

2.1 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.

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

2.2 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.

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

2.3 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;
```

2.4 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: 12
Segundo valor: 3
Terceiro valor: 27
Valor máximo: 27
Valor mínimo: 3
Amplitude: 24

Primeiro valor: 6
Segundo valor: 6
Terceiro valor: 6
Valor máximo: 6
Valor mínimo: 6
Amplitude: 0

2.5 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: 7
Segundo valor: 9
Terceiro valor: 7
Mediana: 7

Primeiro valor: 7
Segundo valor: 12
Terceiro valor: 9
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.

- ▷ **2.6** Baseando-se na solução do exercício 2.5 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 */  
}
```

- ▷ **2.7** 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).

2.8 Simule a execução dos seguintes fragmentos de programas e indique os resultados.

```
(a) int i = 1;
    while (i <= 128) {
        printf("%d ", i);
        i *= 2;
    }
```

```
(b) int i = 0;
    while (i*i < 25)
        i ++;
    printf("%d\n", i);
```

```
(c) int n = 9384;
    while (n > 0) {
        printf("%d\n", n%10);
        n /= 10;
    }
```

```
(d) int n = 9384;
    do {
        printf("%d\n", n);
        n /= 10;
    } while (n > 0);
```

2.9 Escreva um programa que lê uma sequência de inteiros terminada pelo valor 0 e determina o *máximo* (i.e. maior valor da sequência).

2.10 Escreva um programa que lê uma sequência de valores em vírgula-flutuante terminada pelo valor 0 e calcula a sua *média aritmética* (i.e. a soma dos valores a dividir pela sua contagem).

▷ **2.11** Escreva uma definição da função `int soma_divisores(int n)` que calcula a soma dos divisores de n inferiores a ele próprio. Por exemplo: a soma dos divisores de 12 é 16 porque os divisores de 12 menores que 12 são $\{1, 2, 3, 4, 6\}$ e $1 + 2 + 3 + 4 + 6 = 16$.

2.12 Escreva um programa que lê dois valores inteiros de numerador e denominador e escreve a fração correspondente simplificada. Exemplo:

Numerador: 56

Denominador: 32

A fração 56/32 é equivalente a 7/4

Sugestão: Para simplificar uma fração p/q basta dividir o numerador e denominador pelo máximo divisor comum de p, q .

2.13 Calcular uma raiz quadrada \sqrt{a} para $a > 0$ corresponde a encontrar a solução positiva da equação $x^2 - a = 0$. Podemos calcular aproximadamente a solução usando o “método Babilónico” (é um caso particular do *método de Newton* para resolução de equações):

1. Escolhemos uma primeira aproximação (por exemplo, $x_0 = a/2$);
2. Usando x_0 calculamos uma aproximação melhor $x_1 = \frac{1}{2}(x_0 + a/x_0)$
3. Repetimos este processo calculando x_2, x_3, \dots etc. até atingir a precisão pretendida pela a recorrência $x_{n+1} = \frac{1}{2}(x_n + a/x_n)$.

Escreva um programa que lê o valor de a e calcula 10 aproximações a \sqrt{a} pelo método acima imprimindo os valores sucessivos. *Sugestão:* utilize um ciclo `for`.

2.14 Escreva um programa que lê um inteiro não-negativo e imprime a lista de fatores primos (i.e. uma versão do comando `factor` do sistema Linux). Exemplo:

```
Numero inteiro: 24  
24 : 2 2 2 3
```

Temos então que $24 = 2 \times 2 \times 2 \times 3$.

Sugestão: utilize um ciclo em que faz divisões sucessivas por inteiros crescentes começando por 2, depois 3, etc. Note que se dividir tantas vezes quanto possível por um inteiro já não será possível dividir pelos seus múltiplos, logo vamos apenas obter os fatores primos tal como pretendido.