

Lista de Exercícios – Organização de Computadores Prof. Jorge Luiz e Silva

1. Fazer um programa utilizando a linguagem assembly que leia n dados com ECO, e armazene esses dados em um vetor de n posições.
Obs: - Apague a tela inicialmente,
- leia o n , (valores ASCII no máximo 9) transformando-o em valor numérico para ser armazenado no CX na contagem do loop.
Todo valor numérico em ASCII tem a configuração 3xh, basta tirar o 3 e ficar somente com o valor de x, ou seja, 0xh, implicando em máscara and 0fh.
2. Fazer um programa utilizando a linguagem assembly tal que leia um conjunto n de dados numéricos e os armazene esses dados em um vetor de n posições.
Obs: Apague a tela inicialmente, leia o n , e leia o conjunto de dados rejeitando aqueles que não são numéricos, convertendo-os no valor decimal correspondente.
3. Fazer um programa utilizando a linguagem assembly tal que leia um conjunto n de dados numéricos e encontre o maior valor entre eles, escrevendo o resultado em tela.
Obs: Apague a tela inicialmente, leia o n , leia o conjunto de dados rejeitando aqueles que não são numéricos, convertendo-os no valor decimal correspondente, encontre o maior entre eles, e

escreva o resultado com a frase: “Resultado final:” e o valor em seguida (valor esse reconvertido em ASCII).

4. Fazer um programa utilizando a linguagem assembly tal que leia 3 vetores numéricos de tamanho n e some esses 3 vetores, mostrando o resultado final da soma.

Obs: Crie rotina de leitura de vetor, apague a tela inicialmente, leia o n, leia o conjunto de dados rejeitando aqueles que não são numéricos, convertendo-os no valor decimal correspondente, realize a soma dos vetores (a soma não pode ultrapassar o valor 9 para poder transformá-lo novamente em ASCII para escrita final) e escreva o resultado com a frase: “Vetor final:” e o valor em seguida (valor esse convertido em ASCII).

Obs final:	parâmetro	função	dado
	AH – 1	ECO	AL
	AH – 2	ESCREVE	DL
	AH – 9	MENSAGEM	DX
	AH – 4CH	DOS	

code segment

 Assume cs:code,ds:data

inic:

code ends

data segment

data ends

end inic

Escrita de mensagem:

```
mov dx, offset men (string terminada por $)
mov ah,9
int 21h
```

Ex:

```
dados segment
```

```
men    db    `Valor da Variável`: 0dh,0ah,`$`
```

```
vet    dB    10 dup(?)
```

```
dados ends
```