

Tabela 8.3: Precisão dos tipos fundamentais de ponto flutuante

| Tipo | Precisão |
|--------------------------|-------------------|
| <code>float</code> | 6 casas decimais |
| <code>double</code> | 15 casas decimais |
| <code>long double</code> | 19 casas decimais |

Fonte: Elaborada pelo autor

8.2 Exercícios

Exercício 8.1 (KING, 2008): Escreva um programa que leia 10 valores decimais, calcule o somatório e a média aritmética dos valores fornecidos e apresente o resultado. O cálculo deve ser feito por meio da função `void somatorioMedia(float a[], int n, float *somatorio, float *media)`.

Arquivo com a solução: [ex8.1.c](#)

Entrada

```
n[0]: 3
n[1]: 6
n[2]: 7.6
n[3]: 5
n[4]: 4
n[5]: 3
n[6]: 9.8
n[7]: 3
n[8]: 4
n[9]: 7
```

Saída

```
Somatorio: 52.40
Media: 5.24
```

Exercício 8.2 (KING, 2008): Escreva um programa que leia dois valores inteiros e que use a função `void trocar(int *n1, int *n2)` para trocar o valor de uma variável com a outra. Ao final, apresente a ordem original e os valores invertidos.

Arquivo com a solução: [ex8.2.c](#)

Entrada

```
n1: 6
n2: 19
```

Saída

```
Antes:
    n1: 6
    n2: 19
Depois:
    n1: 19
    n2: 6
```

Exercício 8.3 (KING, 2008): Escreva um programa que leia um valor inteiro que representa uma quantidade de tempo em segundos e que obtenha a quantidade de horas, minutos e segundos contidos nessa quantidade original. O cálculo deve ser feito por meio da função `void decompoeTempo(int totalSegundos, int *horas, int *minutos, int *segundos)`.

Arquivo com a solução: [ex8.3.c](#)

Entrada

```
Total de segundos: 12456
```

Saída

```
12456 segundo(s) corresponde(m) a:
    3 hora(s)
    27 minuto(s)
    36 segundo(s)
```

Exercício 8.4 (KING, 2008): Escreva um programa que leia um valor inteiro que representa o dia de um ano (1 a 365) e o ano em si. Não há necessidade de verificar se o dia do ano fornecido está no intervalo correto. A partir desses dados, o programa deve calcular qual é o mês e o dia do mês que correspondem ao dia do ano fornecido. Para isso, utilize as funções `void decompoeData(int diaDoAno, int ano, int *mes, int *dia)` e `bool ehBissexto(int ano)`. Lembrando que um ano bissexto é todo o ano que é divisível por 400 ou por 4, mas não por 100.

Arquivo com a solução: [ex8.4.c](#)

Entrada

```
Dia do ano: 123
Ano: 2019
```

Saída

```
0 dia 123 do ano 2019 cai no dia 3 do mes 5.
```

Entrada

```
Dia do ano: 123
Ano: 2016
```

Saída

```
0 dia 123 do ano 2016 cai no dia 2 do mes 5.
```

Exercício 8.5 (KING, 2008): Escreva um programa que leia um array de inteiros de 10 posições e um valor a mais, que será usado para verificar se o mesmo existe no conjunto fornecido. Como resultado do processamento, deve ser apresentado o primeiro índice em que se encontrou o valor desejado. Para isso, utilize a função `int buscar(const int *a, int n, int chave)`. Essa função deve retornar -1 caso o valor não seja encontrado.

Arquivo com a solução: [ex8.5.c](#)

Entrada

```
n[0]: 2
n[1]: 3
n[2]: 5
n[3]: 7
n[4]: 2
n[5]: 3
n[6]: 9
n[7]: 1
n[8]: 2
n[9]: 8
Buscar por: 3
```

Saída

O valor 3 foi encontrado na posicao 1.

Entrada

```
n[0]: 2
n[1]: 3
n[2]: 5
n[3]: 7
n[4]: 2
n[5]: 3
n[6]: 9
n[7]: 1
n[8]: 2
n[9]: 8
Buscar por: 15
```

Saída

O valor 15 nao foi encontrado.

Exercício 8.6 (KING, 2008): Escreva um programa que calcule e apresente o produto interno dos valores contidos em dois arrays de números decimais de 5 posições. Para isso, utilize a função `void produtoInterno(const double *a1, const double *a2, double *pi, int n)`. O produto interno de dois arrays corresponde a $pi[0] = a1[0] * a2[0]$, $pi[1] = a1[1] * a2[1]$, ..., $pi[n - 1] = a1[n - 1] * a2[n - 1]$.

Arquivo com a solução: [ex8.6.c](#)

Entrada

```
a1[0]: 2
a1[1]: 3
a1[2]: 4
a1[3]: 5
a1[4]: 6
a2[0]: 2
a2[1]: 2
a2[2]: 2
a2[3]: 2
a2[4]: 2
```

Saída

```
2.00 x 2.00 = 4.00
3.00 x 2.00 = 6.00
4.00 x 2.00 = 8.00
5.00 x 2.00 = 10.00
6.00 x 2.00 = 12.00
```