

```
package funções;

import java.util.Arrays;
import java.util.Collections;
import java.util.Random;
import java.util.Scanner;

public class Funcoes_arrays {

    public static int gerarnumerosAleatorios(int vet[]) {

        Random r = new Random();
        for (int i = 0; i < vet.length; i++) {
            vet[i] = r.nextInt(10);

            System.out.println(r.nextInt(10));

        }
        return 0;
    }

    public static int Usuario() {
        Scanner entrada = new Scanner(System.in);
        System.out.println("Digite um numero:");
        return entrada.nextInt();
    }

    private static void preencherVetor(Class<vet[]> vet) {
        for (int i = 0; i < vet.length; i++) {
            vet[i] = gerarnumerosAleatorios(vet);
        }
    }
}
```

```
public static String imprimirVetor(int vet[]) {
```

```
String resultado = "-";
```

```
for (int i : vet) {
```

```
    resultado += "-" + i;
```

```
}
```

```
return resultado;
```

```
}
```

```
/* pesquisa */
```

```
public static int pesquisar(int num, int vetor[]) {
```

```
for (int i : vetor) {
```

```
    if (vetor[i] == num) {
```

```
        return i;
```

```
    }
```

```
}
```

```
}
```

```
/* pesquisa por indice */
```

```
public static int pesquisaindice(int vet[], int indice) {
```

```
return vet[indice];
```

```
}
```

```
/* separacao */
```

```
public static void separação(int vet[], int vetNegativo[], int vetPositivo[]) {
```

```
int indiceNegativo = 0;
```

```
int indicePositivo = 0;
```

```
for (int i = 0; i < vet.length; i++) {  
    if (vet[i] < 0) {  
        vetNegativo[indiceNegativo] = vet[i];  
        indiceNegativo++;  
    } else {  
        vetPositivo[indicePositivo] = vet[i];  
        indicePositivo++;  
    }  
}  
  
}
```

```
/* somatorio */  
public static int somatorio(int vet[]) {  
    int somatorio = 0;  
    for (int i : vet) {  
        somatorio += vet[i];  
    }  
  
}
```

```
return somatorio;  
}
```

```
/* junção */  
public static int[] junção(int vetA[], int vetB[]) {  
    int contador = 0;  
  
    int junção[] = new int[vetA.length * 2];  
    for (int i : junção) {  
        if (i == vetA.length) {
```

```
contador = 0;
```

```
/* junção */
```

```
}
```

```
if (i < vetA.length) {
```

```
    junção[i] = vetA[contador];
```

```
    contador++;
```

```
} else {
```

```
    junção[i] = vetB[contador];
```

```
}
```

```
}
```

```
return junção;
```

```
}
```

```
/* inversao */
```

```
public static int[] inversaodeOrdem(int vet[]) {
```

```
    int contador;
```

```
    int aux[] = new int[vet.length];
```

```
    for (int i : vet) {
```

```
        aux[i] = vet[contador];
```

```
        contador++;
```

```
    }
```

```
    return aux;
```

```
}
```

```
/* inverso */
```

```
public static int[] inversoVetores(int vetA[], int vetB[]) {
```

```

int inverso[] = new int[vetA.length ];
for (int i : inverso) {
    if (i == vetA.length) {
        vetB.length=i;
    } else {
        vetA.length=i;
    }

    return inversoVetores;

/* priorizaçao */
public static int priorizaçao(int vetA[], int vetB[]) {

    int B[] = new int[vetA.length];
    for (int i : B) {
        vetB[i] = vetA[i];
    }

    return priorizaçao(null, null);
}

public static int compactação() {

    int quantzero = 0;
    int quantum = 0;

}

public static void main(String[] args) {
    int vetor[] = new int[50];

```

```
Usuario();  
gerarNumerosAleatorios(vetor);  
for (int i : vetor) {  
    System.out.print(i + " ");  
}  
  
//    System.out.println(pesquisar);  
  
}  
  
}
```