

PCS-2302 / PCS-2024

Lab. de Fundamentos de Eng. de Computação

Aula 01 – Exercícios Turma 5

Introdução

Máquina de von Neumann

Professores:

Anarosa Alves Franco Brandão (PCS 2302)
Marcos A. Simplicio Junior (PCS 2302/2024)
Ricardo L. A. Rocha (PCS 2024)

Monitores: Diego Queiroz, Felipe Leno, Michel Bieleveld

Exercícios (1)

Entrega em sala de aula:

- TYGXXA01E01.zip → substitua Y, XX de acordo

Objetivo: Familiarização com o simulador e a linguagem da MVN
Execute os programas aula01_prog1.mvn, aula01_prog2.mvn e aula01_prog3.mvn. Gere os **mapas completos** da memória **antes** (mas após carregar programa na MVN) e **depois** de cada **execução**, salvando-os, respectivamente, nos arquivos aula01_progZ_a_gXX.txt e aula01_progZ_d_gXX.txt.

- Para isso, use o comando “m” do simulador

Onde,

Z é o número do programa

Y é o número da turma

XX é o número do grupo

Exercícios (2)

Entrega em sala de aula:

• TYGXXA01E02.mvn → substitua Y, XX de acordo

Desenvolva um programa que, dado um conjunto de 3 números, produza um outro conjunto em que estes números estejam organizados em ordem **crescente**. Use auto-modificação de código para tratar a lista de números e também uma sub-rotina **menor(A, B)**. O retorno da sub-rotina é feito por meio do acumulador, portanto, se $A < B$, o resultado no acumulador deve ser **ZERO (0000)**, e em caso contrário, deve ser **UM (0001)**.

Nos comentários do código documente as condições, premissas e limitações consideradas.

Endereço de início do programa principal: 0000

Endereço do conjunto original (entrada) : 0002..0008 (terminador: FFFF)

Endereço do conjunto crescente (saída) : 000A..0010 (terminador: FFFF)

Exercícios (3)

Entrega em sala de aula:

- TYGXXA01E03.mvn → substitua Y, XX de acordo

Desenvolva um programa que, dada uma lista de números inteiros positivos, realize a busca de um dado número para verificar a quantidade de ocorrências deste número na lista. Além da **auto-modificação**, desenvolva a sub-rotina **igual(A, B)**. O retorno da sub-rotina é feito por meio do acumulador, portanto, se **A = B**, o resultado no acumulador deve ser **ZERO (0000)**, e em caso contrário, deve ser **UM (0001)**.

IMPORTANTE: A lista fornecida tem como terminador de lista o valor FFFF.

Nos comentários do código documente as condições, premissas e limitações consideradas.

Endereço de início do programa principal: 0000

Endereço contendo quantidade de ocorrências (saída) : 0002

Endereço do número a ser buscado (entrada) : 0004

Endereço de início da lista (entrada) : 0006.. (terminador: FFFF)

Critério de correção

1. Usou os **nomes** pedidos para os programas
2. Respeitou os **endereços de memória** pedidos
3. Executou corretamente os seguintes **testes**:

Teste	Entrada (0002..0008)	Saída (000A..0010)
E2_1	0003, 0002, 0001, FFFF	0001, 0002, 0003, FFFF
E2_2	0002, 0001, 0001, FFFF	0001, 0001, 0002 , FFFF
E2_3	8001, 8002, 8003, FFFF	8001, 8002, 8003 , FFFF

Teste	Entrada (0004: {0006..})	Saída (0002)
E3_1	0001: {0001, 0002, 0003, 0002, 0003, 0002, FFFF}	0001
E3_2	0002: {0001, 0002, 0003, 0002, 0003, 0002, FFFF}	0003
E3_3	0004: {0001, 0002, 0003, 0002, 0003, 0002, FFFF}	0000
E3_4	0002: {FFFF}	0000