

Exercícios de vetores – Lista 1

Questão 01 - Escreva um algoritmo que solicite ao usuário a entrada de 5 nomes, e que exiba a lista desses nomes na tela. Após exibir essa lista, o programa deve mostrar também os nomes na ordem inversa em que o usuário os digitou, um por linha.

```
package liex11;
import java.util.Scanner;
public class Liex11 {
    static Scanner leia = new Scanner(System.in);
    public static void main(String[] args) {
        String nome[] = new String[5];
        for (int i=0; i < nome.length; i++) {
            System.out.println("Digite o nome:");
            nome[i] = leia.nextLine();
        }
        for (int i=nome.length-1; i >=0; i--) {
            System.out.println(nome[i]);
        }
    }
}
```

Questão 02 - Crie um programa que solicite a entrada de 10 números pelo usuário, armazenando-os em um vetor, e então monte outro vetor com os valores do primeiro multiplicados por 5. Exiba os valores dos dois vetores na tela, simultaneamente, em duas colunas (um em cada coluna), uma posição por linha.

```
package lista12;
import java.util.Scanner;
public class Lista12 {
    static Scanner leia = new Scanner(System.in);
    public static void main(String[] args) {
        int num1[] = new int[10];
        int num2[] = new int[10];
        for (int i=0; i < num1.length; i++) {
            System.out.println("Digite o numero:");
            num1[i] = leia.nextInt();
            num2[i] = num1[i] * 5;
        }
        for (int j=0; j < num1.length; j++) {
            System.out.println(num1[j] + " - " + num2[j]);
        }
    }
}
```

Questão 03 - Crie um programa que armazene 10 números digitados pelo usuário em dois vetores: um somente para números pares, e outro somente para números ímpares. Após, exiba os valores dos dois vetores na tela, em sequência.
Obs.: As posições que não receberem valores exibirão o número zero. Não se preocupe com isso por enquanto.

```
package liex13;
import java.util.Scanner;
public class Liex13 {
    static Scanner leia = new Scanner(System.in);
    public static void main(String[] args) {
        char sexo;
        int par[] = new int[10];
        int impar[] = new int[10];
        int num=0;
        for (int i=1; i < 10; i++) {
            System.out.println("Digite um número:");
            num = leia.nextInt();
            if ( num% 2 == 0) {
                par[i] = num;
            }
            else {
                impar[i]=num;
            }
        }
        for (int i=1; i < 10; i++) {
            System.out.println(par[i] + " " + impar[i]);
        }
    }
}
```

Questão 4: Crie um algoritmo que receba os dados de dois vetores separadamente. O primeiro vetor terá 10 elementos e o segundo vetor 7. Ordene esses vetores separadamente de forma ascendente. Depois preencha um terceiro de vetor de 17 posições também ordenado.

```
package bubble;

public class Buble {

    public static void ordenacao_bubbleSort(int[ ] vetor) {
        int out, in;
        int nElems = vetor.length;
        int temp = 0;
        for(out=nElems-1; out>0; out--) // loop externo (trás para frente)
            for(in=0; in<out; in++) // loop interno (em frente)
                if( vetor[in] > vetor[in+1] ) { // fora de ordem?
                    temp = vetor[in];
                    vetor[in] = vetor[in+1];
                    vetor[in+1] = temp;
                }
            } // fim bubbleSort()

    public static void main(String argv[]) {
        int[ ] a = new int[ ] {77,99,44,55,22,88,11,27,66,33};
```

```

int[ ] b = new int[ ] {71,9,47,34,23,65,15};
int[ ] c = new int[17];

System.out.println();
ordenacao_bubbleSort(a);
for (int v=0;v<a.length; v++) {
    System.out.println(a[v]);
}
System.out.println();
ordenacao_bubbleSort(b);
for (int t=0;t<b.length; t++) {
    System.out.println(b[t]);
}
int j=0, v=0, i=0;
while (j<10 && v<7) {
    if (a[j] < b[v]) {
        c[i]= a[j];
        j++;
    }
    else{
        c[i]= b[v];
        v++;
    }
    i++;
}
if (j==10) {
    for (int x=v;x<7;x++) {
        c[i]=b[x];
        i++;
    }
}
else {
    for (int y=j;y<10;y++) {
        c[i]=a[y];
        i++;
    }
}
for (int u=0;u<c.length; u++) {
    System.out.println(c[u]);
}
}
}

```

Questão 5: Receba dois vetores um com o Nome do país, outro com a quantidade de pedidos, ordene de forma ascendente pela quantidade de pedidos. E imprima o vetor de forma ascendente.

```

package liex05;
public class Liex05 {
    public static void main(String[] args) {
        int[ ] a = new int[ ] {130,450,460,350,500,20,35,120,230,200};
        String[ ] b = new String[ ]
{"Brasil","Inglaterra","USA","México","china","Azerbaijão","Argentina","França","Italia","Espanha"};
    }
}

```

```

        int out, in;
        int nElems = a.length;
        int temp = 0;
        String temps = "";
        for(out=nElems-1; out>0; out--) { // loop externo (trás para frente)
            for(in=0; in<out; in++) { // loop interno (em frente)
                if( a[in] > a[in+1] ) { // fora de ordem?
                    temp = a[in];
                    a[in] = a[in+1];
                    a[in+1] = temp;
                    temps = b[in];
                    b[in] = b[in+1];
                    b[in+1] = temps;
                }
            }
        }
    } // fim bubbleSort()

    for (int t=0; t<a.length; t++) {
        System.out.println(b[t] + " - " + a[t]);
    }
}

```

Questão 6: Crie um algoritmo que preencha uma matriz (10 X 10), utilizando as regras a seguir:

- Para Coluna maior que a linha = $2L+1$
- Para Coluna igual à linha = 1
- Para Coluna menor que a linha = $2C + 3L$

```

package liex16;
public class Liex16 {
    public static void main(String[] args) {
        int[ ][ ] quant = new int[ 10 ][ 10 ];
        for (int l = 0; l < quant.length; l++) {
            for (int c = 0; c < quant[l].length; c++) {
                if (c>l) {
                    quant[l][c] = (2*l)+1;
                }
                if(c==l) {
                    quant[l][c] = 1;
                }
                if (c<l) {
                    quant[l][c] = 2*c + 3*l;
                }
            }
        }

        for (int l = 0; l < quant.length; l++) {
            for (int c = 0; c < quant[l].length; c++) {
                System.out.print(quant[l][c]+ " ");
            }
            System.out.println(" ");
        }
    }
}

```

Questão 7: Crie um algoritmo que preencha uma matriz (8 X 8), utilizando as regras a seguir:

- Para Coluna maior que a linha = $3C+4L$
- Para Coluna igual à linha = 2
- Para Coluna menor que a linha = $2C + 3L$

```
package liex17;
public class Liex17 {
    public static void main(String[] args) {
        int[ ][ ] mat = new int[ 8 ][ 8 ];
        for (int l = 0; l < mat.length; l++) {
            for (int c = 0; c < mat[l].length; c++) {
                if (c>l) {
                    quant[l][c] = (3*c)+4*l;
                }
                if(c==l) {
                    quant[l][c] = 2;
                }
                if (c<l) {
                    quant[l][c] = 2*c + 3*l;
                }
            }
        }

        for (int l = 0; l < quant.length; l++) {
            for (int c = 0; c < quant[l].length; c++) {
                System.out.print(quant[l][c]+ " ");
            }

            System.out.println(" ");
        }
    }
}
```