

## Ficha de Trabalho 1: Resolução

### GRUPO I

#### 1. Resolução:

```
string x = "O rato roeu a rolha...";
string y = "...da garrafa do rei da Rússia";

// a)
Console.WriteLine("A string x tem {0} carateres", x.Length);

// b)
if (x.Contains("rolha"))
{
    Console.WriteLine("A string x contém a palavra \"rolha\"");
}

if (y.Contains("rolha"))
{
    Console.WriteLine("A string y contém a palavra \"rolha\"");
}

// c)
Console.WriteLine("A palavra roeu começa no índice {0}", x.IndexOf("roeu"));

// d)
Console.WriteLine("Índice da última ocorrência do carater 'i': {0}",
y.LastIndexOf('i'));

// e)
string x2 = "Análise da frase \"\"";
x2 = x2.Insert(18, x);
Console.WriteLine("String x2: {0}", x2);

// f)
string z = x.Replace(' ', '_');
Console.WriteLine("String z: {0}", z);

// g)
Console.WriteLine("String y em maiúsculas: {0}", y.ToUpper());

// h)
x = x.ToUpper();
Console.WriteLine("String x em maiúsculas: {0}", x);

// i)
Console.WriteLine("Primeiro carater da string x: {0}", x.Substring(0, 1));

// j)
Console.WriteLine("Último carater da string x: {0}", x.Substring(x.Length - 1, 1));

// k)
Console.WriteLine("Todos os carateres da string y exceto o primeiro: {0}",
y.Substring(1));

// l)
```

```
Console.WriteLine("3º carater da string y: {0}", y.Substring(3, 1));

// m)
Console.WriteLine("3 primeiros caracteres da string x: {0}", x.Substring(0, 3));

// n)
Console.WriteLine("3 últimos caracteres da string x: {0}", x.Substring(x.Length - 3));
```

## 2. Resolução:

```
string nome, primeiroNome, apelido;
int indiceUltimoEspaco;

Console.Write("Nome: ");
nome = Console.ReadLine();

primeiroNome = nome.Substring(0, nome.IndexOf(' '));
indiceUltimoEspaco = nome.LastIndexOf(' ');
apelido = nome.Substring(indiceUltimoEspaco + 1, nome.Length - indiceUltimoEspaco - 1);

Console.WriteLine("\nPrimeiro nome: {0}", primeiroNome);
Console.WriteLine("Apelido: {0}", apelido);
```

## GRUPO II

## 1. Resolução:

```
string s;

Console.Write("Insira uma string: ");
s = Console.ReadLine();

for (int i = 0; i < s.Length; ++i)
{
    Console.Write(s[i]);
    if (i < s.Length - 1)
    {
        Console.Write("-");
    }
}
```

## 2. Resolução:

```
string s;

Console.Write("Insira uma string: ");
s = Console.ReadLine();

for (int i = 0; i < s.Length; ++i)
{
    Console.Write(s[i]);
    if ((i < s.Length - 1) && (s[i + 1] != ' ') && (s[i] != ' '))
    {
        Console.Write("-");
    }
}
```

## Ficha de Trabalho 1

```
}  
}
```

### 3. Resolução:

```
string s;  
int comprimento = 0;  
  
Console.Write("Insira uma string: ");  
s = Console.ReadLine();  
  
foreach (char c in s)  
{  
    ++comprimento;  
}  
  
Console.WriteLine($"A string possui {comprimento} caracteres");
```

### 4. Resolução:

```
string s;  
  
Console.Write("Insira uma string: ");  
s = Console.ReadLine();  
  
for (int i = s.Length - 1; i >= 0; --i)  
{  
    Console.Write(s[i]);  
}
```

### 5. Resolução:

```
string s;  
int palavras = 1;  
  
Console.Write("Insira uma string: ");  
s = Console.ReadLine();  
  
for (int i = 0; i < s.Length - 1; ++i)  
{  
    if ((s[i] == ' ') && (s[i + 1] != ' '))  
    {  
        ++palavras;  
    }  
}  
  
Console.WriteLine($"A string possui {palavras} palavras");
```

## 6. Resolução:

```
string s;  
char carater;  
int ocorrencias = 0;  
  
Console.Write("Insira uma string: ");  
s = Console.ReadLine();  
  
Console.Write("Insira o carater a pesquisar: ");  
carater = Convert.ToChar(Console.ReadLine());  
  
for (int i = 0; i < s.Length - 1; ++i)  
{  
    if (s[i] == carater)  
    {  
        ++ocorrencias;  
    }  
}  
  
Console.WriteLine($"O carater '{carater}' ocorre {ocorrencias} vezes na string");
```

## 7. Resolução:

```
string username = "darth_vader", password = "12345";  
string user, pass;  
  
Console.Write("Insira o username: ");  
user = Console.ReadLine();  
  
Console.Write("Insira a password: ");  
pass = Console.ReadLine();  
  
if ((username == user) && (password == pass))  
{  
    Console.WriteLine("Acesso concedido: introduziu os dados corretos");  
}  
else  
{  
    Console.WriteLine("Acesso negado: username e/ou password incorretos!");  
}
```

## GRUPO III

### 1. Resolução:

```
string s = "C# is a general-purpose, multi-paradigm programming language  
encompassing strong typing, lexically scoped, imperative, declarative, functional,  
generic, object-oriented (class-based), and component-oriented programming  
disciplines. It was developed around 2000 by Microsoft as part of its .NET  
initiative, and later approved as an international standard by Ecma (ECMA-334) and  
ISO (ISO/IEC 23270:2018). Mono is the name of the free and open-source project to  
develop a compiler and runtime for the language. C# is one of the programming  
languages designed for the Common Language Infrastructure (CLI). C# was designed by  
Anders Hejlsberg, and its development team is currently led by Mads Torgersen. The  
most recent version is 8.0, which was released in 2019 alongside Visual Studio 2019  
version 16.3.";
```

## Ficha de Trabalho 1

```
string s2 = s.ToLower().Replace(' ', '\n').Replace(',', '\0').Replace('.', '\0');  
Console.WriteLine(s2);
```

### 2. Resolução:

```
public static bool ValidarIPv4(string endereco)  
{  
    if (endereco == string.Empty)  
    {  
        return false;  
    }  
  
    int indicePonto1, indicePonto2, indicePonto3;  
    string octeto1, octeto2, octeto3, octeto4;  
  
    // Verificar que os 3 pontos existem (e obter o respetivo índice)  
  
    // Obter o índice do 1º ponto  
    if ((indicePonto1 = endereco.IndexOf('.')) == -1)  
    {  
        return false;  
    }  
  
    // Obter o índice do 2º ponto  
    if ((indicePonto2 = endereco.IndexOf('.', indicePonto1 + 1)) == -1)  
    {  
        return false;  
    }  
  
    // Obter o índice do 3º ponto  
    if ((indicePonto3 = endereco.IndexOf('.', indicePonto2 + 1)) == -1)  
    {  
        return false;  
    }  
  
    // Obter octetos  
    octeto1 = endereco.Substring(0, indicePonto1);  
    octeto2 = endereco.Substring(indicePonto1 + 1, indicePonto2 - (indicePonto1 +  
1));  
    octeto3 = endereco.Substring(indicePonto2 + 1, indicePonto3 - (indicePonto2 +  
1));  
    octeto4 = endereco.Substring(indicePonto3 + 1, endereco.Length - (indicePonto3 +  
1));  
  
    // Validar os octetos  
    return (ValidarOcteto(octeto1) && ValidarOcteto(octeto2) &&  
ValidarOcteto(octeto3) && ValidarOcteto(octeto4));  
}  
  
public static bool ValidarOcteto(string octeto)  
{  
    int octetoNumerico;  
  
    if (int.TryParse(octeto, out octetoNumerico))  
    {  
        if (octetoNumerico >= 0 && octetoNumerico <= 255)
```

## Ficha de Trabalho 1

```
{  
    if ((octetoNumerico > 9) && (octeto[0] == '0'))  
    {  
        return false;  
    }  
    return true;  
}  
}  
return false;  
}
```