

```
1 // EXERCÍCIO 01
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     float horas_trabalhadas, salario_hora, salario_mensal;
10
11     cout << "Quantas horas trabalhadas nesse mês?" << endl;
12     cin >> horas_trabalhadas;
13
14     cout << "Qual seu salário por hora?" << endl;
15     cin >> salario_hora;
16
17     salario_mensal = (horas_trabalhadas * salario_hora);
18
19     cout << "Seu salário é de R$ " << salario_mensal << "\n";
20
21     return 0;
22 }
```

```
1 // EXERCÍCIO 02
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     float n1, n2, n3, resultado1, resultado2, resultado3;
10
11     cout << "Insira um número inteiro (N1)..." << endl;
12     cin >> n1;
13
14     cout << "Insira outro número inteiro (N2)..." << endl;
15     cin >> n2;
16
17     cout << "Insira um número decimal (N3)..." << endl;
18     cin >> n3;
19
20     resultado1 = (2 * n1) * (n2 / 2);
21     resultado2 = (3 * n1) + (n3);
22     resultado3 = (n3 * n3 * n3);
23
24     cout << "Produto do dobro de N1 com metade N2 = " << resultado1 << "\n";
25     cout << "Soma do triplo de N1 com N3 = " << resultado2 << "\n";
26     cout << "N3 ao cubo = " << resultado3 << "\n";
27
28     return 0;
29 }
```

```
1 // EXERCÍCIO 03
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     float horas_trabalhadas, salario_hora, salario_bruto, salario_liquido, inss, ir,
    sindicato;
10
11     cout << "Quantas horas trabalhadas nesse mês?" << endl;
12     cin >> horas_trabalhadas;
13
14     cout << "Qual seu salário por hora?" << endl;
15     cin >> salario_hora;
16
17     salario_bruto = (horas_trabalhadas * salario_hora);
18
19     ir = salario_bruto * 0.11;
20     inss = salario_bruto * 0.08;
21     sindicato = salario_bruto * 0.05;
22     salario_liquido = salario_bruto - ir - inss - sindicato;
23
24     cout << "|===== SALARIO BRUTO =====|" << "\n";
25     cout << " R$ " << salario_bruto << "\n" << "\n";
26
27     cout << "|===== DESCONTOS =====|" << "\n";
28     cout << " Imposto de Renda R$ " << ir << "\n";
29     cout << " INSS R$ " << inss << "\n";
30     cout << " Sindicato R$ " << sindicato << "\n" << "\n";
31
32     cout << "|===== SALARIO LÍQUIDO =====|" << "\n";
33     cout << " R$ " << salario_liquido << "\n";
34     cout << "|=====|" << "\n";
35
36     return 0;
37 }
```

```
1 // EXERCÍCIO 04
2
3 #include <iostream>
4 #include <cmath>
5 using namespace std;
6
7 int main () {
8     setlocale(LC_ALL, "pt_BR.UTF-8");
9
10    float area, volume_tinta, qtde_latas, preco_total, valor_total;
11
12    cout << "Qual a area a ser pintada (em m2)?" << "\n";
13    cin >> area;
14
15    volume_tinta = area/3;
16    qtde_latas = ceil(volume_tinta/18);
17    valor_total = 80 * qtde_latas;
18
19    cout << "QUANTIDADE DE LATAS NECESSÁRIAS: " << qtde_latas << "\n";
20    cout << "VALOR TOTAL: R$ " << valor_total << "\n";
21
22    return 0;
23 }
```

```
1 // EXERCÍCIO 05
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     float tamanho_arquivo, velocidade_link, tamanho_megabits, tempo_download_minutos;
10
11     cout << "Qual o tamanho do arquivo (MB - Megabytes)? " << "\n";
12     cin >> tamanho_arquivo;
13
14     cout << "Qual a velocidade de conexão (Mbps - Megabits por segundo)? " << "\n";
15     cin >> velocidade_link;
16
17     tamanho_megabits = tamanho_arquivo * 8;
18     tempo_download_minutos = (tamanho_megabits / velocidade_link)/60;
19
20     cout << "TEMPO DE DOWNLOAD: " << tempo_download_minutos << " min. \n";
21
22     return 0;
23 }
```

```
1  // EXERCÍCIO 06
2
3  #include <iostream>
4  using namespace std;
5
6  void calc_dosagem(int dosagem_mg) {
7      int qtde_gotas = dosagem_mg/25;
8      cout << "Tomar " << qtde_gotas << " gotas." << "\n";
9  }
10
11 int main () {
12     setlocale(LC_ALL, "pt_BR.UTF-8");
13     int idade;
14     float peso;
15
16     cout << "Informe a idade do paciente: " << "\n";
17     cin >> idade;
18
19     cout << "Informe o peso do paciente: " << "\n";
20     cin >> peso;
21
22     if (idade ≥ 12) {
23         if (peso ≥ 60) {
24             calc_dosagem(1000);
25         } else {
26             calc_dosagem(875);
27         }
28     }
29
30     if (idade < 12) {
31         if (peso ≤ 15) {
32             calc_dosagem(200);
33         } else if (peso > 15 && peso ≤ 30) {
34             calc_dosagem(500);
35         } else {
36             calc_dosagem(750);
37         }
38     }
39
40     return 0;
41 }
```

```
1 // EXERCÍCIO 07
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     float n1, n2, n3;
10
11     cout << "Insira o primeiro número (N1) " << "\n";
12     cin >> n1;
13
14     cout << "Insira o segundo número (N2) " << "\n";
15     cin >> n2;
16
17     cout << "Insira o terceiro número (N3) " << "\n";
18     cin >> n3;
19
20     if (n1 ≥ n2 && n1 ≥ n3) {
21         if (n2 ≥ n3) {
22             cout << n1 << ", " << n2 << ", " << n3 << "\n";
23         } else {
24             cout << n1 << ", " << n3 << ", " << n2 << "\n";
25         }
26     } else if (n2 ≥ n1 && n2 ≥ n3) {
27         if (n1 ≥ n3) {
28             cout << n2 << ", " << n1 << ", " << n3 << "\n";
29         } else {
30             cout << n2 << ", " << n3 << ", " << n1 << "\n";
31         }
32     } else {
33         if (n1 ≥ n2) {
34             cout << n3 << ", " << n1 << ", " << n2 << "\n";
35         } else {
36             cout << n3 << ", " << n2 << ", " << n1 << "\n";
37         }
38     }
39
40     return 0;
41 }
```

```
1 // EXERCÍCIO 08
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     float ladoA, ladoB, ladoC;
10
11     cout << "Insira o valor do lado A " << "\n";
12     cin >> ladoA;
13
14     cout << "Insira o valor do lado B " << "\n";
15     cin >> ladoB;
16
17     cout << "Insira o valor do lado C " << "\n";
18     cin >> ladoC;
19
20     if ((ladoA + ladoB > ladoC) && (ladoB + ladoC > ladoA) && (ladoA + ladoC > ladoB)){
21         if (ladoA == ladoB && ladoB == ladoC) {
22             cout << "Triângulo Equilátero" << "\n";
23         } else if (ladoA == ladoB || ladoA == ladoC || ladoB == ladoC) {
24             cout << "Triângulo Isósceles" << "\n";
25         } else {
26             cout << "Triângulo Escaleno" << "\n";
27         }
28     } else {
29         cout << "Não é possível formar um triângulo!" << "\n";
30     }
31
32     return 0;
33 }
```



```
1 // EXERCÍCIO 09
2
3 #include <iostream>
4 #include <math.h>
5 using namespace std;
6
7 void bashkara (float a, float b, float delta) {
8     float x1, x2;
9
10    if (delta == 0) {
11        x1 = (-b + sqrt(delta)) / 2*a;
12        cout << x1 << "\n";
13    } else {
14        x1 = (-b + sqrt(delta)) / 2*a;
15        x2 = (-b - sqrt(delta)) / 2*a;
16        cout << x1 << " e " << x2 << "\n";
17    }
18 }
19
20 int main () {
21     setlocale(LC_ALL, "pt_BR.UTF-8");
22     float a, b, c;
23
24     cout << "***** CALCULO DE EQUAÇÃO 2o GRAU *****" << "\n";
25     cout << "Formato  $ax^2 + bx + c$ " << "\n";
26
27     cout << "Insira o valor de A: ";
28     cin >> a;
29
30     if (a == 0) {
31         cout << "A equação não é de segundo grau... " << "\n";
32         return 0;
33     }
34
35     cout << "Insira o valor de B: ";
36     cin >> b;
37
38     cout << "Insira o valor de C: ";
39     cin >> c;
40
41
42     // DELTA
43     float delta = b*b - 4*a*c;
44     if (delta < 0) {
45         cout << "A equação não possui raízes reais..." << "\n";
46         return 0;
47     } else if (delta == 0) {
48         cout << "A equação possui apenas uma raiz real..." << "\n";
49         bashkara(a, b, delta);
50     } else {
51         bashkara(a, b, delta);
52     }
53
54     return 0;
55 }
```

```
1 // EXERCÍCIO 10
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     int ano;
10
11     cout << "Informe o ano para saber se ele é bissexto: " << "\n";
12     cin >> ano;
13
14     if (ano % 4 == 0) {
15         cout << "O ano de " << ano << " é bissexto!" << "\n";
16     } else {
17         cout << "O ano de " << ano << " não é bissexto!" << "\n";
18     }
19
20     return 0;
21 }
```

```
1 // EXERCÍCIO 11
2 #include <iostream>
3 #include <string>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8     int numero, centenas, dezenas, unidades;
9     string mensagem;
10
11     cout << "Informe um numero maior que zero:" << "\n";
12     cin >> numero;
13
14     if (numero ≥ 1000 || numero == 0) {
15         cout << "Número inválido!!" << "\n";
16         return 0;
17     }
18
19     if (numero ≥ 100) {
20         centenas = (numero - (numero % 100)) / 100;
21         numero = numero - centenas*100;
22
23         if (centenas > 1) {
24             mensagem = to_string(centenas) + " centenas, ";
25         } else {
26             mensagem = to_string(centenas) + " centena, ";
27         }
28     }
29
30     if (numero < 100) {
31         dezenas = (numero - (numero % 10)) / 10;
32         numero = numero - dezenas*10;
33         unidades = numero;
34
35         if (dezenas > 1) {
36             mensagem = mensagem + to_string(dezenas) + " dezenas e ";
37         } else {
38             mensagem = mensagem + to_string(dezenas) + " dezena e ";
39         }
40     }
41
42     if (numero < 10) {
43         unidades = numero;
44
45         if (unidades > 1) {
46             mensagem = mensagem + to_string(unidades) + " unidades";
47         } else {
48             mensagem = mensagem + to_string(unidades) + " unidade";
49         }
50     }
51
52     cout << mensagem << "\n";
53
54     return 0;
55 }
```

```
1 // EXERCÍCIO 12
2
3 #include <iostream>
4 #include <math.h>
5 using namespace std;
6
7 int contar_notas(int valor_nota, int valor) {
8     int numero_notas = floor(valor/valor_nota);
9     int valor_atualizado = valor - numero_notas * valor_nota;
10
11     cout << "NOTAS DE " << valor_nota << ": " << numero_notas << "\n";
12
13     return valor_atualizado;
14 }
15
16 int main() {
17     setlocale(LC_ALL, "pt_BR.UTF-8");
18
19     int valor_saque;
20
21     cout << "Qual valor a ser sacado? " << "\n";
22     cin >> valor_saque;
23
24     if (valor_saque > 600 || valor_saque < 10) {
25         cout << "Valor de saque inválido!" << "\n";
26         cout << "Min R$ 10,00 e máximo de R$ 600,00!" << "\n";
27         return 0;
28     }
29
30     if (valor_saque ≥ 100) {
31         valor_saque = contar_notas(100, valor_saque);
32     }
33
34     if (valor_saque ≥ 50) {
35         valor_saque = contar_notas(50, valor_saque);
36     }
37
38     if (valor_saque ≥ 10) {
39         valor_saque = contar_notas(10, valor_saque);
40     }
41
42     if (valor_saque ≥ 5) {
43         valor_saque = contar_notas(5, valor_saque);
44     }
45
46     if (valor_saque ≥ 1) {
47         valor_saque = contar_notas(1, valor_saque);
48     }
49
50     return 0;
51 }
```

```
1  // EXERCÍCIO 13
2
3  #include <iostream>
4  using namespace std;
5
6  int verificador_digito(string cpf, int limite) {
7      int acc = 0;
8      int verificador = limite + 1;
9
10     for (int i = 0; i < limite; i++) {
11         acc += (cpf[i] - '0') * verificador;
12         verificador--;
13     }
14
15     int resto = acc % 11;
16     if (resto < 2) {
17         return 0;
18     } else {
19         return (11 - resto);
20     }
21 }
22
23 int main () {
24     setlocale(LC_ALL, "pt_BR.UTF-8");
25
26     string cpf;
27     int verificador1, verificador2;
28
29     cout << "***** VALIDADOR DE CPF *****" << "\n";
30     cout << "Insira o número do CPF (somente números): ";
31     cin >> cpf;
32
33     // PRIMEIRA VERIFICAÇÃO:
34     verificador1 = verificador_digito(cpf, 9);
35
36     // SEGUNDA VERIFICAÇÃO:
37     verificador2 = verificador_digito(cpf, 10);
38
39     if ((cpf[9] - '0') == verificador1 && (cpf[10] - '0') == verificador2) {
40         cout << "CPF válido!" << "\n";
41     } else {
42         cout << "CPF inválido!" << "\n";
43     }
44
45     return 0;
46 }
```

```
1 // EXERCÍCIO 14
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     int numero_max, elemento_medio;
10    string resultado;
11
12    cout << "Informe o número máximo (deve ser ímpar): ";
13    cin >> numero_max;
14
15    for (int i = 1; i ≤ numero_max; i++) {
16        for (int espacoInicial = 1; espacoInicial < i; espacoInicial++) {
17            cout << " " << " ";
18        }
19
20        for (int j = i; j ≤ numero_max - i + 1; j++) {
21            cout << j << " ";
22        }
23
24        for (int espacoFinal = 1; espacoFinal < i; espacoFinal++) {
25            cout << " " << " ";
26        }
27
28        cout << "\n";
29
30        elemento_medio = (1 + numero_max) / 2;
31
32        if (i == elemento_medio) {
33            break;
34        }
35    }
36
37    return 0;
38 }
```

```
1 // EXERCÍCIO 15
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     int max_numeros_fibonacci, num_anterior = 0, num_anterior_anterior = 1, atual;
10
11     cout << "Informe a quantidade de números da série Fibonacci: " << "\n";
12     cin >> max_numeros_fibonacci;
13
14     for (int i = 0; i < max_numeros_fibonacci; i++) {
15         if (i ≤ 1) {
16             cout << i << " ";
17         } else {
18             atual = num_anterior + num_anterior_anterior;
19             cout << atual << " ";
20             num_anterior = num_anterior_anterior;
21             num_anterior_anterior = atual;
22         }
23     }
24
25     cout << "\n";
26
27     return 0;
28 }
```

```
1 // EXERCÍCIO 16
2
3 #include <iostream>
4 #include <stdint.h>
5 using namespace std;
6
7 int main () {
8     setlocale(LC_ALL, "pt_BR.UTF-8");
9     double soma_graos = 0, qtde_inserida = 1;
10
11     for (int i = 1; i ≤ 64; i++) {
12         soma_graos += qtde_inserida;
13         qtde_inserida = qtde_inserida * 2;
14     }
15
16     cout << soma_graos << " grãos." << "\n";
17     return 0;
18 }
```



```
1 // EXERCÍCIO 17
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     float total = 0, valor_produto = 0, valor_recebido, troco;
10    bool continuar = true;
11
12    while (continuar) {
13        cout<< "LOJAS TABAJARA" << "\n";
14        int numero_produto = 1;
15        do {
16
17            cout << "Produto " << numero_produto << ": R$ ";
18            cin >> valor_produto;
19            total += valor_produto;
20            numero_produto++;
21        } while (valor_produto != 0);
22
23        cout << "*****" << "\n";
24        cout << "TOTAL: R$ " << total << "\n";
25
26        do {
27            cout << "Valor recebido: R$ ";
28            cin >> valor_recebido;
29
30            if (valor_recebido < total) {
31                cout << "Valor recebido é menor que o total. Informe valor correto!" <<
32                "\n";
33            }
34        } while (valor_recebido < total);
35
36        troco = valor_recebido - total;
37
38        cout << "TROCO: R$ " << troco << "\n" << "\n";
39    }
40
41    return 0;
42 }
```

```
1 // EXERCÍCIO 18
2
3 #include <iostream>
4 using namespace std;
5
6 void calc_votos() {
7     int
8     votos_cand1 = 0,
9     votos_cand2 = 0,
10    votos_cand3 = 0,
11    votos_cand4 = 0,
12    votos_nulo = 0,
13    votos_branco = 0,
14    votos_total = 0,
15    opcao;
16    int votos_auto = 0;
17
18    string mensagem = "Voto registrado!";
19
20    do {
21        cout << "|===== ELEIÇÃO PRESIDENCIAL =====| " << "\n";
22        cout << "| 1 - TÍZIO | " << "\n";
23        cout << "| 2 - CAIO | " << "\n";
24        cout << "| 3 - SEMPRÔNIO | " << "\n";
25        cout << "| 4 - BELTRANO | " << "\n";
26        cout << "| 5 - NULO | " << "\n";
27        cout << "| 6 - BRANCO | " << "\n";
28        cout << "|=====| " << "\n";
29
30        do {
31            cout << "Para votar, escolha um número: " << "\n";
32            cin >> opcao;
33
34            switch(opcao) {
35                case 1:
36                    votos_cand1++;
37                    cout << mensagem << "\n" << "\n";
38                    break;
39
40                case 2:
41                    votos_cand2++;
42                    cout << mensagem << "\n" << "\n";
43                    break;
44
45                case 3:
46                    votos_cand3++;
47                    cout << mensagem << "\n" << "\n";
48                    break;
49
50                case 4:
51                    votos_cand4++;
52                    cout << mensagem << "\n" << "\n";
53                    break;
54
55                case 5:
56                    votos_nulo++;
```

```
57         cout << mensagem << "\n" << "\n";
58         break;
59
60     case 6:
61         votos_branco++;
62         cout << mensagem << "\n" << "\n";
63         break;
64
65     case 0:
66         cout << "Encerrando Votação..." << "\n" << "\n";
67         break;
68
69     default:
70         cout << "Opção Inválida!" << "\n";
71         break;
72     }
73
74     } while (opcao > 6 || opcao != 0);
75
76     } while (opcao != 0);
77
78     votos_total = votos_cand1 + votos_cand2 + votos_cand3 + votos_cand4 + votos_branco +
votos_nulo;
79     float nulos_sobre_validos = (1.0f * votos_branco / votos_total) * 100;
80
81     cout << "----- RESULTADO -----" << "\n";
82     cout << " Candidato 1: " << votos_cand1 << " votos" << "\n";
83     cout << " Candidato 2: " << votos_cand2 << " votos" << "\n";
84     cout << " Candidato 3: " << votos_cand3 << " votos" << "\n";
85     cout << " Candidato 4: " << votos_cand4 << " votos" << "\n";
86     cout << " Votos Nulos: " << votos_nulo << " votos" << "\n";
87     cout << " Votos em Branco: " << votos_branco << " votos" << "\n";
88     cout << " Porcentagem de votos em branco: " << nulos_sobre_validos << "%" << "\n";
89     cout << " TOTAL DE VOTOS: " << votos_total << " votos" << "\n";
90 }
91
92 int main () {
93     setlocale(LC_ALL, "pt_BR.UTF-8");
94     calc_votos();
95     return 0;
96 }
```

```
1  // EXERCÍCIO 19
2
3  #include <iostream>
4  using namespace std;
5
6  string remove_caracter(string palavra, int posicao) {
7      string nova_palavra;
8
9      for (int i = 0; i < palavra.length(); i++) {
10         if (i != posicao - 1) {
11             nova_palavra += palavra[i];
12         }
13     }
14
15     return nova_palavra;
16 }
17
18 int main () {
19     setlocale(LC_ALL, "pt_BR.UTF-8");
20     string palavra;
21     int posicao;
22
23     do {
24         cout << "Insira a string:" << "\n";
25         cin >> palavra;
26
27         cout << "Em qual posição deseja remover o caractere (1 a " << palavra.length() <<
28         ")? " << "\n";
29         cin >> posicao;
30
31         if (palavra.length() > 100 || posicao > palavra.length()) {
32             cout << "Valores inválidos... Tente novamente." << "\n";
33         }
34     } while (palavra.length() > 100 || posicao > palavra.length());
35
36     cout << "Nova palavra: " << remove_caracter(palavra, posicao) << "\n";
37
38     return 0;
39 }
```

```
1 // EXERCÍCIO 20
2
3 #include <iostream>
4 using namespace std;
5
6 string adiciona_caracter(string palavra, int posicao, char caractere) {
7     string nova_palavra;
8
9     if (posicao > palavra.length()) {
10         nova_palavra = palavra;
11         for (int i = palavra.length(); i ≤ posicao; i++) {
12             if (i ≠ posicao - 1) {
13                 nova_palavra += " ";
14             } else {
15                 nova_palavra += caractere;
16             }
17         }
18     } else {
19         for (int i = 0; i < palavra.length(); i++) {
20             if (i = posicao - 1) {
21                 nova_palavra += caractere;
22                 nova_palavra += palavra[i];
23             } else {
24                 nova_palavra += palavra[i];
25             }
26         }
27     }
28     return nova_palavra;
29 }
30
31 int main () {
32     setlocale(LC_ALL, "pt_BR.UTF-8");
33     string palavra;
34     int posicao;
35     char caractere;
36
37     do {
38         cout << "Insira a string:" << "\n";
39         cin >> palavra;
40
41         cout << "Qual caractere deseja adicionar?" << "\n";
42         cin >> caractere;
43
44         cout << "Em qual posição deseja adicionar o caractere (1 a 100)?" << "\n";
45         cin >> posicao;
46
47         if (palavra.length() > 100 || posicao ≥ 100) {
48             cout << "Valores inválidos... Tente novamente." << "\n";
49         }
50
51     } while (palavra.length() > 100 || posicao ≥ 100);
52     cout << "Nova palavra: " << adiciona_caracter(palavra, posicao, caractere) << "\n";
53     return 0;
54 }
```

```
1  // EXERCÍCIO 21
2
3  #include <iostream>
4  #include <string>
5
6
7  using namespace std;
8
9  int main () {
10     setlocale(LC_ALL, "pt_BR.UTF-8");
11
12     string pesquisa;
13     bool resultado;
14
15     string palavras[10] = {
16         "molusco", "rato", "arara", "cavalo", "galinha", "mosquito", "cachorro",
17         "girafa", "golfinho", "ouriço"
18     };
19
20     cout << "Qual palavra deseja pesquisar?" << "\n";
21     cin >> pesquisa;
22
23     for (int i = 0; i < 10; i++) {
24         if (pesquisa == palavras[i]) {
25             cout << "Resultado encontrado na posição " << i << "\n";
26             resultado = true;
27             break;
28         } else {
29             resultado = false;
30         }
31     }
32
33     if (!resultado) {
34         cout << "Pesquisa não encontrou resultados." << "\n";
35     }
36
37
38     return 0;
39 }
```

```
1 // EXERCÍCIO 22
2
3 #include <iostream>
4 #include <string>
5
6 using namespace std;
7
8 int main () {
9     setlocale(LC_ALL, "pt_BR.UTF-8");
10
11     int numeros[10], referencia, contadorMaior = 0, contadorMenor = 0, repeticao = 0;
12
13     for (int i = 0; i < 10; i++) {
14         cout << "Informe um valor (" << i + 1 << " de 10): ";
15         cin >> numeros[i];
16     }
17
18     cout << "Informe um número de referência: " << "\n";
19     cin >> referencia;
20
21     cout << "\n";
22
23     cout << "Números MAIORES que " << referencia << ":\n";
24     for (int i = 0; i < 10; i++) {
25         if (numeros[i] > referencia) {
26             cout << numeros[i] << " ";
27             contadorMaior++;
28         } else if (numeros[i] < referencia) {
29             contadorMenor++;
30         } else {
31             repeticao++;
32         }
33     }
34
35     cout << "\n";
36
37     if (contadorMaior == 0) {
38         cout << "Não há números maiores que " << referencia << "\n";
39     }
40
41     cout << "\n";
42
43     cout << "Quantidade de números MENORES que " << referencia << ":\n";
44     cout << contadorMenor << "\n";
45
46     cout << "\n";
47
48     cout << "Quantidade de números IGUAIS a " << referencia << ":\n";
49     cout << repeticao << "\n";
50
51     return 0;
52 }
```

```
1 // EXERCÍCIO 23
2
3 #include <iostream>
4 #include <vector>
5
6 using namespace std;
7
8 int main () {
9     setlocale(LC_ALL, "pt_BR.UTF-8");
10
11     int lista7[100], vetor_posicao = 0, i = 0;
12
13     while (vetor_posicao < 100) {
14         if ((i % 7 != 0) || (i + 3 % 10) == 0) {
15             lista7[vetor_posicao] = i;
16             cout << lista7[vetor_posicao] << " ";
17             vetor_posicao++;
18         }
19         i++;
20     }
21
22     cout << "\n";
23
24     return 0;
25 }
```



```
1 // EXERCÍCIO 24
2
3 #include <iostream>
4 using namespace std;
5
6 bool procura_repeticao(int numero, int lista[], int index) {
7     bool repeticao = false;
8
9     for (int j = 0; j ≤ index; j++) {
10         if (numero == lista[j] && index ≠ j) {
11             repeticao = true;
12             break;
13         }
14     }
15
16     return repeticao;
17 }
18
19 int main () {
20     setlocale(LC_ALL, "pt_BR.UTF-8");
21
22     int numeros[10], verificacao_repetido;
23
24     for (int i = 0; i < 10; i++) {
25         cout << "Insira um numero (" << i + 1 << " de 10): ";
26         cin >> verificacao_repetido;
27
28         while (procura_repeticao(verificacao_repetido, numeros, i)) {
29             cout << "Número repetido. Insira valor diferente: ";
30             cin >> verificacao_repetido;
31         }
32
33         numeros[i] = verificacao_repetido;
34     }
35
36     for (int i = 0; i < 10; i++) {
37         cout << numeros[i] << " ";
38     }
39
40     cout << "\n";
41
42     return 0;
43 }
```

```
1 // EXERCÍCIO 25
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     int lin, col;
10
11     cout << "Insira a quantidade de linhas da matriz: ";
12     cin >> lin;
13
14     cout << "Insira a quantidade de colunas da matriz: ";
15     cin >> col;
16
17     int matriz[lin][col], matriz_transposta[col][lin];
18
19     for (int i = 0; i < lin; i++) {
20         for (int j = 0; j < col; j++) {
21             cout << "Insira o valor [" << i << "][" << j << "];";
22             cin >> matriz[i][j];
23         }
24     }
25
26     //transposição
27     for (int i = 0; i < lin; i++) {
28         for (int j = 0; j < col; j++) {
29             matriz_transposta[j][i] = matriz[i][j];
30         }
31     }
32
33     cout << "MATRIZ " << lin << "x" << col << "\n";
34     for (int i = 0; i < lin; i++) {
35         for (int j = 0; j < col; j++) {
36             cout << matriz[i][j] << " ";
37         }
38         cout << "\n";
39     }
40
41     cout << "\n";
42
43     cout << "MATRIZ TRANSPOSTA " << col << "x" << lin << "\n";
44     for (int i = 0; i < col; i++) {
45         for (int j = 0; j < lin; j++) {
46             cout << matriz_transposta[i][j] << " ";
47         }
48
49         cout << "\n";
50     }
51
52     return 0;
53 }
```

```
1 // EXERCÍCIO 26
2
3 #include <iostream>
4 #include <stdlib.h> /* srand, rand */
5 #include <time.h> /* time */
6 using namespace std;
7
8 int main () {
9     setlocale(LC_ALL, "pt_BR.UTF-8");
10    srand (time(NULL));
11
12    int matriz[10][10], maiorValor = 0, lin, col;
13
14    for (int i = 0; i < 10; i++) {
15        for (int j = 0; j < 10; j++) {
16            matriz[i][j] = rand() % 899 + 100;
17            cout << matriz[i][j] << " ";
18        }
19        cout << "\n";
20    }
21
22    for (int i = 0; i < 10; i++) {
23        for (int j = 0; j < 10; j++) {
24            if (matriz[i][j] > maiorValor) {
25                maiorValor = matriz[i][j];
26                lin = i;
27                col = j;
28            }
29        }
30    }
31
32    cout << "\n";
33
34    cout << "Maior valor é " << maiorValor << ". Local: " << "linha " << lin + 1 << ",
35    coluna " << col + 1 << "\n";
36
37    return 0;
38 }
```

```
1 // EXERCÍCIO 27
2
3 #include <iostream>
4 using namespace std;
5
6 int main () {
7     setlocale(LC_ALL, "pt_BR.UTF-8");
8
9     int pulsacoesDia[24][4], pulsacaoMediaCamas[4];
10    float camaPulsacaoMediaMaior = 0, pulsacaoMedia = 0;
11
12    // Captura de dados
13    for (int i = 0; i < 24; i++) {
14        for (int j = 0; j < 4; j++) {
15            cout << "Insira a pulsação do paciente." << "\n";
16            cout << "Cama " << j + 1 << " hora: " << i + 1 << "\n";
17            cin >> pulsacoesDia[i][j];
18        }
19        cout << "\n";
20    }
21
22    // Media dos pacientes.
23    for (int j = 0; j < 4; j++) {
24        int acc = 0;
25
26        for (int i = 0; i < 24; i++) {
27            acc += pulsacoesDia[i][j];
28        }
29        pulsacaoMediaCamas[j] = acc/24;
30        cout << "Média pulsações CAMA " << j + 1 << ": " << pulsacaoMediaCamas[j] <<
31        "\n";
32    }
33
34    // Cama com maior média
35    for (int i = 0; i < 4; i++) {
36        if (pulsacaoMediaCamas[i] > pulsacaoMedia) {
37            pulsacaoMedia = pulsacaoMediaCamas[i];
38            camaPulsacaoMediaMaior = i;
39        }
40    }
41
42    cout << "Cama com maior pulsação média: " << camaPulsacaoMediaMaior + 1 << "\n";
43
44    return 0;
45 }
```

```
1 // EXERCÍCIO 28
2
3 #include <iostream>
4 #include <stdlib.h> /* srand, rand */
5 #include <time.h> /* time */
6 using namespace std;
7
8 void lerMatriz(int matriz[][5]) {
9     srand (time(NULL));
10    for(int i = 0; i < 2; i++) {
11        for(int j = 0; j < 5; j++) {
12            cout << matriz[i][j] << " ";
13        }
14        cout << "\n";
15    }
16 }
17
18 void gerarMatriz(int matriz[][5]) {
19    for(int i = 0; i < 2; i++) {
20        for(int j = 0; j < 5; j++) {
21            matriz[i][j] = rand() % 20 - 1;
22            cout << matriz[i][j] << " ";
23        }
24        cout << "\n";
25    }
26 }
27
28 void matrizMultiplicaL1L2() {
29     int matriz[2][5];
30     gerarMatriz(matriz);
31
32     for(int j = 0; j < 5; j++) {
33         int multiplica = 0;
34         multiplica = matriz[0][j] * matriz[1][j];
35         matriz[1][j] = multiplica;
36     }
37
38     cout << "\n";
39
40     lerMatriz(matriz);
41
42 }
43
44 void matrizSomaL1L2() {
45     int matriz[2][5];
46     gerarMatriz(matriz);
47
48     for(int j = 0; j < 5; j++) {
49         int soma = 0;
50         soma = matriz[0][j] + matriz[1][j];
51         matriz[1][j] = soma;
52     }
53
54     cout << "\n";
55     lerMatriz(matriz);
56 }
```

```
57
58 int main () {
59     setlocale(LC_ALL, "pt_BR.UTF-8");
60
61     cout << "Soma de Lin1 + lin2" << "\n";
62     matrizSomaL1L2();
63
64     cout << "\n";
65
66     cout << "Multiplicação de Lin1 + lin2" << "\n";
67     matrizMultiplicaL1L2();
68     return 0;
69 }
```

```
1 // EXERCÍCIO 29
2
3 #include <iostream>
4 using namespace std;
5
6 string procuraCaracter (string palavra, char caractereProcurado) {
7     int posicaoCaracter;
8     int contador = 0;
9     string resultado = "";
10    for (int i = 0; i < palavra.length(); i++) {
11        if (palavra[i] == caractereProcurado) {
12            resultado += i + '0';
13        }
14    }
15
16    return resultado;
17 }
18
19 int main () {
20     setlocale(LC_ALL, "pt_BR.UTF-8");
21
22     string palavra;
23     char caractereProcurado;
24
25     cout << "Informe a palavra a ser pesquisada: ";
26     cin >> palavra;
27
28     cout << "Informe o carácter a ser encontrado: ";
29     cin >> caractereProcurado;
30
31     string pesquisa = procuraCaracter(palavra, caractereProcurado);
32
33     if (pesquisa.length() == 0) {
34         cout << "Caracter procurado não existe!!" << "\n";
35     } else {
36         for(int i = 0; i < pesquisa.length(); i++) {
37             cout << "Carácter " << "|" << caractereProcurado << "|" << " na posição: " <<
38             (pesquisa[i] + 1) - '0' << "\n";
39         }
40     }
41
42     return 0;
43 }
```

```
1 // EXERCÍCIO 30
2
3 #include <iostream>
4 using namespace std;
5
6 bool valorPerfeito(int numero) {
7     int acc = 0;
8     bool perfeito = false;
9
10    for (int i = 1; i < numero; i++) {
11        if (numero % i == 0) {
12            acc += i;
13        }
14    }
15
16    if (acc == numero) {
17        perfeito = true;
18    }
19
20    return perfeito;
21 }
22
23 int main () {
24     setlocale(LC_ALL, "pt_BR.UTF-8");
25
26     int numero;
27
28     cout << "|===== NUMERO PERFEITO =====|" << "\n";
29     cout << "Qual número deseja verificar se é perfeito?" << "\n";
30     cin >> numero;
31
32     if (valorPerfeito(numero) == 1) {
33         cout << "O número " << numero << " É perfeito!" << "\n";
34     } else {
35         cout << "O número " << numero << " NÃO é perfeito!" << "\n";
36     }
37
38     return 0;
39 }
```