

Target Sistemas Respostas

QUESTÃO 1

O valor final da variável SOMA será 91.

O trecho de código apresentado utiliza um laço de repetição do tipo "enquanto" para calcular a soma dos números inteiros de 1 a 13.

Inicialmente, as variáveis INDICE, SOMA e K são inicializadas com os valores 13, 0 e 0, respectivamente. Em seguida, o laço de repetição "enquanto" é executado enquanto a variável K for menor que INDICE. Dentro do laço, o valor de K é incrementado em 1 a cada iteração e o valor de SOMA é atualizado somando-se a ele o valor de K.

Dessa forma, o laço executa 13 iterações, somando os números inteiros de 1 a 13. Ao final do laço, o valor final da variável SOMA será a soma de todos esses números, que pode ser calculado usando uma progressão aritmética. Portanto, o valor final da variável SOMA será 91.

QUESTÃO 2

O código foi implementado em C# no Visual Studio na Solução TargetSistemasRibeirãoPreto, que se encontra neste repositório, porém, também, segue o código abaixo:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Fibonacci
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.Write("Digite o número que deseja buscar na sequência de Fibonacci: ");
14             int numeroProcurado = int.Parse(Console.ReadLine()); // lê o número informado pelo usuário
15
16             Console.WriteLine();
17
18             int a = 0; // primeiro número da sequência
19             int b = 1; // segundo número da sequência
20
21             while (a <= numeroProcurado) // enquanto não ultrapassar o número procurado
22             {
```

```

23         if (a == numeroProcurado) // se o número procurado for encontrado
24         {
25             Console.WriteLine(numeroProcurado + " pertence à sequência de
                Fibonacci.");
26             break;
27         }
28
29         int temp = b;
30         b = a + b; // calcula o próximo número da sequência
31         a = temp; // atualiza o valor de a com o valor anterior de b
32     }
33
34     if (a > numeroProcurado) // se ultrapassar o número procurado
35     {
36         Console.WriteLine(numeroProcurado + " não pertence à sequência de
                Fibonacci.");
37     }
38
39     Console.ReadKey(); // aguarda o usuário pressionar uma tecla para encerrar o
        programa
40     }
41 }
42 }

```

QUESTÃO 3

a) O próximo número é 9. A sequência é formada por números ímpares, aumentando de 2 em 2.

$$\begin{aligned}
 b_n &= 2n - 1 \\
 b_1 &= 2 \cdot 1 - 1 = 1 \\
 b_2 &= 2 \cdot 2 - 1 = 3 \\
 b_3 &= 2 \cdot 3 - 1 = 5 \\
 b_4 &= 2 \cdot 4 - 1 = 7 \\
 b_5 &= 2 \cdot 5 - 1 = 9 \text{ (próximo número)}
 \end{aligned}$$

b) O próximo número é 128. A sequência é formada por potências de 2, dobrando a cada termo.

$$\begin{aligned}
 c_n &= 2^n \\
 c_1 &= 2^1 = 2 \\
 c_2 &= 2^2 = 4 \\
 c_3 &= 2^3 = 8 \\
 c_4 &= 2^4 = 16 \\
 c_5 &= 2^5 = 32 \\
 c_6 &= 2^6 = 64 \\
 c_7 &= 2^7 = 128 \text{ (próximo número)}
 \end{aligned}$$

c) O próximo número é 49. A sequência é formada pelos quadrados dos números naturais, começando em 0.

$$\begin{aligned}d_n &= n^2 \\d_0 &= 0^2 = 0 \\d_1 &= 1^2 = 1 \\d_2 &= 2^2 = 4 \\d_3 &= 3^2 = 9 \\d_4 &= 4^2 = 16 \\d_5 &= 5^2 = 25 \\d_6 &= 6^2 = 36 \\d_7 &= 7^2 = 49 \text{ (próximo número)}\end{aligned}$$

d) O próximo número é 100. A sequência é formada pelos quadrados dos números pares, começando em 2.

$$\begin{aligned}e_n &= (2n)^2 \\e_1 &= (2 \cdot 1)^2 = 4 \\e_2 &= (2 \cdot 2)^2 = 16 \\e_3 &= (2 \cdot 3)^2 = 36 \\e_4 &= (2 \cdot 4)^2 = 64 \\e_5 &= (2 \cdot 5)^2 = 100 \text{ (próximo número)}\end{aligned}$$

e) O próximo número é 13. A sequência é formada pela soma dos dois números anteriores, começando com 1 e 1 (sequência de Fibonacci).

$$\begin{aligned}f_n &= f_{n-1} + f_{n-2} \\f_1 &= 1 \\f_2 &= 1 \\f_3 &= 1 + 1 = 2 \\f_4 &= 1 + 2 = 3 \\f_5 &= 2 + 3 = 5 \\f_6 &= 3 + 5 = 8 \\f_7 &= 5 + 8 = 13 \text{ (próximo número)}\end{aligned}$$

f) O próximo número é 20. A sequência parece não seguir um padrão claro, mas é possível notar que cada número é o número anterior mais algum valor. Podemos observar que o valor que é adicionado começa em 8 e diminui por 2 a cada termo. Portanto, temos a seguinte sequência: 2, 10, 12, 16, 17, 18, 19, 20.

QUESTÃO 4

Os dois veículos (o carro e o caminhão) estão se aproximando um do outro em direção ao ponto de encontro na estrada. Quando eles se encontram, ambos estarão a mesma distância da cidade de Ribeirão Preto.

No entanto, a distância que cada um dos veículos percorreu para chegar ao ponto de encontro é diferente. O carro estava mais distante de Ribeirão Preto, então percorreu a distância em velocidade maior para chegar ao ponto de encontro. O caminhão, por sua vez, estava mais perto de Ribeirão Preto, então percorreu uma distância menor em uma velocidade menor.

Mas, apesar de terem percorrido distâncias diferentes para chegar ao ponto de encontro, quando eles se cruzarem estarão na mesma distância da cidade de Ribeirão Preto.

QUESTÃO 5

O código foi implementado em C# no Visual Studio na Solução TargetSistemasRibeirãoPreto, que se encontra neste repositório, porém, também, segue o código abaixo:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace InverteString
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             // Solicita ao usuário que informe uma string
14             Console.Write("Informe uma string: ");
15
16             // Recebe a string informada pelo usuário
17             string input = Console.ReadLine();
18
19             // Converte a string para um array de caracteres
20             char[] charArray = input.ToCharArray();
21
22             // Define a variável de índice do início do array
23             int start = 0;
24
25             // Define a variável de índice do final do array
26             int end = charArray.Length - 1;
27
28             // Enquanto o índice do início for menor que o índice do final
29             while (start < end)
30             {
31                 // Define a variável temporária para armazenar o valor do índice do início
32                 char temp = charArray[start];
33
34                 // Define o valor do índice do início como o valor do índice do final
35                 charArray[start] = charArray[end];
36
37                 // Define o valor do índice do final como o valor armazenado na variável temporária
38                 charArray[end] = temp;
```

```
39
40         // Incrementa o índice do início
41         start++;
42
43         // Decrementa o índice do final
44         end--;
45     }
46
47     // Converte o array de caracteres de volta para uma string
48     string output = new string(charArray);
49
50     // Exibe a string invertida na tela
51     Console.WriteLine("String invertida: " + output);
52
53     Console.ReadLine();
54 }
55 }
56 }
```
