

Questão 1

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

namespace Lista13Questão1
{
    internal class Program
    {
        private static void quickSortInverso(int[] array, int esq, int dir)
        {
            int part;
            if (esq < dir)
            {
                part = particaoInversa(array, esq, dir);
                quickSortInverso(array, esq, part - 1);
                quickSortInverso(array, part + 1, dir);
            }
        }
        private static int particaoInversa(int[] array, int inicio, int fim)
        {
            int pivot = array[fim];
            int part = inicio - 1;
            for (int i = inicio; i < fim; i++)
            {
                if (array[i] > pivot)
                {
                    part++;
                    swap(array, part, i);
                }
            }
            part++;
            swap(array, part, fim);
            return (part);
        }
        private static void quickSort(int[] array, int esq, int dir)
        {
            int part;
```

```

    if (esq < dir)
    {
        part = particao(array, esq, dir);
        quickSort(array, esq, part - 1);
        quickSort(array, part + 1, dir);
    }
}

```

```

private static int particao(int[] array, int inicio, int fim)
{
    int pivot = array[fim];
    int part = inicio - 1;
    for (int i = inicio; i < fim; i++)
    {
        if (array[i] < pivot)
        {
            part++;
            swap(array, part, i);
        }
    }
    part++;
    swap(array, part, fim);
    return (part);
}

```

```

private static void swap(int[] array, int i, int j)
{
    int temp = array[i];
    array[i] = array[j];
    array[j] = temp;
}

```

```

static void Main(string[] args)
{
    int escolha;
    do
    {

```

```

        Console.WriteLine("\nInsira a ação desejada: 1 - Quicksort | 2 - Quicksort Inverso |
3 - Sair");
        escolha = int.Parse(Console.ReadLine());
        Console.WriteLine("Insira o tamanho do vetor desejado: ");
        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.Write(vet[i] + " ");
}
Console.WriteLine("\nTecle enter para continuar");
Console.ReadLine();
switch (escolha)
{
    case 1:
        quickSort(vet, 0, tam - 1);
        break;
    case 2:
        quickSortInverso(vet, 0, tam - 1);
        break;
    default:
        break;
}
Console.WriteLine("Resultado: ");
for (int i = 0; i < vet.Length; i++)
{
    Console.Write(vet[i] + " ");
}
} while (escolha != 3);
}
}
}

```

Questão 2

Quicksort

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

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

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

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

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

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

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

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

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

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

1 1 3 4 5 6 10 20 9 2 - Passo 11

1 1 3 4 5 6 10 20 9 2 - Passo 12

1 1 3 4 5 6 10 20 9 2 - Passo 13

1 1 3 4 5 6 10 20 9 2 - Passo 14

1 1 3 4 5 6 10 20 9 2 - Passo 15

Questão 3

Número de comparações

	<u>Quicksort</u>
Vetor crescente	19
Vetor decrescente	190
Vetor aleatório	69

Número de movimentações

	<u>Quicksort</u>
Vetor crescente	0
Vetor decrescente	327
Vetor aleatório	171