Questão 2:

```java
import java.util.*;

public class Bolha {
    public static void bolha(int[] array, int n) {

        int temp;
        // i marca até onde já está ordenado
        for (int i = 0; i < n - 1; i++) {
            for (int j = n - 1; j > i; j--) {
                if (array[j] < array[j - 1]) {
                    temp = array[j];
                    array[j] = array[j - 1];
                    array[j - 1] = temp;
                }
            }
        }



    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Digite o tamanho do vetor");
        int tam = sc.nextInt();

        int[] vet = new int[tam];

        System.out.println("Preencha o Vetor");
        for (int i = 0; i < vet.length; i++) {
            vet[i] = sc.nextInt();
        }

        bolha(vet, tam);

        for (int i = 0; i < vet.length; i++) {
            System.out.print(vet[i] + " ");
        }
    }
}
```

Questão 3 :

```java
import java.util.*;

public class Insercao {
    public static void insercao(int[] array, int n) {

        for (int i = 1; i < n; i++) {
            int tmp = array[i];
            int j = i - 1;
            while ((j >= 0) && (array[j] > tmp)) {
                array[j + 1] = array[j];
                j--;
            }
            array[j + 1] = tmp;
        }

    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Digite o tamanho do vetor");
        int tam = sc.nextInt();

        int[] vet = new int[tam];

        System.out.println("Preencha o Vetor");
        for (int i = 0; i < vet.length; i++) {
            vet[i] = sc.nextInt();
        }

        insercao(vet, tam);

        System.out.println("Vetor Ordenado:");
        for (int i = 0; i < vet.length; i++) {
            System.out.print(vet[i] + " ");
        }

        sc.close();
    }
}
```

Questão 4:

```java
import java.util.*;

public class Shellsort {

    public static void shellsort(int[] array, int n) {
        int h = 1;
        do {
            h = (h * 3) + 1;
        } while (h < n);
        do {
            h /= 3;
            for (int cor = 0; cor < h; cor++) {
                insercaoCor(array, n, cor, h);
            }
        } while (h != 1);
    }

    public static void insercaoCor(int[] array, int n, int cor, int h) {
        for (int i = (h + cor); i < n; i += h) {
            int tmp = array[i];
            int j = i - h;
            while ((j >= 0) && (array[j] > tmp)) {
                array[j + h] = array[j];
                j -= h;
            }
            array[j + h] = tmp;
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Informe o tamanho do vetor: ");
        int n = sc.nextInt();
        int[] array = new int[n];

        System.out.println("Informe os números para preencher o vetor:");
        for (int i = 0; i < n; i++) {
            array[i] = sc.nextInt();
        }

        shellsort(array, n);

        System.out.println("Vetor ordenado:");
        for (int i = 0; i < n; i++) {
            System.out.print(array[i] + " ");
        }
```

```
        System.out.println();

        sc.close();
    }
}
```

Questão 5:

Respostas :

a) Seleção:

a) Utilizando o algoritmo de Seleção:

Passo 1: [10, 1, 3, 20, 5, 0, 1, -3, 9, 2, 2, 7, 3, 4]
Passo 2: [-3, 1, 3, 20, 5, 0, 1, 10, 9, 2, 2, 7, 3, 4]
Passo 3: [-3, 0, 3, 20, 5, 1, 1, 10, 9, 2, 2, 7, 3, 4]
Passo 4: [-3, 0, 1, 20, 5, 3, 1, 10, 9, 2, 2, 7, 3, 4]
Passo 5: [-3, 0, 1, 2, 5, 3, 1, 10, 9, 20, 2, 7, 3, 4]
Passo 6: [-3, 0, 1, 2, 2, 3, 1, 10, 9, 20, 5, 7, 3, 4]
Passo 7: [-3, 0, 1, 2, 2, 3, 1, 10, 9, 20, 5, 7, 3, 4]
Passo 8: [-3, 0, 1, 1, 2, 3, 2, 10, 9, 20, 5, 7, 4,]
Passo 9: [-3, 0, -, 1, 2, 3, 2, 10, 9, 20, 5, 7, 4, 1]
Passo 10: [-3, 0, -, 1, 2, 3, 2, 10, 9, 20, 5, 7, 4, 1]
Passo 11: [-3, 0, -, 1, 2, 3, 2, 10, 9, 20, 5, 7, 4, 1, 2]
Passo 12: [-3, 0, -, 1, 2, 2, 3, 10, 9, 20, 5, 7, 4, 1, 3]
Passo 13: [-3, 0, -, 1, 2, 2, 3, 10, 9, 20, 5, 7, 4, 1, 3]
Passo 14: [-3, 0, -, 1, 2, 2, 3, 4, 9, 20, 5, 7, 10, 1, 3]
Passo 15: [-3, 0, -, 1, 2, 2, 3, 4, 1, 20, 5, 7, 10, 9, 3]
Passo 16: [-3, 0, -, 1, 2, 2, 3, 4, 1, 3, 5, 7, 10, 9, 20]

b) Bolha:

Passo 1: [1, 10, 3, 20, 5, 0, 1, -3, 9, 2, 2, 7, 3, 4]
Passo 2: [1, 3, 10, 5, 0, 1, -3, 9, 2, 2, 7, 3, 4, 20]
Passo 3: [1, 3, 5, 0, 1, -3, 9, 2, 2, 7, 3, 4, 10, 20]
Passo 4: [1, 3, 0, 1, -3, 9, 2, 2, 5, 3, 4, 7, 10, 20]
Passo 5: [1, 0, 1, -3, 9, 2, 2, 3, 4, 5, 7, 3, 10, 20]
Passo 6: [0, 1, -3, 1, 2, 2, 3, 4, 5, 3, 7, 9, 10, 20]
Passo 7: [-3, 0, 1, 1, 2, 2, 3, 4, 3, 5, 7, 9, 10, 20]
Passo 8: [-3, 0, 1, 1, 2, 2, 3, 3, 4, 5, 7, 9, 10, 20]
Passo 9: [-3, 0, 1, 1, 2, 2, 3, 3, 4, 5, 7, 9, 10, 20]
Passo 10: [-3, 0, 1, 1, 2, 2, 3, 4, 3, 5, 7, 9, 10, 20]
Passo 11: [-3, 0, 1, 1, 2, 2, 3, 4, 5, 3, 7, 9, 10, 20]
Passo 12: [-3, 0, 1, 1, 2, 2, 3, 4, 5, 7, 3, 9, 10, 20]
Passo 13: [-3, 0, 1, 1, 2, 2, 3, 4, 5, 7, 3, 9, 10, 20]
Passo 14: [-3, 0, 1, 1, 2, 2, 3, 4, 5, 7, 3, 9, 10, 20]

c) Ordenação por inserção:

Passo 0: [10, 1, 3, 20, 5, 0, 1, -3, 9, 2, 2, 7, 3, 4]

Passo 1: [1, 10, 3, 20, 5, 0, 1, -3, 9, 2, 2, 7, 3, 4]

Passo 2: [1, 3, 10, 20, 5, 0, 1, -3, 9, 2, 2, 7, 3, 4]

Passo 3: [1, 3, 10, 20, 5, 0, 1, -3, 9, 2, 2, 7, 3, 4]

Passo 4: [1, 3, 5, 10, 20, 0, 1, -3, 9, 2, 2, 7, 3, 4]

Passo 5: [0, 1, 3, 5, 10, 20, 1, -3, 9, 2, 2, 7, 3, 4]

Passo 6: [0, 1, 1, 3, 5, 10, 20, -3, 9, 2, 2, 7, 3, 4]

Passo 7: [-3, 0, 1, 1, 3, 5, 10, 20, 9, 2, 2, 7, 3, 4]

Passo 8: [-3, 0, 1, 1, 3, 5, 9, 10, 20, 2, 2, 7, 3, 4]

Passo 9: [-3, 0, 1, 1, 2, 3, 5, 9, 10, 20, 2, 7, 3, 4]

Passo 10: [-3, 0, 1, 1, 2, 2, 3, 5, 9, 10, 20, 7, 3, 4]

Passo 11: [-3, 0, 1, 1, 2, 2, 3, 5, 7, 9, 10, 20, 3, 4]

Passo 12: [-3, 0, 1, 1, 2, 2, 3, 4, 5, 7, 9, 10, 20]

d) Ordenação por Shellsort:

Passo 1: h = 7
[3, 1, 3, 4, 5, 0, 1, -3, 9, 2, 2, 7, 10, 20]

Passo 2: h = 3
[3, 1, -3, 2, 2, 0, 1, 4, 5, 3, 7, 10, 9, 20]

Passo 3: h = 1
[1, -3, 0, 1, 2, 2, 3, 3, 4, 5, 7, 9, 10, 20]

Questão 6 :

Número de comparações

	Seleção	Bolha	Inserção	ShellSort
Vetor Crescente	190	190	19	52
Vetor Decrescente	190	190	190	52
Vetor Aleatório	190	190	209	50

Número de movimentações

	Seleção	Bolha	Inserção	ShellSort
Vetor Crescente	0	0	19	48
Vetor Decrescente	38	190	209	48
Vetor Aleatório	190	190	209	50

Para calcular o número de comparações e movimentações de cada algoritmo, consideramos o pior caso para cada um dos três vetores.

O algoritmo de seleção realiza N-1 comparações no pior caso, onde N é o número de elementos no vetor. Como o vetor crescente, decrescente e aleatório têm o mesmo tamanho, o número de comparações do algoritmo de seleção é o mesmo para todos os três: 19 comparações.

O algoritmo de bolha realiza N*(N-1)/2 comparações no pior caso e N*(N-1)/2 trocas. Portanto, o número de comparações e movimentações do algoritmo de bolha é o mesmo para os três vetores:

Número de comparações: 190

Número de movimentações: 190 para o vetor aleatório e 0 para os vetores crescente e decrescente.

O algoritmo de inserção realiza N*(N-1)/2 comparações e N*(N-1)/2 trocas no pior caso. Portanto, o número de comparações e movimentações do algoritmo de inserção é diferente para cada um dos três vetores:

Para o vetor crescente, o algoritmo de inserção realiza apenas 19 comparações e 19 trocas, já que o vetor já está ordenado.

Para o vetor decrescente, o algoritmo de inserção realiza 190 comparações e 209 trocas.

Para o vetor aleatório, o algoritmo de inserção realiza 209 comparações e 209 trocas.

O algoritmo de ShellSort tem um desempenho melhor que os algoritmos anteriores. No pior caso, o número de comparações e movimentações depende do intervalo inicial usado. Usamos o intervalo de Knuth, que é calculado como (3^k - 1)/2, onde k é o maior inteiro tal que 3^k - 1 <= N. O número de comparações e movimentações do algoritmo de ShellSort para cada um dos três vetores é:

Para o vetor crescente, o algoritmo de ShellSort realiza 52 comparações e 48 movimentações.

Para o vetor decrescente, o algoritmo de ShellSort realiza 52 comparações e 48 movimentações.

Para o vetor aleatório, o algoritmo de ShellSort realiza 50 comp