

Olá, terráqueos! Meu nome é Lucas Eduardo de Freitas, atualmente trabalho como Desenvolvedor Fullstack JR, e estou sempre buscando ampliar meus horizontes, para poder evoluir cada dia mais! Neste documento, estarei lhes apresentando a Documentação referente ao terceiro exercício do Desafio de Lógica de Programação. Então, sem mais delongas, vamos lá!

Questão 03

Um texto precisa ser encriptado usando o seguinte esquema. Primeiro, os espaços são removidos do texto. Então, os caracteres são escritos em um grid, no qual as linhas e colunas tem as seguintes regras:

$$\sqrt{T} \leq \text{linha} \leq \text{coluna} \leq \sqrt{T}$$

- Considere T, como o tamanho do texto.
- Se certifique de que linhas x colunas $\geq T$.
- Se múltiplos grids satisfazem as condições, escolha aquele com a menor área.

Escreva um algoritmo que ao receber uma string s, mostre a mensagem encriptada de acordo com as regras descritas.

Conforme visualizamos no enunciado da terceira questão, teremos que criar um algoritmo que Encripte o texto digitado, formando uma GRID com linhas e colunas, seguindo as validações solicitadas.

Com isso, criaremos um [Console Application](#), que possibilitará este processo.

Após criar nossa aplicação, vamos criar a classe Encriptador.cs, que possuirá os valores iniciais e um método que retornará o texto digitado sem considerar os espaços. O código ficará da seguinte forma:

```
namespace Questao3
{
    2 referências
    public class Encriptador
    {
        string texto = "tenha um bom dia";

        2 referências
        public string TextoSemEspacos()
        {
            string textoSemEspaco = Regex.Replace(texto, @"\s", "");

            return textoSemEspaco;
        }
    }
}
```

A ferramenta `Regex.Replace` vem da biblioteca "RegularExpressions", que possibilita a recolocação de informações.

Na classe Program.cs, será onde todas as nossas ações serão executadas e mostradas em tela, conforme abaixo:

```
namespace Questao3
{
    0 referências
    class Program
    {
        0 referências
        static void Main(string[] args)
        {
            Console.WriteLine("QUESTÃO 3 - ENCRIPtar TEXTOS\n");
            Encriptador encriptador = new();

            Console.Write("Encriptando");
            Thread.Sleep(600);
            Console.Write(".");
            Thread.Sleep(700);
            Console.Write(".");
            Thread.Sleep(800);
            Console.WriteLine(".\n");
            Thread.Sleep(900);

            string texto = encriptador.TextoSemEspacos() ;

            int quantidadeCaracteres = encriptador.TextoSemEspacos().Length;

            int separador = 0;

            int linha = (int)Math.Floor(Math.Sqrt(quantidadeCaracteres));

            int coluna = (int)Math.Ceiling(Math.Sqrt(quantidadeCaracteres));

            if (linha * coluna < quantidadeCaracteres)
            {
                linha = coluna;
            }

            char[,] matrizBase = new char[linha, coluna];
```

```
        for (int i = 0; i < linha; i++)
        {
            for (int j = 0; j < coluna; j++)
            {
                if (separador < texto.Length)
                {
                    matrizBase[i, j] = texto[separador];
                    separador++;
                }
            }
        }

        for (int i = 0; i < linha; i++)
        {
            for (int j = 0; j < coluna; j++)
            {
                if (matrizBase[i, j] == 0)
                {
                    break;
                }
                Console.Write(matrizBase[j, i]);
            }
            Console.Write(" ");
        }

        Console.WriteLine(" \n");
        Thread.Sleep(1000);

        Console.WriteLine("FIM DA QUESTÃO 3\n");

        Console.WriteLine("=====\n");
```

Como podemos visualizar, no início, há uma apresentação simples da nossa aplicação, seguida pela instanciação de uma variável para que seja possível utilizar os métodos da classe Encriptador.cs, logo após isso, instanciaremos a variável quantidadeCaracteres que definirá a quantidade de letras contidas na frase digitada. Após isso, definiremos as linhas e as colunas acompanhadas de nosso Separador, depois teremos uma validação sobre o tamanho das linhas e das colunas. No primeiro “FOR” definiremos os separadores que definirão os limites de nossa matriz, no segundo “FOR”, utilizaremos o Separador para montar e mostrar em tela nossa matriz.

A resposta com base na frase digitada será a seguinte (seguindo o primeiro exemplo para facilitar):

```

Microsoft Visual Studio
QUESTÃO 3 - ENCRYPTAR TEXTOS

Encriptando...

taoa eum nmd h

FIM DA QUESTÃO 3

=====

```

PROVA REAL:

Temos a nossa frase “tenha um bom dia”, então, inicialmente, removeremos todos os espaços contidos na mesma, ficará da seguinte forma: “tenhaumbomdia”. Com isso, poderemos contar a quantidade de caracteres, que é igual a 13!

Como a explicação propriamente dita ficaria um tanto quanto abstrata, quando explicada em um documento, resolvi “desenhar” nossa matriz e depois juntá-la em uma linha, para demonstrar a resposta correta (de acordo com o separador calculado com base no enunciado, será uma matriz 4x4):

TENHAUMBOMDIA =	1	2	3	4	
	T	/	E	/	N / H
	A	/	U	/	M / B
	O	/	M	/	D / I
	A				
	= TAOA EUM NMD HBI				

OBS: Durante a prova real, pude perceber que havia duas letras faltando em meu resultado (o B e o I na ultima coluna). Com isso, ajustei o código e, agora, a resposta em tela está devidamente correta:

```
Selecionar Console de Depuração do Microsoft
QUESTÃO 3 - ENCRIPITAR TEXTOS

Encriptando...

taoa eum nmd hbi

FIM DA QUESTÃO 3

=====
```

A solução foi inverter os índices em um dos FOR da apresentação em tela da matriz, pois o valor por ventura se tornava 0 e ignora estas duas últimas letras:

```
for (int i = 0; i < linha; i++)
{
    for (int j = 0; j < coluna; j++)
    {
        if (matrizBase[j, i] == 0)
        {
            break;
        }
        Console.Write(matrizBase[j, i]);
    }
    Console.Write(" ");
}
```

Este ajuste será incluído no commit desta Documentação no GitHub!

=====

Maravilha! Os resultados se equivalem, seguindo a mesma explicação de nossa prova real.

Agradeço pela atenção até o fim desta documentação, decidi escrevê-la de forma mais pessoal, pois assim facilitaria a interpretação de quaisquer grupos de pessoas!

Atenciosamente, Lucas Eduardo de Freitas

</end>