

3.1 Os seguintes fragmentos de programas ilustram o resultado de operadores lógicos; indique o resultado de cada um, assumindo que i , j , k são variáveis inteiras.

- | | |
|--|--|
| <p>(a) $i = 10; j = 5;$
 <code>printf("%d", !i < j);</code></p> <p>(b) $i = 2; j = 1;$
 <code>printf("%d", !!i + !j);</code></p> <p>(c) $i = 5; j = 0; k = -5;$
 <code>printf("%d", i && j k);</code></p> | <p>(d) $i = 1; j = 2; k = 3;$
 <code>printf("%d", i < j k);</code></p> <p>(e) $i = 3; j = 4; k = 5;$
 <code>printf("%d ", i < j ++j < k);</code>
 <code>printf("%d %d %d", i, j, k);</code></p> <p>(f) $i = 7; j = 8; k = 9;$
 <code>printf("%d ", i - 7 && j++ < k);</code>
 <code>printf("%d %d %d", i, j, k);</code></p> |
|--|--|

3.2 As condições expressas nos seguintes fragmentos de programa não surtem o efeito indicado nas mensagens. Para cada um deles simule a execução e indique o resultado. Em seguida corrija as condições para que correspondam à intenção expressa na mensagem.

- | | |
|--|--|
| <p>(a) <code>int n = 0;</code>
 <code>if (1 <= n <= 10)</code>
 <code>printf("n entre 1 e 10\n");</code></p> <p>(b) <code>int i = 1;</code>
 <code>if (i == 2 3)</code>
 <code>printf("i igual 2 ou 3\n");</code></p> | <p>(c) <code>int i = 2, j = 1, k = 0;</code>
 <code>if (i == j == k)</code>
 <code>printf("i,j,k iguais\n");</code></p> <p>(d) <code>int i = 1, j = 2, k = 3;</code>
 <code>if (i != (j && k))</code>
 <code>printf("i diferente j e k\n");</code></p> |
|--|--|

3.3 A instrução `if` seguinte é desnecessariamente complicada; simplifique-a para uma única atribuição usando operadores lógicos.

```
if(age >= 13)
  if(age <= 19)
    teenager = 1;
  else
    teenager = 0;
else if(age < 13)
  teenager = 0;
```

3.4 Escreva um programa que lê 3 valores inteiros correspondentes aos comprimentos de lados de um triângulo e o classifique como *equilátero* (3 lados iguais), *isósceles* (2 lados iguais) ou *escaleno* (lados todos diferentes). Exemplos de execução:

Primeiro lado: <u>3</u>	Primeiro lado: <u>3</u>	Primeiro lado: <u>3</u>
Segundo lado: <u>3</u>	Segundo lado: <u>2</u>	Segundo lado: <u>4</u>
Terceiro lado: <u>3</u>	Terceiro lado: <u>3</u>	Terceiro lado: <u>5</u>
Triângulo equilátero.	Triângulo isósceles.	Triângulo escaleno.

3.5 Escreva um programa que lê 3 valores inteiros da entrada-padrão e escreve o *valor máximo*, o *valor mínimo* e a *amplitude* (i.e. o valor máximo menos o valor mínimo). Exemplos:

Primeiro valor: <u>12</u>	Primeiro valor: <u>6</u>
Segundo valor: <u>3</u>	Segundo valor: <u>6</u>
Terceiro valor: <u>27</u>	Terceiro valor: <u>6</u>
Valor máximo: 27	Valor máximo: 6
Valor mínimo: 3	Valor mínimo: 6
Amplitude: 24	Amplitude: 0

3.6 Escreva um programa que lê 3 valores inteiros da entrada-padrão e escreve o *número de valores distintos* (1, 2 ou 3). Exemplos de execução:

Primeiro valor: <u>7</u>	Primeiro valor: <u>7</u>
Segundo valor: <u>9</u>	Segundo valor: <u>12</u>
Terceiro valor: <u>7</u>	Terceiro valor: <u>9</u>
Resposta: 2 valores distintos.	Resposta: 3 valores distintos.

3.7 Escreva um programa que lê 3 valores inteiros da entrada-padrão e escreva a *mediana*, isto é, o valor no meio quando os colocamos por ordem crescente. Assim, se os valores forem a, b, c com $a \leq b \leq c$, então a mediana será b . Exemplos de execução:

Primeiro valor: <u>7</u>	Primeiro valor: <u>7</u>
Segundo valor: <u>9</u>	Segundo valor: <u>12</u>
Terceiro valor: <u>7</u>	Terceiro valor: <u>9</u>
Mediana: 7	Mediana: 9

Sugestão: uma forma de obter a mediana é somar os 3 valores e subtrair o valor máximo e o valor mínimo; isto funciona mesmo nos casos em que alguns valores são iguais.

▷ **3.8** Baseando-se na solução do exercício 3.7 complete a definição da seguinte função para calcular a mediana de 3 inteiros a, b e c .

```
int mediana(int a, int b, int c) {
    /* calcular e retornar a mediana de a,b,c */
}
```

▷ **3.9** A linguagem C não tem um operador para potências, mas podemos calcular x^n efetuando multiplicações sucessivas (assumindo $n \geq 0$ inteiro): $x^n = \underbrace{x \times x \times \dots \times x}_{n \text{ fatores}}$.

Complete a definição da seguinte função para calcular potências.

```
int potencia(int x, int n) {
    /* calcular e retornar x*x*...*x (n fatores) */
}
```

Sugestão: utilize um ciclo `for`. Tenha atenção que $x^0 = 1$ (o produto de zero fatores dá 1).