando-se o sinal do segundo operando.
- O Neander não possui instrução de multiplicação. Entretanto, uma multiplicação pode ser transformada em uma sequência de somas (e no trabalho nem é necessário realizar multiplicações).
- O Neander não possui instrução de divisão. Entretanto, uma divisão pode ser transformada em uma sequência de subtrações condicionais (subtrai-se o divisor do dividendo até que esse fique zero, e soma-se 1 ao quociente para cada subtração realizada). E no trabalho o valor do quociente nem é necessário.
- Para números inteiros positivos, o algoritmo de Euclides pode ser implementado sem utilizar divisões:

```
function gcd(a, b)
  while a ≠ b
    if a > b
      a := a - b
    else
      b := b - a
  return a
```

Os trabalhos serão corrigidos de forma automática, com **20** valores diferentes. Portanto, devem ser observadas rigorosamente as seguintes especificações:

- o código do programa deve iniciar no endereço 0 da memória.
- a primeira instrução executável deve estar no endereço 0.
- os endereços para os dois operandos e para o resultado devem ser exatamente os especificados acima, inclusive na ordem dos bytes.
- usar para variáveis adicionais ou para código extra os endereços de memória de 131 em diante.
- no cálculo, os valores de **a** (endereço 128) e **b** (endereço 129) não devem ser modificados.

O trabalho deverá ser entregue no Moodle, na área de “Entrega do Trabalho Neander”, na forma de um arquivo compactado (formato Zip ou Rar) composto por:

- um arquivo de memória do Neander, contendo o programa.
- um arquivo texto, com documentação e comentários. Lembre-se de incluir seu nome completo e seu número de cartão nas primeiras linhas deste arquivo.
- Para nomear os arquivos, utilize todo o seu nome, usando maiúsculas e minúsculas, sem espaços em branco e sem acentos. Assim, por exemplo, o aluno Um de Três Quatro deve denominar os seus arquivos de **UmDeTresQuatro.MEM**, **UmDeTresQuatro.TXT** e **UmDeTresQuatro.ZIP** (ou **RAR**).

Data de Entrega: 30/04/2015, via Moodle

Alguns casos de teste

Teste	A (End. 128)	B (End.129)	mdc (End. 130)
1	33	27	3
2	16	15	1
3	14	15	1
4	27	-3 (253)	3
5	-3 (253)	-9 (247)	3
6	42	0	42
7	0	0	0
8	121	33	11
9	25	125	25
10	25	110	5