

**Questão 1** .....

Programas são construídos com uso de expressões compostas de operações sobre dados que podem ser valores literais ou definidos por meio de variáveis. A avaliação de uma expressão em tempo de execução do programa resulta em um valor que pode ser atribuído a uma variável ou pode ser usado para tomada de decisão quanto ao fluxo do programa. Dois tipos bastante usados de expressão são a expressão numérica e a expressão lógica. Enquanto uma expressão numérica relaciona dados numéricos e produz um resultado também numérico, uma expressão lógica relaciona dados lógicos (com valor verdadeiro ou falso) e produz um resultado também lógico.

Tomando como referência o texto apresentado e considerando uma linguagem fortemente tipada, como Java, avalie as afirmações a seguir.

- I. O resultado da avaliação de uma expressão lógica pode ser atribuído a uma variável numérica.
- II. Em uma expressão numérica podem ser usados operadores de soma, subtração, multiplicação e divisão.
- III. O fluxo de execução de um programa pode ser decidido a partir do resultado de uma expressão lógica.

É correto o que se afirma em:

- ☐ I, apenas
- ☐ II, apenas
- ☐ I e II, apenas
- ☐ II e III, apenas
- ☐ I, II e III

**Questão 2** .....  
O que é impresso com a execução do código Java abaixo?

```
public class T {  
    public static void main(String args[]) {  
        int k = 64;  
        int x = 100;  
        int r = 0;  
        if ( x < k )  
            r = k - x;  
        else  
        {  
            k = k/2; 32  
            r = x - k; 100-32  
        }  
        System.out.println(r);  
    }  
} 68
```

**Questão 3** .....  
O que é impresso com a execução do código Java abaixo?

```
public class T {  
    public static void main(String args[]) {  
        int k = 32;  
        int x = 2;  
        int y = 40;  
        int r = 0;  
        while ( x < y && k > 1 )  
        {  
            x = x + k;  
            k = k/2;  
            r++;  
        }  
        System.out.println(r);  
    }  
}
```

$$\begin{array}{r} 2 + 32 = 34 \\ \hline 16 \\ 34 + 16 = 50 \\ 8 \end{array}$$

1

2

2

**Questão 4** .....  
O que é impresso com a execução do código Java abaixo?

```
public class T {  
    public static void main(String args[]) {  
        int x = 5;  
        int r = 2;  
        for ( int i = 0; i < x; i++ )  
        {  
            r = r * 2;  
            r--;  
        }  
        System.out.println(r);  
    }  
}
```

Handwritten annotations in blue:

- Next to `int x = 5;`: `3`
- Next to `int r = 2;`: `5`
- Next to `for ( int i = 0; i < x; i++ )`: `5 5`
- Next to `r = r * 2;`: `17`
- Next to `r--;`: `33`
- Next to the closing brace of the `main` method: `33`

**Questão 5** .....  
O que é impresso com a execução do código Java abaixo?

```
public class T {  
    public static void main(String args[]) {  
        int x = 10;  
        int r = 0;  
        int k = x % 4;  
        switch ( k ) {  
            case 0: r = 7; break;  
            case 1: r = 11; break;  
            case 2: r = 13; break;  
            case 3: r = 17; break;  
        }  
        System.out.println(r);  
    }  
}
```

10 4  
2 2  
—  
2

13

**Questão 6** .....

Considere o seguinte trecho de código Java contendo o comando de desvio **if**:

```
int a = 10;
int b = 8;
boolean c = false;
if (_____)
    System.out.println("verde");
else
    System.out.println("azul");
```

Uma expressão lógica que pode ser usada para preencher o código do comando **if** tal que seja impressa a palavra **verde** é:

- ☐ a < b && c —
- ☐ a < b || c
- ☒ a > b || c
- ☐ a > b && c —
- ☐ a > c && c —

**Questão 7** .....

Considere o seguinte trecho de código Java contendo o comando de repetição while que implementa o Algoritmo de Divisão Euclidiana, isto é, realiza a divisão de um número inteiro (n) por um número natural (k), resultando num quociente e num resto, tal que o resto é sempre menor que o divisor k:

```
int n = 20;
int k = 3;
int quociente = 0;
int resto = n;
while (_____)
{
    quociente++;
    resto = _____;
}
System.out.println("Quociente da divisão = " + quociente);
System.out.println("Resto da divisão = " + resto);
```

Para que o código calcule e imprima o quociente e o resto da divisão inteira de n por k, as duas lacunas no código devem ser preenchidas, na ordem, pelos códigos:

- ☐ resto >= k E resto = resto - k
- ☐ resto > k E resto = resto - k
- ☐ resto > k E resto = resto + k
- ☒ resto < k E resto = resto + k
- ☐ resto <= k E resto = resto - k

**Questão 8** .....  
Modifique o código da classe Q1 (quadro abaixo) de forma que os valores **a,b,c** impressos ao final da sua execução sejam os mesmos valores **x,y,z** fornecidos como entrada, mas em ordem **crescente**.

```
import java.util.Scanner;
public class Q1 {
    public static void main(String args[])
    {
        Scanner teclado = new Scanner(System.in);

        System.out.print("Digite x: ");
        float x = teclado.nextFloat();

        System.out.print("Digite y: ");
        float y = teclado.nextFloat();

        System.out.print("Digite z: ");
        float z = teclado.nextFloat();

        float a = 0, b = 0, c = 0;

        // INSIRA AQUI O SEU CÓDIGO

        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
    }
}
```





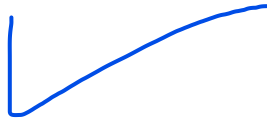
**Questão 9** .....

Modifique o código da classe Q2 (quadro abaixo) de forma que, ao final da sua execução, seja impressa uma mensagem dizendo se o número fornecido como **chave** pertence ou não ao vetor **temperaturas**. Caso pertença, deve ainda ser impresso o índice do vetor onde foi encontrado. O código deve ser escrito com uso do comando **while** e não pode usar o comando **break**.

```
import java.util.Scanner;
public class Q2 {
    public static void main(String[] args)
    {
        Scanner teclado = new Scanner(System.in);
        int[ ] temperaturas = new int[10];
        for (int i = 0; i < 10; i++)
        {
            System.out.print("Digite um número inteiro: ");
            temperaturas[i] = teclado.nextInt();
        }
        System.out.print("Digite um número para procurar: ");
        int chave = teclado.nextInt();

        // INSIRA O SEU CÓDIGO AQUI

    }
}
```



**Questão 10** .....  
Modifique o código da classe Q1 (quadro abaixo) no local indicado de forma que, ao final da execução, imprima os valores mínimo e máximo armazenados no vetor **numeros**.

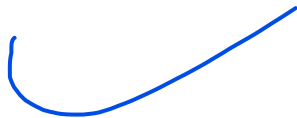
```
import java.util.Scanner;

public class Q1 {
    public static void main(String args[])
    {
        Scanner teclado = new Scanner(System.in);

        int N = 0;
        do
        {
            System.out.print("Digite a quantidade de números: ");
            N = teclado.nextInt();
        }
        while (N < 1);

        float[] numeros = new float[N];
        for (int i = 0; i < N; i++)
        {
            System.out.print("Digite um número real: ");
            float x = teclado.nextFloat();
            numeros[i] = x;
        }

        // INSIRA O SEU CÓDIGO AQUI
    }
}
```



**Questão 11** .....

Modifique o código da classe **Q** (quadro abaixo) de forma que, ao final da sua execução, seja impressa uma mensagem dizendo se o número fornecido como **chave** pertence ou não ao vetor **temperaturas**. Caso pertença, deve ainda ser impresso o índice do vetor onde foi encontrado. O código deve ser escrito com uso do comando **while** e não pode usar o comando **break**.

```
import java.util.Scanner;
public class Q {
    public static void main(String[] args)
    {
        Scanner teclado = new Scanner(System.in);
        int[ ] temperaturas = new int[10];
        for (int i = 0; i < 10; i++)
        {
            System.out.print("Digite um número inteiro: ");
            temperaturas[i] = teclado.nextInt();
        }
        System.out.print("Digite um número para procurar: ");
        int chave = teclado.nextInt();

        // INSIRA O SEU CÓDIGO AQUI
    }
}
```



**Questão 12** .....

Escreva um programa em Java que leia uma sequência de  $n$  números reais fornecidos pelo usuário, faça a ordenação crescente da sequência e a imprima a sequência ordenada.

- O próprio valor de  $n$  deve ser fornecido pelo usuário.
- A sequência deve ser armazenada em um vetor.
- Pode ser usado qualquer algoritmo de ordenação.
- Na implementação do algoritmo de ordenação, pode ser usado o comando `while` ou o comando `for`, conforme apropriado.