

Exe 1:

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

namespace exe_1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            double[] temperatura = new double[5];
            double temp;

            for (int i = 0; i < temperatura.Length; i++)
            {
                Console.WriteLine($"Informe a temperatura do dia {i + 1}:");
                temperatura[i] = double.Parse(Console.ReadLine());
            }

            Console.WriteLine("Informe a temperatura a pesquisar:");
            temp = double.Parse(Console.ReadLine());

            Console.WriteLine("Essa temperatura ocorreu nos seguintes dias:");

            for (int i = 0; i < temperatura.Length; i++)
            {
                if (temp == temperatura[i])
```

```

        {
            Console.Write(i + 1);
            Console.Write(",");
        }
    }

    Console.ReadLine();
}
}
}

```

Exe 2:

```

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

namespace exe_2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // Declaração de Variáveis //

            int quantD;

            // //

            Console.WriteLine("Quantos dígitos terá o número:");
            quantD = int.Parse(Console.ReadLine());

```

```
int[] palindromo = new int[quantD];

for (int i = 0; i < palindromo.Length; i++)
{
    Console.WriteLine($"Informe o {i + 1} digito do seu número");
    palindromo[i] = int.Parse(Console.ReadLine());
}

bool teste = true;

for (int i = 0, final = (palindromo.Length - 1); i < (palindromo.Length/2); i++, final--)
{
    if (palindromo[i] != palindromo[final])
    {
        teste = false;
    }
}

if (teste == true)
{
    Console.WriteLine("é verdadeiro");
}
else
{
    Console.WriteLine("é falso");
}

Console.ReadLine();

}

}
```

```
}
```

Exe 3:

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

namespace exe_3
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // Declaração de Variáveis //

            double menorN, maiorN, pesquisaN, somaN, mediaN, maiorM;
            int opcao = 0, selecionar = 0;

            maiorM = 0;
            mediaN = 0;
            somaN = 0;

            menorN = double.MaxValue;
            maiorN = double.MinValue;

            // Vetor //

            double[] notas = new double[5];

            for (int i = 0, cont = 1; i < notas.Length;)
```

```

{
    Console.Write($"Informe a {cont} nota:");
    double notaentrada = double.Parse(Console.ReadLine());

    if (notaentrada >= 0 && notaentrada <= 100)
    {
        notas[i] = notaentrada;
        i++;
        cont++;
    }
    else
    {
        Console.WriteLine("Informe uma nota válida.");
    }
}

while (opcao != 5)
{
    Console.WriteLine();
    Console.WriteLine("-----");
    Console.WriteLine("MENU DE OPÇÕES");
    Console.WriteLine("1. Mostrar a menor nota");
    Console.WriteLine("2. Mostrar maior nota");
    Console.WriteLine("3. Pesquisar nota");
    Console.WriteLine("4. Mostrar média das notas e quantidade de notas acima da
média");
    Console.WriteLine("5. Sair");
    Console.WriteLine("-----");

    Console.Write("Informe a Opção:");
    opcao = int.Parse(Console.ReadLine());
}

```

```
if (opcao == 1)
{
    for (int i = 0; i < notas.Length; i++)
    {
        if (notas[i] < menorN)
        {
            menorN = notas[i];
        }
    }
    Console.WriteLine($"A menor nota foi {menorN}");
}
```

```
if (opcao == 2)
{
    for (int i = 0; i < notas.Length; i++)
    {
        if (notas[i] > maiorN)
        {
            maiorN = notas[i];
        }
    }
    Console.WriteLine($"A maior nota foi {maiorN}");
}
```

```
if (opcao == 3)
{
    Console.Write("Informe a nota que deseja pesquisar:");
    pesquisaN = double.Parse(Console.ReadLine());

    Console.WriteLine($"Nota pesquisada: {pesquisaN}");
}
```

```
Console.Write($"A nota consta nas posições:");

for (int i = 0; i < notas.Length; i++)
{
    if (notas[i] == pesquisaN)
    {
        Console.Write(i + 1);
        Console.Write(",");
    }
}

if (opcao == 4)
{
    for (int i = 0; i < notas.Length; i++)
    {
        somaN = somaN + notas[i];
    }

    mediaN = somaN / notas.Length;
    Console.WriteLine($"A média das notas foi: {mediaN}");

    for (int i = 0; i < notas.Length; i++)
    {
        if (notas[i] > mediaN)
        {
            maiorM++;
        }
    }

    Console.WriteLine($"A quantidade de notas acima da média foi: {maiorM}");
}
```

```

    }
}
}
}

```

Exe 4:

```

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

namespace exe_4
{
    internal class Program
    {
        static void Main(string[] args)
        {

            string[] candidatos = { "Perna Longa", "Pluto", "Mickey", "Bob Esponja", "Cebolinha" };
            int[] votos = new int[5];
            int votosNulos = 0;

            Console.WriteLine("Digite até 100 votos (0 a 4 para candidatos):");
            for (int i = 0; i < 100; i++)
            {
                Console.Write($"Voto {i + 1}: ");
                int voto = Convert.ToInt32(Console.ReadLine());
            }
        }
    }
}

```



```
    if (voto >= 0 && voto <= 4)
    {
        votos[voto]++;
    }
    else
    {
        votosNulos++;
    }
}
```

```
int maisVotado = 0;
int menosVotado = 0;
```

```
for (int i = 1; i < votos.Length; i++)
{
    if (votos[i] > votos[maisVotado])
    {
        maisVotado = i;
    }
    if (votos[i] < votos[menosVotado])
    {
        menosVotado = i;
    }
}
```

```
// Exibindo os resultados
```

```
Console.WriteLine("\nResultados da votação:");
```

```
Console.WriteLine($"Candidato mais votado: {candidatos[maisVotado]} com  
{votos[maisVotado]} votos.");
```

```
Console.WriteLine($"Candidato menos votado: {candidatos[menosVotado]} com  
{votos[menosVotado]} votos.");
```

```

        Console.WriteLine($"Quantidade de votos nulos: {votosNulos}");
    }
}
}

```

Exe 5:

```

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

```

```

namespace exe_5
{
    internal class Program
    {
        static void Main(string[] args)
        {
            double[] notas = new double[60];
            int quantidadeNotas = 0;

```

```

            Console.WriteLine("Digite as notas dos alunos (digite um valor negativo para
terminar):");

            while (quantidadeNotas < 60)
            {
                Console.Write($"Nota do aluno {quantidadeNotas + 1}: ");

                double nota = Convert.ToDouble(Console.ReadLine());

```

```
if (nota >= 0)
{
    notas[quantidadeNotas] = nota;
    quantidadeNotas++;
}
else
{
    quantidadeNotas = 60;
}
}
```

```
if (quantidadeNotas > 0)
{

    double soma = 0;
    for (int i = 0; i < quantidadeNotas; i++)
    {
        soma += notas[i];
    }

    double media = soma / quantidadeNotas;
```

```
Console.WriteLine("Notas dos alunos:");
for (int i = 0; i < quantidadeNotas; i++)
{
    Console.WriteLine(notas[i]);
}
```

```
Console.WriteLine("Notas acima da média:");
```

```

        for (int i = 0; i < quantidadeNotas; i++)
        {
            if (notas[i] > media)
            {
                Console.WriteLine(notas[i]);
            }
        }
    }
    else
    {
        Console.WriteLine("Nenhuma nota foi registrada.");
    }
}
}
}

```

Exe 6:

```

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

```

```

namespace exe_6
{
    internal class Program
    {
        static void Main(string[] args)
        {

```

```
double[] vetA = new double[10];  
double[] vetB = new double[10];  
char[] vetC = new char[10];  
double[] vetD = new double[10];
```

```
Console.WriteLine("Digite 10 elementos para o VetA:");  
for (int i = 0; i < 10; i++)  
{  
    Console.Write($"Elemento {i + 1}: ");  
    vetA[i] = Convert.ToDouble(Console.ReadLine());  
}
```

```
Console.WriteLine("Digite 10 elementos para o VetB:");  
for (int i = 0; i < 10; i++)  
{  
    Console.Write($"Elemento {i + 1}: ");  
    vetB[i] = Convert.ToDouble(Console.ReadLine());  
}
```

```
Console.WriteLine("Digite 10 operações (+, -, *, /) para o VetC:");  
for (int i = 0; i < 10; i++)  
{  
    Console.Write($"Operação {i + 1}: ");  
    vetC[i] = Convert.ToChar(Console.ReadLine());  
}
```

```
for (int i = 0; i < 10; i++)
```

```

{
    switch (vetC[i])
    {
        case '+':
            vetD[i] = vetA[i] + vetB[i];
            break;
        case '-':
            vetD[i] = vetA[i] - vetB[i];
            break;
        case '*':
            vetD[i] = vetA[i] * vetB[i];
            break;
        case '/':
            if (vetB[i] != 0)
            {
                vetD[i] = vetA[i] / vetB[i];
            }
            else
            {
                Console.WriteLine($"Divisão por zero em operação {i + 1}. Resultado definido
como 0.");
                vetD[i] = 0;
            }
            break;
        default:
            Console.WriteLine($"Operação inválida {i + 1}: {vetC[i]}");
            vetD[i] = 0;
            break;
    }
}

```

```
Console.WriteLine("Resultado do VetD:");  
for (int i = 0; i < 10; i++)  
{  
    Console.WriteLine($"Elemento {i + 1}: {vetD[i]}");  
}  
}  
}
```