

Questão 1:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lista12
{
    internal class Program
    {
        public static int[] Selection(int[] vet, int n)
        {
            for (int i = 0; i < n - 1; i++)
            {
                int menor = i;
                for (int j = i + 1; j < n; j++)
                {
                    if (vet[menor] > vet[j])
                    {
                        menor = j;
                    }
                }
                int temp = vet[i];
                vet[i] = vet[menor];
                vet[menor] = temp;
            }
            return vet;
        }

        public static int[] SelectionInvertido(int[] vet, int n)
        {
            for (int i = 0; i < n - 1; i++)
            {
                int maior = i;
                for (int j = i + 1; j < n; j++)
                {
                    if (vet[maior] > vet[j])
                    {
                        maior = j;
                    }
                }
                int temp = vet[i];
                vet[i] = vet[maior];
            }
        }
    }
}
```

```

        vet[maior] = temp;
    }
    return vet;
}
public static int[] Bubblesort(int[] vet, int n)
{
    for (int i = n - 1; i > 0; i--)
    {
        for (int j = 0; j < i; j++)
        {
            if (vet[j] > vet[j + 1])
            {
                int temp = vet[j];
                vet[j] = vet[j + 1];
                vet[j + 1] = temp;
            }
        }
    }
    return vet;
}
public static int[] BubblesortInvertido(int[] vet, int n)
{
    for (int i = n - 1; i > 0; i--)
    {
        for (int j = 0; j < i; j++)
        {
            if (vet[j] < vet[j + 1])
            {
                int temp = vet[j];
                vet[j] = vet[j + 1];
                vet[j + 1] = temp;
            }
        }
    }
    return vet;
}
public static int[] InsertionSort(int[] vet, int n)
{
    for (int i = 1; i < n; i++)
    {
        int temp = vet[i];
        int j = i - 1;
        while (j >= 0 && vet[j] > temp)
        {

```

```

        vet[j + 1] = vet[j];
        j--;
    }
    vet[j + 1] = temp;
}
return vet;
}
public static int[] InsertionSortInvertido(int[] vet, int n)
{
    for (int i = 1; i < n; i++)
    {
        int temp = vet[i];
        int j = i - 1;
        while (j >= 0 && vet[j] < temp)
        {
            vet[j + 1] = vet[j];
            j--;
        }
        vet[j + 1] = temp;
    }
    return vet;
}
static void Main(string[] args)
{
    Console.WriteLine("Insira o tamanho do vetor que deseja criar");
    int tam = int.Parse(Console.ReadLine());
    Random random = new Random();
    int[] vet = new int[tam];
    for(int i = 0; i < vet.Length; i++)
    {
        vet[i] = random.Next(1,101);
    }
    Console.WriteLine("Insira qual operação deseja fazer. 1 - Seleção | 2 - Bubblesort | 3 - Inserção | 4 - Seleção Decrescente | 5 - Bubblesort Decrescente | 6 - Inserção decrescente");
    int escolha = int.Parse(Console.ReadLine());
    switch (escolha)
    {
        case 1:
            vet = Selection(vet, tam);
            break;
        case 2:
            vet = Bubblesort(vet, tam);
            break;
    }
}

```

```
        case 3:
            vet = InsertionSort(vet, tam);
            break;
        case 4:
            vet = SelectionInvertido(vet, tam);
            break;
        case 4:
            vet = BubblesortInvertido(vet, tam);
            break;
        case 5:
            vet = InsertionSortInvertido(vet, tam);
            break;
        default:
            break;
    }
    Console.WriteLine("Resultado: ");
    for(int i = 0; i < vet.Length; i++)
    {
        Console.Write(vet[i] + " ");
    }
    Console.ReadLine();
}
}
```

Questão 2:

Bubblesort

Passo 1 : 1 3 10 5 6 1 4 9 2 20

Passo 2: 1 3 5 6 1 4 9 2 10 20

Passo 3: 1 3 5 1 4 6 2 9 10 20

Passo 4 : 1 3 1 4 5 2 6 9 10 20

Passo 5 : 1 1 3 4 2 5 6 9 10 20

Passo 6: 1 1 3 2 4 5 6 9 10 20

Passo 7: 1 1 2 3 4 5 6 9 10 20

Passo 8: 1 1 2 3 4 5 6 9 10 20

Passo 9: 1 1 2 3 4 5 6 9 10 20

Selection

Passo 1: 1 10 3 20 5 6 1 4 9 2

Passo 2: 1 1 3 20 5 6 10 4 9 2

Passo 3: 1 1 2 20 5 6 10 4 9 3

Passo 4: 1 1 2 3 5 6 10 4 9 20

Passo 5: 1 1 2 3 4 6 10 5 9 20

Passo 6: 1 1 2 3 4 5 10 6 9 20

Passo 7: 1 1 2 3 4 5 6 10 9 20

Passo 8: 1 1 2 3 4 5 6 9 10 20

Insertion

Passo 1: 1 10 3 20 5 6 1 4 9 2

Passo 2: 1 3 10 20 5 6 1 4 9 2

Passo 3: 1 3 10 20 5 6 1 4 9 2

Passo 4: 1 3 5 10 20 6 1 4 9 2

Passo 5: 1 3 5 6 10 20 1 4 9 2

Passo 6: 1 1 3 5 6 10 20 4 9 2

Passo 7: 1 1 3 4 5 6 10 20 9 2

Passo 8: 1 1 3 4 5 6 9 10 20 2

Passo 9: 1 1 2 3 4 5 6 9 10 20

Questão 3:

Número de comparações

	Seleção	Bolha	Inserção
Vetor crescente	190	190	19
Vetor decrescente	190	190	19
Vetor aleatório	190	190	19

Número de movimentações

	Seleção	Bolha	Inserção
Vetor crescente	57	0	19
Vetor decrescente	57	570	209
Vetor aleatório	57	222	93

Questão 4:

O algoritmo por seleção não é estável, já que na hora de ordenar algum elemento, ele não mantém a ordem original dos elementos. Ele faz a troca dos elementos independente se a condição foi atendida ou não (seria estável se movesse todos elementos para a direita, por exemplo, ou como é presente no bubble sort, nem alteraria os elementos se fossem iguais).

```
for (int i = 0; i < (n - 1); i++) {  
    int menor = i;  
    for (int j = (i + 1); j < n; j++) {  
        if (array[menor] > array[j]) {  
            menor = j;  
        }  
    }  
    int temp = array[i];  
    array[i] = array[menor];  
    array[menor] = temp;  
}
```