

Exercícios Propostos de Linguagem de Programação C++

1. Crie um programa que receba os valores antigo e atual de um produto. Chame uma sub-rotina que determine o percentual de acréscimo entre esses valores. O resultado deverá ser mostrado no programa principal.

```
#include <iostream>

double calcularporcentagem(double valorantigo, double valoratual) {
    if (valorantigo == 0) {

        std::cerr << "Erro: 0 valor antigo não pode ser zero." << std::endl;
        return 0.0;
    }

    double porcentagem = ((valoratual - valorantigo) / valorantigo) * 100.0;

    return porcentagem;
}

int main() {

    double valorantigo, valoratual;

    std::cout << "Informe o valor antigo do produto: ";
    std::cin >> valorantigo;

    std::cout << "Informe o valor atual do produto: ";
    std::cin >> valoratual;

    double porcentagem = calcularporcentagem(valorantigo, valoratual);

    std::cout << "O percentual de acréscimo é: " << porcentagem << "%" <<
    std::endl;

    return 0;
}
```

2. Faça uma sub-rotina que leia cinco valores inteiros, determine e mostre o maior e o menor deles.

```
#include <iostream>
```

```

void menoremaior(int valores[], int tamanho, int &maior, int &menor) {
    maior = menor = valores[0];

    for (int i = 1; i < tamanho; ++i) {
        if (valores[i] > maior) {
            maior = valores[i];
        }

        if (valores[i] < menor) {
            menor = valores[i];
        }
    }
}

int main() {
    const int tamanho = 5;
    int valores[tamanho];

    std::cout << "Informe cinco valores inteiros:" << std::endl;

    for (int i = 0; i < tamanho; i++) {
        std::cout << "Valor " << i + 1 << ": ";
        std::cin >> valores[i];
    }

    int maior, menor;

    menoremaior(valores, tamanho, maior, menor);

    std::cout << "O maior valor é: " << maior << std::endl;
    std::cout << "O menor valor é: " << menor << std::endl;
    return 0;
}

```

3. Faça uma sub-rotina que receba um vetor A de dez elementos inteiros como parâmetro. Ao final dessa função, deverá ter sido gerado um vetor B contendo o fatorial de cada elemento de A. O vetor B deverá ser mostrado no programa principal.

```

#include <iostream>

int calcularfatorial(int num) {
    if (num == 0 || num == 1) {
        return 1;
    } else {
        return num * calcularfatorial(num - 1);
    }
}

```

```
void Vetorfatorial(int vetorA[], int vetorB[], int tamanho)
{
    for (int i = 0; i < tamanho; i++) {
        vetorB[i] = calcularfatorial(vetorA[i]);
    }
}
```

```
int main() {
    const int tamanho = 10;
    int vetorA[tamanho];
    int vetorB[tamanho];
```

```
std::cout << "Informe dez numeros inteiros para o vetor A:" << std::endl;
```

```
for (int i = 0; i < tamanho; i++) {
    std::cout << "elemento " << i + 1 << ": ";
    std::cin >> vetorA[i];
}
```

```
Vetorfatorial(vetorA, vetorB, tamanho);
```

```
std::cout << "Vetor B (fatoriais): ";
```

```
for (int i = 0; i < tamanho; i++) {
    std::cout << vetorB[i] << " ";
}
std::cout << std::endl;
```

```
return 0;
}
```

```
#include <iostream>
#include <string>
```

```
std::string mudarpalavra(const std::string& frase, const std::string& palavra, const
std::string& novapalavra) {
    std::string frasemudada = frase;
    size_t posicao = frasemudada.find(palavra);

    while (posicao != std::string::npos) {
        frasemudada.replace(posicao, palavra.length(), novapalavra);
        posicao = frasemudada.find(palavra, posicao + novapalavra.length());
    }
    return frasemudada;
}
```

```
int main() {
    std::string frase, palavra, novapalavra;
```

```

std::cout << "Digite uma frase: ";
std::getline(std::cin, frase);

std::cout << "Digite a palavra a ser substituída: ";
std::cin >> palavra;

std::cout << "Digite a nova palavra: ";
std::cin >> novapalavra;

std::string frasemudada = mudarpalavra(frase, palavra, novapalavra);

std::cout << "Frase modificada: " << frasemudada << std::endl;

return 0;
}

```

4. Faça uma sub-rotina que receba um único valor representando segundos. Essa sub-rotina deverá convertê-lo para horas, minutos e segundos. Todas as variáveis devem ser passadas como parâmetro, não havendo variáveis globais.

```

#include <iostream>

void converterSegundos(int totalsegundos, int &horas, int &minutos, int &segundos) {
    horas = totalsegundos / 3600;
    minutos = (totalsegundos % 3600) / 60;
    segundos = totalsegundos % 60;
}

int main() {
    int totalsegundos, horas, minutos, segundos;

    std::cout << "Informe o total de segundos: ";
    std::cin >> totalsegundos;

    converterSegundos(totalsegundos, horas, minutos, segundos);

    std::cout << "Horas: " << horas << " minutos: " << minutos << " segundos: " << segundos << std::endl;
    return 0;
}

```

5. Crie uma sub-rotina que receba como parâmetro a hora de início e a hora de término de um jogo, ambas subdivididas em dois valores distintos: horas e minutos. A sub-rotina deverá retornar a duração expressa em minutos,

considerando que o tempo máximo de duração de um jogo é de 24 horas e que ele pode começar em um dia e terminar no outro.

```
#include <iostream>

int calculoeminutos(int horainicio, int minutoinicio, int horatermino,
int minutotermino) {
    const int minutosporhora = 60;
    const int horaspordia= 24;

    int minutosinicio = horainicio * minutosporhora + minutoinicio;
    int minutotermino = horatermino * minutosporhora + minutotermino;

    int duracaoEmMinutos = (minutotermino - minutosinicio + horaspordia*
minutosporhora) % (horaspordia* minutosporhora);

    return duracaoEmMinutos;
}

int main() {
    int horainicio, minutoinicio, horatermino, minutotermino;

    std::cout << "Informe a hora de início do jogo (horas:minutos): ";
    std::cin >> horainicio >> minutoinicio;

    std::cout << "Informe a hora de término do jogo (horas:minutos): ";
    std::cin >> horatermino >> minutotermino;

    int duracaoEmMinutos = calculoeminutos(horainicio, minutoinicio,
horatermino, minutotermino);

    std::cout << "A duração do jogo é de " << duracaoEmMinutos << " minutos."
<< std::endl;

    return 0;
}
```

6. Faça uma sub-rotina que receba como parâmetro uma matriz A(5,5) e retorne a soma de seus elementos.

```
#include <iostream>

int somamatriz(int matriz[5][5]) {
    int soma = 0;
```

```

for (int i = 0; i < 5; i++) {
for (int j = 0; j < 5; j++) {
soma += ma#include <iostream>
#include <string>

```

```

std::string mudarpalavra(const std::string& frase, const std::string&
palavra, const std::string& novapalavra) {
    std::string frasemudada = frase;
    size_t posicao = frasemudada.find(palavra);

```

```

    while (posicao != std::string::npos) {
        frasemudada.replace(posicao, palavra.length(), novapalavra);
        posicao = frasemudada.find(palavra, posicao +
novapalavra.length());
    }
    return frasemudada;
}

```

```

int main() {
    std::string frase, palavra, novapalavra;

```

```

    std::cout << "Digite uma frase: ";
    std::getline(std::cin, frase);

```

```

    std::cout << "Digite a palavra a ser substituída: ";
    std::cin >> palavra;

```

```

    std::cout << "Digite a nova palavra: ";
    std::cin >> novapalavra;

```

```

    std::string frasemudada = mudarpalavra(frase, palavra, novapalavra);

```

```

    std::cout << "Frase modificada: " << frasemudada << std::endl;

```

```

    return 0;
}triz[i][j];
}
}

```

```

return soma;
}

```

```

int main() {
int matriz[5][5];

```

```

std::cout << "Digite os elementos da matriz 5x5:" << std::endl;

```

```

for (int i = 0; i < 5; i++) {
for (int j = 0; j < 5; j++) {
std::cout << "Matriz[" << i << "]" << j << "]: ";
std::cin >> matriz[i][j];
}
}

int soma = somamatriz(matriz);

std::cout << "A soma dos elementos da matriz é: " << soma<< std::endl;
return 0;
}

```

7. Faça um programa que receba duas frases e gere uma terceira que represente a combinação das palavras das duas frases recebidas.

```

#include <iostream>
#include <sstream>
#include <vector>

std::string combinarfrase(const std::string& frase1, const std::string&
frase2) {
std::istringstream stream1(frase1);
std::istringstream stream2(frase2);

std::vector<std::string> palavras;
std::string palavra;

while (stream1 >> palavra) {
palavras.push_back(palavra);
}

while (stream2 >> palavra) {
palavras.push_back(palavra);
}

std::ostringstream resultado;
for (const auto& p : palavras) {
resultado << p << " ";
}
}

```

```

}

return resultado.str();
}

int main() {
    std::string frase1, frase2;

    std::cout << "Digite a primeira frase: ";
    std::getline(std::cin, frase1);

    std::cout << "Digite a segunda frase: ";
    std::getline(std::cin, frase2);

    std::string resultado = combinarfrase(frase1, frase2);

    std::cout << "A terceira frase combinada é: " << resultado << std::endl;

    return 0;
}

```

8. Faça um programa que receba uma frase, calcule e mostre a quantidade de vezes que a palavra AULA aparece na frase digitada.

```

#include <iostream>
#include <string>
#include <algorithm>

int contarpalavra(const std::string& frase) {
    const std::string palavramaiuscula = "AULA";
    const std::string palavraminuscula = "aula";

    std::string frasemini = frase;
    std::transform(frasemini.begin(), frasemini.end(), frasemini.begin(),
        ::tolower);

    int cont = 0;
    size_t posicaomaiuscula = frase.find(palavramaiuscula);

    while (posicaomaiuscula != std::string::npos) {
        ++cont;
        posicaomaiuscula = frase.find(palavramaiuscula, posicaomaiuscula + 1);
    }

    size_t posicaominuscula = frasemini.find(palavraminuscula);
}

```



```

while (posicaominuscula!= std::string::npos) {
++cont;
posicaominuscula= frasemini.find(palavraminuscula, posicaominuscula+ 1);
}

return cont;
}

int main() {
std::string frase;

std::cout << "Digite uma frase: ";
std::getline(std::cin, frase);

int quantidadedevezes = contarpalavra(frase);
std::cout << "As palavras 'AULA' aparecem " << quantidadedevezes << "
vezes na frase." << std::endl;

return 0;
}

```

9. Faça um programa para criptografar uma frase dada pelo usuário, ou seja, a criptografia deverá inverter a frase.

Exemplo:

- Frase: EU ESTOU NA ESCOLA
- Saída: ALOCSE AN UOTSE EU

```

#include <iostream>
#include <string>
#include <algorithm>

std::string fraseinversor(const std::string& frase) {
std::string fraseinvertida= frase;

std::re#include <iostream>
#include <string>

std::string mudarpalavra(const std::string& frase, const std::string&
palavra, const std::string& novapalavra) {
    std::string frasemudada = frase;
    size_t posicao = frasemudada.find(palavra);

    while (posicao != std::string::npos) {
        frasemudada.replace(posicao, palavra.length(), novapalavra);
        posicao = frasemudada.find(palavra, posicao +
novapalavra.length());
    }
}

```

```

    }
    return frasemudada;
}

int main() {
    std::string frase, palavra, novapalavra;

    std::cout << "Digite uma frase: ";
    std::getline(std::cin, frase);

    std::cout << "Digite a palavra a ser substituída: ";
    std::cin >> palavra;

    std::cout << "Digite a nova palavra: ";
    std::cin >> novapalavra;

    std::string frasemudada = mudarpalavra(frase, palavra, novapalavra);

    std::cout << "Frase modificada: " << frasemudada << std::endl;

    return 0;
}

return fraseinvertida;
}

int main() {
    std::string frase;

    std::cout << "Digite uma frase:";
    std::getline(std::cin, frase);

    std::string fraseinvertida= fraseinversor(frase);

    std::cout << "Frase invertida: " << fraseinvertida<< std::endl;

    return 0;
}

```

10. Faça um programa que receba uma frase e uma palavra. Caso a frase contenha a palavra ESCOLA, deverá substitui-la pela palavra digitada.

Exemplo:

- Frase: EU MORO PERTO DE UMA ESCOLA. MAS ESSA ESCOLA NÃO É A MELHOR.
- Palavra: PADARIA
- Resposta: EU MORO PERTO DE UMA PADARIA. MAS ESSA PADARIA NÃO É A MELHOR.

```
#include <iostream>
#include <string>
```

```
std::string mudarpalavra(const std::string& frase, const std::string&
palavra, const std::string& novapalavra) {
std::string frasemudada = frase;
size_t posicao = frasemudada.find(palavra);
```

```
while (posicao != std::string::npos) {
frasemudada.replace(posicao, palavra.length(), novapalavra);
posicao = frasemudada.find(palavra, posicao + novapalavra.length());
}
return frasemudada;
}
```

```
int main() {
std::string frase, palavra, novapalavra;
```

```
std::cout << "Digite uma frase: ";
std::getline(std::cin, frase);
```

```
std::cout << "Digite a palavra a ser substituída: ";
std::cin >> palavra;
```

```
std::cout << "Digite a nova palavra: ";
std::cin >> novapalavra;
```

```
std::string frasemudada = mudarpalavra(frase, palavra, novapalavra);
```

```
std::cout << "Frase modificada: " << frasemudada << std::endl;
```

```
return 0;
}
```