

1. Desenvolva um programa capaz de ler n valores inteiros e os insira em um array. A leitura deve ser feita em uma única linha, separando os valores por espaço. A leitura deve ser encerrada quando um valor -100 é lido. Após a leitura ser realizada, o programa deve utilizar o procedimento **exibe_array** para exibir o array em uma única linha e entre colchetes.

Exemplos:

```
Digite os valores para o array: 1 2 3 4 5 6 7 8 -100
[1, 2, 3, 4, 5, 6, 7, 8]
```

```
Digite os valores para o array: 1 2 3 4 -100
[1, 2, 3, 4]
```

2. Desenvolva um programa capaz de ler n valores inteiros e os insira em um array. A leitura deve ser feita de acordo com o programa feito no exercício 1. Após a leitura, desenvolva a função **calcula_media** e **calcula_dvp** para calcular a média e o desvio padrão dos valores informados. O programa principal deve receber os valores lidos pelo array e retornar a média e o desvio padrão no seguinte formato: MÉDIA +- DVP

Exemplos:

```
Digite os valores para o array: 1 2 3 4 5 6 7 -100
4.00 +- 1.7
```

```
Digite os valores para o array: 1 1 1 1 1 1 1 1 1 1 1 -100
1.00 +- 0.0
```

```
Digite os valores para o array: 10 20 1 40 50 90 100 -100
44.43 +- 29.8
```

3. Desenvolva um programa capaz de ler dois vetores com três elementos inteiros cada. A leitura deve ser feita em uma única linha para cada vetor, com os elementos separados por espaços. Após a leitura, o programa deve exibir o produto escalar entre os vetores digitados no formato exemplificado abaixo. O cálculo do produto escalar deve obrigatoriamente ser feito pela função **produto_escalar**, que recebe como referência os dois arrays e retorna um valor inteiro referente ao resultado do produto escalar.

Exemplos:

```
Digite o primeiro array: 1 2 3
Digite o segundo array: 3 2 1
[1, 2, 3] . [3, 2, 1] = 10

Digite o primeiro array: 10 20 30
Digite o segundo array: -10 2 5
[10, 20, 30] . [-10, 2, 5] = 90
```

4. Desenvolva um programa capaz de ler os valores de uma matriz $m \times n$. A leitura deve ser feita em uma linha para cada linha da matriz (ver ex1). A condição de parada para a leitura das linhas é uma linha vazia (apenas com -100). O programa deve assumir que a quantidade de colunas da primeira linha define a dimensão da matriz, ou seja, qualquer linha subsequente que não tenha a mesma quantidade de linhas da linha inicial deve causar um erro e o encerramento do programa. Após a leitura da matriz, ela deve ser exibida seguindo o padrão mostrado nos exemplos a seguir.

Exemplos:

```
Linha 0: 1 1 1 -100
Linha 1: 2 2 2 -100
Linha 2: 3 3 3 -100
Linha 3: -100
[[1, 1, 1],
 [2, 2, 2],
 [3, 3, 3]]
```

```
Linha 0: 1 1 1 1 1 -100
Linha 1: 1 2 3 -100
Número de colunas incompatível. Saindo...
```

```
Linha 0: 1 2 3 4 5 6 7 8 -100
Linha 1: 8 7 6 5 4 3 2 1 -100
Linha 2: -100
[[1, 2, 3, 4, 5, 6, 7, 8],
 [8, 7, 6, 5, 4, 3, 2, 1]]
```

5. Desenvolva um programa capaz de ler os valores inteiros de uma matriz 3x3 e, a seguir, calcule e exiba sua determinante. O cálculo da determinante deve ser feito pela função determinante, que recebe por referência uma matriz 3x3 e retorna o valor inteiro referente à determinante.

Exemplos:

```
Digite a linha 0: 2 3 4
Digite a linha 1: 1 3 4
Digite a linha 2: 5 6 7
-3
```

```
Digite a linha 0: 1 0 0
Digite a linha 1: 0 1 0
Digite a linha 2: 0 0 1
1
```

6. Desenvolva um programa capaz de ler os valores inteiros de uma matriz 3x3 e, a seguir, exiba a sua forma transposta.

Exemplos:

```
Digite a linha 0: 1 2 3
Digite a linha 1: 4 5 6
Digite a linha 2: 7 8 9
[[1, 4, 7],
 [2, 5, 8],
 [3, 6, 9]]
```

```
Digite a linha 0: 1 0 0
Digite a linha 1: 0 1 0
Digite a linha 2: 0 0 1
[[1, 0, 0],
 [0, 1, 0],
 [0, 0, 1]]
```

7. Desenvolva um programa capaz de ler os valores inteiros de duas matrizes 3x3 e, a seguir, exiba a matriz resultante da multiplicação entre elas.

Dica: talvez seja útil transpor a segunda matriz.

Exemplos:

```
Digite a linha 0: 1 0 0
Digite a linha 1: 0 1 0
Digite a linha 2: 0 0 1
Digite a linha 0: 1 1 1
Digite a linha 1: 2 2 2
Digite a linha 2: 3 3 3
[[1, 1, 1],
 [2, 2, 2],
 [3, 3, 3]]
```

```
Digite a linha 0: 1 2 3
Digite a linha 1: 4 5 6
Digite a linha 2: 7 8 9
Digite a linha 0: 1 1 1
Digite a linha 1: 2 2 2
Digite a linha 2: 3 3 3
[[14, 14, 14],
[32, 32, 32],
[50, 50, 50]]
```

```
Digite a linha 0: 2 0 0
Digite a linha 1: 0 2 0
Digite a linha 2: 0 0 2
Digite a linha 0: 1 1 1
Digite a linha 1: 2 2 2
Digite a linha 2: 3 3 3
[[2, 2, 2],
[4, 4, 4],
[6, 6, 6]]
```