

Instruções:

1. O teste contém sete questões, com pesos diferentes, totalizando 10 (dez) pontos.
2. O tempo máximo de prova é 180 minutos.

Questão 11 ponto

Complete o quadro a seguir com a impressão obtida com a execução do programa abaixo. Somente será considerada correta a resposta que estiver integralmente de acordo com o resultado esperado.

Verde

```
#include <stdio.h>

int main()
{
    int a = 10;
    int b = 4;
    int c = 8;
    if (a > c && c % b == 0)
    {
        if (b > a)
            printf("azul");
        else
            printf("verde");
    }
    else
        printf("vermelho");
    return 0;
}
```

Questão 21 ponto

Complete o quadro a seguir com a impressão obtida com a execução do programa abaixo. Somente será considerada correta a resposta que estiver integralmente de acordo com o resultado esperado.

29

```
#include <stdio.h>

int main()
{
    int t = 0;
    for (int i = 2; i <= 4; i++)
    {
        int p = i * i;
        t = t + p;
    }
    printf("%d", t);
    return 0;
}
```

Handwritten calculations for the loop:

1 ^o i = 2; 2 <= 4; i++ p = 2 * 2 = 4 t = 0 + 4	2 ^o i = 3; 3 <= 4; i++ p = 3 * 3 = 9 t = 4 + 9 = 13
3 ^o i = 4; 4 <= 4; i++ p = 4 * 4 = 16 t = 13 + 16 = 29	i = 5; 5 <= 4; i++ ↳ FALSE

Questão 3 2 pontos

Complete os quadros 1 e 2 com a impressão obtida com a execução do programa abaixo. Cada quadro corresponde a uma linha da impressão. Somente será considerada correta a resposta que estiver integralmente de acordo com o resultado esperado.

- (1) $x = 3$
- (2) $y = 256$

```
#include <stdio.h>
#include <math.h>

int main()
{
    const int MIN = 1;
    const int MAX = 10;
    int i = MIN, x = 0, y = 2;
    while (i <= MAX)
    {
        int resto = i % 3;
        switch( resto )
        {
            case 1: x++; break;
            case 2: y = pow(y, 2);
        }
        i++;
    }
    printf("%d\n", x);
    printf("%d\n", y);

    return 0;
}
```

$i = \text{MIN} = 1$ | $x = 0$
 $\text{MAX} = 10$ | $y = 2$

1x $i = \text{min} = 1; x = 0, y = 2, \text{MAX} = 10$

$R = 1 \% 3 = 1$
 case 1: $x = 1$
 $i = 2$

2x $i = \text{min} = 2, x = 1, y = 2$

$R = 2 \% 3 = 2$ $i = 3$
 case 2: $2^2 = 4$

3x $i = \text{min} = 3, x = 1, y = 4$
 $R = 3 \% 3 = 0$
 $i = 4$

4x $i = \text{min} = 4, x = 1, y = 4$

$R = 4 \% 3 = 1$

case 1: $x = 2$
 $i = 5$

5x $i = \text{min} = 5, x = 2, y = 4$

$R = 5 \% 3 = 2$
 case 2: $y = 4^2 = 16$
 $i = 6$

10x $i = \text{min} = 10, x = 3, y = 256$

$R = 10 \% 3 = 1$

Case 1: $x = 4$
 $y = 256$

6x $i = \text{min} = 6, x = 2, y = 16$

$R = 6 \% 3 = 0$

$i = 7$

7x $i = \text{min} = 7, x = 2, y = 16$

$R = 7 \% 3 = 1$

case 1: $x = 3$
 $i = 8$

8x $i = \text{min} = 8, x = 3, y = 16$

$R = 8 \% 3 = 2$

case 2: $y = 16^2 = 256$

$i = 9$

9x $i = \text{min} = 9, x = 3, y = 256$

$R = 9 \% 3 = 0$

$i = 10$

$x = 3$

$y = 256$

Questão 41 ponto

Complete o quadro a seguir com a impressão obtida com a execução do programa abaixo. Somente será considerada correta a resposta que estiver integralmente de acordo com o resultado esperado.

b, e, g, e, a

```
#include <stdio.h>

int main()
{
    char v[] = { 'b', 'e', 'r', 'i', 'n', 'g', 'e', 'l', 'a' };
    int N = sizeof(v) / sizeof(char);
    for (int i = 0; i < N; i++)
    {
        if (v[i] <= 'g')
            putchar( v[i] );
    }
    return 0;
}
```

Handwritten notes:

- TAMANHO VETOR (4x9) = 36 bytes*
- TAMANHO CHAR = 4*
- bytes ELEMENTOS*
- 0 1 2 3 4 5 6 7 8*
- quantidade d/ Elementos*
- ↳ Printa especifica para um caractere.*

98 103 { 'b', 'e', 'g', 'e', 'a' }

b ≤ g

103 e ≤ g 103

114 r ≤ g 103

105 i ≤ g 103

110 n ≤ g 103

103 g ≤ g 103

103 e ≤ g 103

108 l ≤ g 103

97 a ≤ g 103

Questão 51 ponto

Complete os quadros 1 a 3 com a impressão obtida com a execução do programa abaixo. Cada quadro corresponde a uma linha da impressão. Somente será considerada correta a resposta que estiver integralmente de acordo com o resultado esperado.

- (1) $m = 709$
- (2) $x = 4$
- (3) $y = 6$

```
#include <stdio.h>
#include <math.h>

int main()
{
    int valores[] = { 750, 580, 630, 800, 720, 680, 900, 740, 460, 830 };
    const int MAX = sizeof(valores) / sizeof(valores[0]);
    double t = 0;
    for (int i = 0; i < MAX; i++)
    {
        t = t + valores[i];
    }
    double m = t / MAX;
    int x = 0, y = 0;
    for (int i = 0; i < MAX; i++)
    {
        if (valores[i] < m) x++;
        else
            if (valores[i] > m) y++;
    }
    printf("%.2f\n", m);
    printf("%d\n", x);
    printf("%d\n", y);

    return 0;
}
```

Handwritten notes:

- vetor** (pointing to `valores`)
- TAMANHOS** (pointing to `sizeof`)
- TAMANHO do meu vetor / TAMANHO do meu primeiro valor = quantidade de elementos**
- 1°** $(i = 0; i < 10) ; t = 0$
- 2°** $0 = 0 + 750$
 $t = 750$
- 3°** $1330 = 1330 + 630$
 1960
- 4°** $1960 + 800 = 2760$
- 5°** $2760 + 720 = 3480 \dots$
- 10°** $7090 = T$
- $m = \frac{7090}{10} = 709$
- $m = 709$ (boxed)
- 1°** $750 < 709$ F
- 2°** $580 < 709$ V
- 3°** $630 < 709$ V
- 4°** $800 < 709$ F
- 5°** $720 < 709$ F
- 6°** $680 < 709$ V
- 7°** $900 < 709$ F
- 8°** $740 < 709$ F
- 9°** $460 < 709$ V
- 10°** $830 < 709$ F

RESULTADO

$m = 709$

$x = 4$

$y = 6$

Questão 61 ponto

Complete o quadro a seguir com a impressão obtida com a execução do programa abaixo. Somente será considerada correta a resposta que estiver integralmente de acordo com o resultado esperado.

250

```
#include <stdio.h>

typedef enum {SABADO, DOMINGO, SEGUNDA, TERCA, QUARTA, QUINTA, SEXTA} Dia;
               0       1       2       3       4       5       6

typedef struct
{
    Dia dia;
    double preco;
} Servico;

int main()
{
    Servico s1 = {TERCA, 100.00};
    Servico s2 = {SABADO, 150.00};
    if (s1.dia < s2.dia)
        printf("%.2f", s1.preco - s2.preco);
    else
        printf("%.2f", s1.preco + s2.preco);
    return 0;
}
```

se: {TERCA, 100} < {SABADO, 150}

else:

100 + 150 = 250

Questão 73 pontos

A Sequência de Fibonacci é uma série infinita de números inteiros, sendo que:

- O primeiro termo da série é 0.
- O segundo termo da série é 1.
- Os demais termos são iguais à soma dos dois anteriores.

Assim, os dez primeiros termos da Sequência de Fibonacci são: 0, 1, 1, 2, 3, 5, 8, 13, 21 e 34.

Formalmente, o n -ésimo termo da Sequência de Fibonacci, $F(n)$, pode ser calculado da seguinte maneira:

$$F(n) = \begin{cases} 0, & \text{se } n = 0 \\ 1, & \text{se } n = 1 \\ F(n-1) + F(n-2) & \text{se } n > 1 \end{cases} \quad (1)$$

Escreva um programa em C que calcule e retorne o n -ésimo termo da Sequência de Fibonacci, sendo n um valor inteiro fornecido pelo usuário via teclado. Embora a definição matemática seja recursiva, deve ser implementada uma versão iterativa da função, isto é, com uso de comando de repetição.

RESPOSTA:

```
#include <stdio.h>

int main()
{
    int n, primeiro = 0, segundo = 1, proximo;
    do
    {
        printf("-----Este programa mostra a sequencia de Fibonacci de acordo com n termos-----\n");
        printf("Entre com o valor de n para a sequencia: ");

        scanf("%d", &n);
    } while (n <= 0);

    for (int i = 0; i < n; i++)
    {
        if (i == 0)
        {
            printf("%d", primeiro);
            continue;
        }
        else if (i == 1);
        {
            printf("%d", segundo);
            proximo = primeiro + segundo;
            primeiro = segundo;
            segundo = proximo;
            continue;
        }
        printf("%d ", proximo);
    }
    return 0;
}
```