



Procedimentos e Funções

? EXERCÍCIO 1: RAÍZES QUADRADAS

```
using System;
namespace PF1
{class RaizesQuadradas
    {static void RaizesQuad(int Inf, int Sup)
        {for (int I=Inf; I<=Sup;I++)
            Console.WriteLine("{0,5:F1} ",Math.Sqrt(I));
        Console.WriteLine(); }
    static void Main(string[] args)
        {int Inf, Sup;
        Inf=1; Sup=20;
        RaizesQuad(Inf, Sup);
        Inf=25; Sup=40;
        RaizesQuad(Inf, Sup);
        Inf=100; Sup=120;
        RaizesQuad(Inf, Sup);
        }}}}
```

? EXERCÍCIO 2: SELECÇÃO DE NOMES

```
using System;
namespace PF2
{class SeleccaoNomes
    {static void EscreverLista(string[] Nomes, int Conta)
        {for (int I=0; I<Conta;I++)
            Console.WriteLine(Nomes[I]); }
    static void Main(string[] args)
        {string[] Nomes={"Ana", "António", "Beatriz", "Joana",
        "Raul", "Vitoria"};
        int X=4;
        EscreverLista(Nomes, X);
        Console.WriteLine("Seleccionámos {0} de {1} nomes",X,
        Nomes.Length);}}}
```

? EXERCÍCIO 3: TROCA DE VALORES

```
using System;
namespace PF3
{class TracaDeValores
```

```
{static void Troca(ref string X, ref string Y)
{string Temporaria;
Temporaria=X;
X=Y;
Y=Temporaria;}
static void Main(string[] args)
{string X, Y;
Console.Write("Digite o valor da primeira variavel (X) ");
X=Console.ReadLine();
Console.Write("Digite o valor da segunda variavel (Y) ");
Y=Console.ReadLine();
Troca(ref X, ref Y);
Console.WriteLine("X= {0}", X);
Console.WriteLine("Y= {0}", Y);
}}}
```

? EXERCÍCIO 4: PASSAGEM DE REFERÊNCIAS

```
using System;
namespace PF4
{class RefeOut
{static void PotRef (ref double X, int Expoente)
{X= Math.Pow(X,Expoente);}
static void PotOut (out double Z, int Expoente)
{Z=3;
Z=Math.Pow(Z, 3);}
static void Main(string[] args)
{double X=4;
int Expoente=3;
double Z;
Console.WriteLine("Valor original de X={0}", X);
Console.WriteLine("Z não foi inicializado");
Console.WriteLine("Valor original do expoente={0}", Expoente);
PotRef(ref X, Expoente);
PotOut( out Z, Expoente);
Console.WriteLine("X levantado ao expoente={0}", X);
Console.WriteLine("Z levantado ao expoente={0}", Z);
Console.WriteLine("Valor final de X={0}", X);
Console.WriteLine("Valor final de Z={0}", Z);
Console.WriteLine("Valor final do expoente={0}", Expoente);
}}}
```

? EXERCÍCIO 5: MAIOR E MENOR DE TRÊS NÚMEROS

```
using System;
namespace PF5
{
    class MaiorEMenorDe3
    {
        static int Maiorde3(int X, int Y, int Z)
        {
            if (X < Y)
                X=Y;
            if (X < Z)
                X=Z;
            return X;
        }
        static int Menorde3(int X, int Y, int Z)
        {
            if (X > Y)
                X=Y;
            if (X > Z)
                X=Z;
            return X;
        }
    }
    static void Main(string[] args)
    {
        int X, Y, Z;
        Console.Write("Digite o primeiro valor (X) ");
        X=Convert.ToInt16(Console.ReadLine());
        Console.Write("Digite o segundo valor (Y) ");
        Y=Convert.ToInt16(Console.ReadLine());
        Console.Write("Digite o terceiro valor (Z) ");
        Z=Convert.ToInt16(Console.ReadLine());
        Console.WriteLine("Maior={0}", Maiorde3(X, Y, Z));
        Console.WriteLine("Menor={0}", Menorde3(X, Y, Z));
    }
}
```

? EXERCÍCIO 6: CONTAGEM DE NOMES

```
using System;
namespace PF6
{
    class ContagemNomes
    {
        static void ContaIniciais(string Nome, ref int Conta)
        {
            if ((Nome.Substring(0, 1) == "A") || (Nome.Substring(0, 1) == "B")
                || (Nome.Substring(0, 1) == "C"))
                Conta++;
        }
        static void Main(string[] args)
        {
            int Conta=0;
            string Nome;
            Console.Write("Digite um nome ou ZZZ ");
            Nome=Console.ReadLine().ToUpper();
            while (Nome.ToUpper() != "ZZZ")
                ContaIniciais(Nome, ref Conta);
        }
    }
}
```

```

        Console.Write("Digite um nome ou ZZZ ");
        Nome=Console.ReadLine().ToUpper();}
    Console.WriteLine("Há {0} nomes começados por A, B ou C",
        Conta);
    }
}
```

? EXERCÍCIO 7: EFEITOS INDESEJADOS

```

using System;
namespace PF7
{class EfeitosIndesejados
    {static void Triplo(int A)
        {A=3 * A;}
    static void Main(string[] args)
        {Console.Write("Digite um número ");
        int A=Convert.ToInt16(Console.ReadLine());
        Triplo(A);
        Console.WriteLine("Triplo do número={0}", A);
        }
    }
}
```

```

using System;
namespace PF7R
{class CorreccaoEfeitosIndesejadosR
    {static void Triplo(ref int A)
        {A=3 * A;}
    static void Main(string[] args)
        {Console.Write("Digite um número ");
        int A=Convert.ToInt16(Console.ReadLine());
        Triplo(ref A);
        Console.WriteLine("Triplo do número={0}", A);
        }
    }
}
```

? EXERCÍCIO 8: HORAS, MINUTOS E SEGUNDOS

```

using System;
namespace PF8
{class HorasMinutosSegundos
    {static int Segundos(int Seg)
        {return Seg%60;}
    static int Minutos(int Seg)
        {return (Seg/60)%60; }
    static int Horas(int Seg)
        {return (Seg/60) / 60; }
    static void Main(string[] args)
        {Console.Write("Digite o número de segundos a converter ");
        }
    }
}
```

```
int Seg=Convert.ToInt32(Console.ReadLine());
int H=Horas(Seg);
int M=Minutos(Seg);
int S=Segundos(Seg);
Console.WriteLine("{0}={1}:{2}:{3}", Seg, H, M, S);
}}}
```

? EXERCÍCIO 9: NOTAS ENTRE 0 E 20

```
using System;
namespace PF9
{class NotasEntre0e20
{static int LeValida (int I)
{int N=0;
bool Valida=false;
while (Valida == false)
{try
{Console.Write("Nota do aluno {0}=", I+1);
N=Convert.ToInt16(Console.ReadLine());
Valida=N>=0 && N<=20;
if (Valida==false)
throw new Exception();}
catch
{Console.WriteLine ("A nota e um número inteiro igual ou" +
"superior a 0 e inferior ou igual a 20");}}
return N;}
static void Main(string[] args)
{int[] X=new int[5];
for (int I=0; I<=X.Length-1; I++)
{X[I]=LeValida(I);}
Console.WriteLine("Notas lidas: ");
for (int I=0; I<=X.Length-1; I++)
{Console.WriteLine(I+1+ "--" + X[I]);}}}}
```

? EXERCÍCIO 10: ESCRITA DE VECTORES

```
using System;
namespace PF10
{class LeituraEEscritaDeVectores
{static void LeituraVector(string[] A)
{for (int I=0; I<=A.Length-1; I++)
{Console.Write("Cidade {0}=", I+1);
A[I]=Console.ReadLine();}}
static void EscritaVector(string[] A)
{foreach ( string I in A)
```

```

        Console.Write(I+"\n");}
static void Main(string[] args)
{Console.Write("Quantas cidades há? ");
  int N=Convert.ToInt16(Console.ReadLine());
  string[] A=new string[N];
  LeituraVector(A);
  EscritaVector(A);
}}}
```

? EXERCÍCIO 11: MÉDIA ARITMÉTICA

```

using System;
namespace PF11
{class MediaAritmetica
  {static void LeituraVector(double[] A)
    {for (int I=0; I<=A.Length-1; I++)
      {Console.Write("Elemento {0}=", I+1);
        A[I]=Convert.ToDouble(Console.ReadLine());}}
  static double Media(double[] A)
    {double Soma=0;
      for (int I=0; I<=A.Length-1; I++)
        Soma += A[I];
      return Soma / (A.Length);}
  static void Main(string[] args)
    {Console.Write("Quantos elementos tem o vector? ");
      int N=Convert.ToInt16(Console.ReadLine());
      double[] A=new double[N];
      LeituraVector(A);
      Console.WriteLine("Média aritmética={0,4:F2}", Media(A));
    }}
```

? EXERCÍCIO 12: CONTAGEM DE ELEMENTOS

```

using System;
namespace PF12
{class Contagens
  {static void Leitura(int[] A)
    {for (int I=0; I<=A.Length-1; I++)
      {Console.Write("Número {0}=", I+1);
        A[I]=Convert.ToInt16(Console.ReadLine());}}
  static void Contagem(int[] A, int[] Conta)
    {for (int I=0; I<=A.Length-2;I++)
      {if (Conta[I]!=0)
        {for (int J=I+1; J<=Conta.Length-1; J++)
          {if (A[I]==A[J])
```

```
        {Conta[I]++;
          Conta[J]=0;}}}}
static void Escrita(int[] A, int[] Conta)
{for (int I=0; I<=Conta.Length-1; I++)
  if (Conta[I]!=0)
    Console.WriteLine("{0} ocorre {1} vez(es) no vector", A[I],
      Conta[I]);}
static void Main(string[] args)
{int[] A=new int[7];
 int [] Conta=new int[7];
 for (int I=0; I<=A.Length-1; I++)
  Conta[I]=1;
 Leitura(A);
 Contagem(A, Conta);
 Escrita(A, Conta);
}}}
```

? EXERCÍCIO 13: ORDENAÇÃO DE VECTORES

```
using System;
namespace P13
{class OrdenacaoVector
 {static void Ordena(string[] X)
  {string Temp;
   for (int P=X.Length-1; P>=1; P--)
    for (int Corrente=0; Corrente<=P-1; Corrente++)
     if (X[Corrente].CompareTo(X[Corrente+1])>0)
      {Temp=X[Corrente];
       X[Corrente]=X[Corrente+1];
       X[Corrente+1]=Temp;}}
 static void EscritaVector(string[] X)
 {foreach ( string I in X)
  Console.Write(I+"\n");
  Console.WriteLine();}
 static void Main(string[] args)
 {string[] A={"Joana", "Rui", "Adelaide", "Joaquim","Teresa",
  "Miguel"};
  string[] B={"Maria", "Gabriela", "Rita", "Martinho"};
  Ordena(A);
  EscritaVector(A);
  Ordena(B);
  EscritaVector(B);
}}}
```

? EXERCÍCIO 14: SOMATÓRIOS DE LINHAS E COLUNAS

```
using System;
namespace PF14
{
    class SomatorioLinhasColunas
    {
        static void Lermatriz(int[,] A)
        {
            for (int I=0; I<=A.GetLength(0)-2; I++)
                for (int J=0; J<=A.GetLength(1)-2; J++)
                    {
                        Console.Write("Elemento A [{0},{1}]? ", I, J);
                        A[I, J]=Convert.ToInt16(Console.ReadLine());
                    }
        }
        static void Somarlinhas(int [,] A)
        {
            int Total;
            for (int I=0; I<=A.GetLength(0)- 2; I++)
            {
                Total=0;
                for (int J=0; J<=A.GetLength(1)-2; J++)
                    Total += A[I, J];
                A[I, A.GetLength(1)-1]=Total;
            }
        }
        static void Somarcolunas(int [,] A)
        {
            int Total;
            for (int J=0; J<=A.GetLength(1)- 2; J++)
            {
                Total=0;
                for (int I=0; I<=A.GetLength(0)-2; I++)
                    Total += A[I, J];
                A[A.GetLength(0)-1, J]=Total;
            }
        }
        static void Escrevermatriz(int[,] A)
        {
            for (int I=0; I<=A.GetLength(0)-1; I++)
            {
                for (int J=0; J<=A.GetLength(1)-1; J++)
                {
                    if (I!=A.GetLength(0)-1 || J!=A.GetLength(1)-1)
                        Console.Write("{0,5}", A[I, J]);
                    Console.WriteLine();
                }
            }
        }
        static void Main(string[] args)
        {
            int N, M, Total;
            Console.Write("Quantas linhas tem a matriz? ");
            N=Convert.ToInt16(Console.ReadLine());
            Console.Write("Quantas colunas tem a matriz? ");
            M=Convert.ToInt16(Console.ReadLine());
            int[,] A=new int[N+1, M+1];
            Lermatriz(A);
            Somarlinhas(A);
            Somarcolunas(A);
            Escrevermatriz(A);
        }
    }
}
```


? EXERCÍCIO 15: ORDENAÇÃO DE MATRIZES

```
using System;
namespace PF15
{
    class Ordenacaomatrizes
    {
        static void Ordena(string[,] A, int Coluna)
        {
            for (int P=A.GetLength(0)-1; P>=1; P--)
                for (int Corrente=0; Corrente<=P-1; Corrente++)
                    if (A[Corrente,Coluna].CompareTo(A[Corrente+1,Coluna])>0)
                        Troca(A, Corrente);
        }
        static void Troca(string[,] A, int Corrente)
        {
            string Temp;
            for (int J=0; J<=A.GetLength(1)-1; J++)
            {
                Temp=A[Corrente, J];
                A[Corrente, J]=A[Corrente+1, J];
                A[Corrente+1,J]=Temp;
            }
        }
        static void Escrita(string[,] A)
        {
            for (int I=0; I<=A.GetLength(0)-1; I++)
            {
                for (int J=0; J<=A.GetLength(1)-1; J++)
                    Console.Write("{0, 10}", A[I, J].PadRight(10));
                Console.WriteLine();
            }
        }
        static void Main(string[] args)
        {
            string[,] A={{ "60", "Rui", "Porto"},
                          {"20", "Joaquim", "Aveiro"},
                          {"30", "Adelino", "Porto"},
                          {"50", "Ricardo", "Coimbra"} };
            Console.Write("Coluna (0,1,2) que é chave de ordenação ");
            int Coluna=Convert.ToInt16(Console.ReadLine());
            Ordena(A, Coluna);
            Escrita(A);
        }
    }
}
```

? EXERCÍCIO 16: MULTIPLICAÇÃO DE MATRIZES

```
using System;
namespace PF16
{
    class MultiplicaMatrizes
    {
        static void Main(string[] args)
        {
            int[,] A= { { 1, 2, 3 }, { 4, 0, 2 } };
            int[,] B= { { 1, 0 }, { 2, 2 }, { 3, 3 } };
            int[,] C=Multiplica(A, B);
            for (int I=0; I<=A.GetLength(0)-1; I++)
            {
                for (int J=0; J<=B.GetLength(1)- 1; J++)
                    Console.Write("{0, 4}", C[I, J]);
                Console.WriteLine();
            }
        }
    }
}
```

```

    }
}
using System;
namespace P16
{
    class MultiplicaMatrizes
    {
        static int[,] Multiplica(int[,] A, int[,] B)
        {
            int Total=0;
            int[,] C=new int[A.GetLength(0), B.GetLength(1)];
            for (int I=0; I<=A.GetLength(0)-1; I++)
            {
                for (int K=0; K<=B.GetLength(1)-1; K++)
                {
                    Total=0;
                    for (int J=0; J<=A.GetLength(1)-1; J++)
                        Total=Total+A[I, J] * B[J, K];
                    C[I, K]=Total;
                }
            }
            return C;
        }
    }
    static void Main(string[] args)
    {
        int[,] A={ { 1, 2, 3 }, { 4, 0, 2 } };
        int[,] B={ { 1, 0 }, { 2, 2 }, { 3, 3 } };
        int [,] C=Multiplica(A, B);
        for (int I=0; I<=A.GetLength(0)-1; I++)
            for (int J=0; J<=B.GetLength(1)-1; J++)

                Console.WriteLine("{0, 4}", C[I, J]);
        Console.WriteLine();
    }
}

```

? EXERCÍCIO 17: SUBTRACÇÃO DO MENOR POR LINHA

```

using System;
namespace PF17
{
    class SubtracaoMenores
    {
        static void Ler(int[,] A)
        {
            for (int I=0; I<=A.GetLength(0)-1; I++)
                for (int J=0; J<=A.GetLength(1)-1; J++)
                    {
                        Console.WriteLine("Digite o elemento A[{0},{1}] ",I,J);
                        A[I, J] =Convert.ToInt16(Console.ReadLine());
                    }
        }
        static void Escrever(int[,] A)
        {
            for (int I=0; I<=A.GetLength(0)-1; I++)
                for (int J=0; J<=A.GetLength(1)-1; J++)
                    Console.WriteLine("{0, 4}", A[I, J]);
        }
        static void Subtrair(int[,] A)
        {
            int Menor;
            for (int I=0; I<=A.GetLength(0)-1; I++)
            {
                Menor=MenorLinha(I, A);
                for (int J=0; J<=A.GetLength(1)-1; J++)
                    A[I,J]-=Menor;
            }
        }
        static int MenorLinha(int I, int[,] A)
        {
            int Menor=0;

```

```
        for (int J=1; J<=A.GetLength(1)-1; J++)
            if (A[I,Menor] > A[I, J])
                Menor=J;
    return A[I,Menor];}
static void Main(string[] args)
{int N=3;
 int[,] A=new int[N ,N];
 Ler (A);
 Subtrair(A);
 Escrever(A);
}}}
```

? EXERCÍCIO 18: CONTENTOR PARA PESQUISA LINEAR

```
using System;
namespace PF18
{class ContentoresParaPesquisa
 {static void Main(string[] args)
  {int X, N=10, I;
   int [] A= new int[N+1];
   for (I=0; I<=N-1; I++)
       A[I]=200+100 *I+ 150;
   Console.Write("Digite o valor a procurar ");
   X=Convert.ToInt16(Console.ReadLine());
   I=Pesquisas.PesquisaLinear(A, X);
   if (I!=N)
       Console.WriteLine("{0} é o elemento índice {1} do vector",
                           X,I);
   else
       Console.WriteLine("{0} não existe no vector!!", X);
  }}}
```

? EXERCÍCIO 19: CONTENTOR PARA PESQUISAS

```
using System;
namespace PF19
{class ContentoresParaPesquisas
 {static void Main(string[] args)
  {int X, N=10, I;
   int [] A= new int[N+1];
   for (I=0; I<=N-1; I++)
       A[I]=200+100 *I;
   Console.Write("Digite o valor a procurar ");
   X=Convert.ToInt16(Console.ReadLine());
   Console.Write("Pesquisa Binária (Sim/...?)");
  }}
```

```
String Perg=Console.ReadLine().ToUpper();
if (Perg != "SIM")
    I=Pesquisas.PesquisaLinear(A, X);
else
    I=Pesquisas.PesquisaBinaria(A, X);
if (I < N)
    Console.WriteLine("{0} é o elemento índice {1} do vector", X,
                      I);
else
    Console.WriteLine("{0} não existe no vector!!", X);
}}}
```

? EXERCÍCIO 20: PÚBLICOS E PRIVADOS

```
using System;
namespace PF20
{
    class ClasseC
    {
        public static string Filosofo()
        {
            return "\nSócrates";
        }
        private static string OutroFilosofo()
        {
            return "Kant";
        }
    }
    class ClasseB
    {
        public static string Maximal()
        {
            string M1="Não sou ateniense nem grego, ";
            string M2=Maxima2();
            string M3=ClasseC.Filosofo();
            //Console.WriteLine(ClasseC.OutroFilosofo());
            return M1+M2+M3;
        }
        private static string Maxima2()
        {
            return "mas sim um cidadão do mundo";
        }
    }
    class ClasseA
    {
        static void Main(string[] args)
        {
            Console.WriteLine(ClasseB.Maximal());
            //Console.WriteLine(ClasseC.OutroFilosofo());
        }
    }
}
```