

```
#include <iostream> // necessario para o cout
#include <locale.h> // em algumas situacoes parece nao ser necessario para o setlocale
#include <string> // necessario para usar variaveis de string
#include <math.h> // necessario para usar pow, sqrt, round, ceil, floor, sin, cos, tan
#include <windows.h> // necessario para usar sleep e beep
#include <conio.h> // necessario para usar getch e getche
#include <fstream> // necessario para gravar arquivos com ofstream
#include <time.h> // necessario para srand (time(NULL));
#include <vector> // usado para vetores

using namespace std;

/*Luis Eduardo Ribeiro Guerra
Mini kit de Dev C++ v0.9 */
/* feito no Dev-C++ 5.11 para livre uso creativo e distribuiçao*/
/* Contato: luiseduardoribeiroguerra@gmail.com */
/*Github com códigos fontes e atualizações:
https://github.com/luisrguerra/devcpp-ptbr-mini-kit-cpp
*/

// variaveis

// char char_ex = 'a'; // um caractere
// string string_ex; // texto
// int int_ex; // numero inteiro
// bool bool_ex; // 0 ou 1; false(0) ou true(1)
// float float_ex; // ponto flutuante
// double double_ex; // ponto flutuante de dupla precisao

/// usado nos exemlos de funcoes
void funcaoteste (){
    cout << "Teste da função realizado" << endl;
}

int funcaoteste2 (int envio, int envio2){
    cout << "Teste da função 2 sendo realizado. " << "Enviado: " << envio << " e " << envio2 << endl;
    return envio + envio2;
}

void passagem_de_parametro_por_referencia (int &y, int x){
    y = x + y;
}

void limpeza_terminal(){ //criei essa funcao para deixar o código menor e menos poluido //ex: limpeza_terminal();
    cout << endl << "O terminal será limpo!" << endl;
    system("PAUSE"); // aperte qualquer tecla para continuar
    system("cls"); //apagar tela
}

void pular_linha(int numero_de_linhas){ // ex: pular_linha(1); //pula uma linha
    for(int contagem = 1; contagem <= numero_de_linhas; contagem++){
        cout << endl;
    }
}

/// utilizado no exemplo de uso de structs
struct pessoa{
    int matricula;
    string cor_cabelo;
};

/// utilizado no exemplo de orientacao por objeto
class estudante{
public:
    int prova;
    int teste;
    int trabalho;

    void notas_set(int prova_set, int teste_set, int trabalho_set){
        prova = prova_set;
        teste = teste_set;
        trabalho = trabalho_set;
    }

    int nota_final_get(){
        return (prova + teste + trabalho)/3;
    }
};

/*comece a ler a partir daqui
\
\
v */

int main(){
    setlocale(LC_ALL,"Portuguese"); // permite utilizar acentuacao do portugues

    //Pagina 1

    cout << "|Mini kit de Dev C++ v0.9|" << " Por Luis Eduardo Ribeiro Guerra" << endl;

    printf("\nMatemática:\n\n");
    cout << "Quatro ao quadrado é: " << pow(4,2) << endl;
    cout << "O resto da divisão de 3 por 2 é: " << 3%2 << endl;
    cout << "5 vezes 2 é: " << 5*2 << endl;
    cout << "10 dividido por 2 é: " << 10/2 << endl;
    cout << "2x2 + (5+7)/2 = " << 2*2 + (5+7)/2 << endl;
    cout << "Raiz quadrada de 16 é: " << sqrt(16) << endl;
    cout << "3.14159 arredondado é: " << round(3.14159) << endl;
    cout << "3.14159 arredondado para cima é: " << ceil(3.14159) << endl;
    cout << "3.14159 arredondado para baixo é: " << floor(3.14159) << endl;
```

```

cout << "Seno de 30 é: " << sin(30) << endl;
cout << "Cosseno de 30 é: " << cos(30) << endl;
cout << "Tangente de 30 é: " << tan(30) << endl;
cout << "O logarítimo de 1000 na base 10 é: " << log10(1000) << endl;
srand (time(NULL)); // torna os número aleatórios realmente aleatórios, necessita de #include <time.h>
cout << "Número aleatório de 0 a 99: " << rand() % 100 << endl;
cout << "Número aleatório de 1 a 100: " << rand() % 100 + 1 << endl;
cout << "Número aleatório de 25 a 50: " << rand() % 25 + 26 << endl;

```

```

limpeza_terminal(); /* Decidi criar e usar uma função para
limpar o terminal para simplificar o código, o código de limpeza esatva
repetido muitas vezes */

```

```

cout << "|Página 2|" << endl;

```

```

// uso do struct
cout << "Testando struct:" << endl;
pessoa luis;
luis.matricula = 2019;
luis.cor_cabelo = "castanho";
cout << "luis.matricula = " << luis.matricula << endl;
cout << "luis.cor_cabelo = " << luis.cor_cabelo << endl;

```

```

cout << endl << "Contando até 3:" << endl;
for (int contagem = 1; contagem <= 3; contagem++) {

    cout << contagem << "\n"; /* \n pula linha*/
    if (contagem >= 3){
        cout << "Fim da contagem." << endl;
        pular_linha(1);
    }
}

```

```

cout << "Contagem regressiva a partir de 3:" << endl;
for (int contagem = 3; contagem >= 1; contagem--) {

    cout << contagem << "\n"; /* \n pula linha*/
    if (contagem <= 1){
        cout << "Fim da contagem." << endl; // endl também pula linha
        pular_linha(1); // isso é uma funcao que u criei para deixar mais facil pular linhas
    }
}

```

```

cout << "Dormindo por 1000 milisegundos...." << endl;
Sleep(1000); /// necessita do #include <windows.h>
cout << "Acordei" << endl;
cout << "Representado uma aspas dupla dentro de uma string:" << " \" \" << endl;
pular_linha(1);

```

```

limpeza_terminal();

```

```

cout << "|Página 3|" << endl;

```

```

cout << "Passagem de parâmetro por referência:" << endl;
int a = 2;
cout << "Valor de 'a': " << a << endl;
int b = 3;
cout << "Valor de 'b': " << b << endl;
cout << "Executando Passagem de parâmetro por referência dentro da função" << endl;
cout << "y = x + y" << endl;
cout << "'y' recebe como referência 'a'" << endl;
cout << "'x' recebe o valor de 'b'" << endl;
cout << "a = (x = b) + a" << endl;
cout << "a = " << b << " + " << a << endl;

```

```

passagem_de_parametro_por_referencia(a,b);

```

```

cout << "Valor de 'a': " << a << endl;

```

```

limpeza_terminal();

```

```

cout << "|Página 4|" << endl;

```

```

///teste de funcao, as funcoes se encontram no topo antes do int main
funcaoteste();
cout << "Retornado: " << funcaoteste2(3,4) << endl;
pular_linha(1);

```

```

cout << "Exemplos do uso de teclas" << endl;
cout << "Pressione uma tecla:" << endl;
char tecla=getche(); // aviso: o programa pausa até ser pressionada uma tecla
//char tecla2=getch(); //nao mostra na tela qual tecla foi pressionada
cout << endl;

```

```

cout << "Pressione uma tecla:" << endl;
char tecla2 = getch(); //nao mostra na tela qual tecla foi pressionada
cout << "Você pressionou a tecla: " << tecla2 << endl;

```

```

limpeza_terminal();

cout << "|Página 5|" << endl;
cout << "Arquivos:" << endl;
pular_linha(1);
///gravar arquivo
ofstream meuarquivo("texto.txt");
if (meuarquivo.is_open()){
    meuarquivo << "01"<< endl;
    meuarquivo << "Hello" <<endl;
    meuarquivo << "Bonjour" << endl;
    meuarquivo << "Hasta la vista" << endl;
    meuarquivo.close();
    cout << "Salvei um arquivo de texto chamado texto.txt" << endl ;
}

///ler arquivo
ifstream meuarquivoleitura("texto.txt");
string textoleitura;
cout << "Olha o que está escrito no arquivo texto.txt:" << endl;
while (getline (meuarquivoleitura, textoleitura)) {
    cout << textoleitura << endl;
}
meuarquivoleitura.close();

pular_linha(1);
cout << "Criando uma pasta pasta_exemplo" << endl;
cout << "Se já houver uma pasta o cmd do Windows avisará" << endl;
/// o system irá executar um comando no terminal cmd para criar uma pasta
system("md \"pasta_exemplo\"");
cout << "Renomeando o arquivo texto.txt para texto_renomeado.txt" << endl;
cout << "Se já houver um arquivo com o mesmo nome o cmd do Windows avisará" << endl;
system("ren texto.txt texto_renomeado.txt");

///Manipulação de arquivo
int janela_sim_nao_notepad = MessageBox( 0, "Deseja abrir o arquivo texto_renomeado.txt no bloco de notas do Windows?", "Notepad", MB_YESNO );
switch (janela_sim_nao_notepad){
    case IDYES:
        /// execução de código
        /// é necessario utilizar \" para representar as aspas dentro da string sem causar problemas
        /// o programa fica pausado até voce finalizar o bloco de notas
        cout << "Feche o bloco de notas para despausar o programa" << endl;
        system("Notepad \"texto_renomeado.txt\"");
    break;
    case IDNO:
        /// execução de código
        cout << "Você não quis abrir arquivo texto_renomeado.txt" << endl;
        break;
}

int janela_sim_nao_txt = MessageBox( 0, "Deseja deletar o arquivo texto_renomeado.txt?", "Deletar", MB_YESNO );
switch (janela_sim_nao_txt){
    case IDYES:
        /// execução de código
        cout << "Você deletou o arquivo texto_renomeado.txt" << endl;
        system("del texto_renomeado.txt");
        break;
    case IDNO:
        /// execução de código
        cout << "Você não quis deletar arquivo texto_renomeado.txt" << endl;
        break;
}

limpeza_terminal();

cout << "|Página 6|" << endl;
pular_linha(1);
cout << "Arrays:" << endl;
pular_linha(1);
cout << "Exemplo de array de string de 1x4 como lista de compras:" << endl;
string array_string_lista_de_compras[4] = {"Coxinha", "Suco", "Bolo", "Fanta Uva"};
cout << "Primeiro item da lista, número 0: " << array_string_lista_de_compras[0] << endl;
cout << "Alterando o primeiro item da lista" << endl;
array_string_lista_de_compras[0] = "Pizza";
cout << "Primeiro item da lista, número 0: " << array_string_lista_de_compras[0] << endl;
cout << "Último item da lista, número 3: " << array_string_lista_de_compras[3] << endl;
cout << "Lendo todo o Array de String 1x4 da lista de compras:" << endl;
for (int contagem = 0; contagem < 4; contagem++) {
    cout << array_string_lista_de_compras[contagem] << " ";
}
pular_linha(2);

cout << "Exemplo de array de char (caracteres) de 3x4 como matriz:" << endl;
cout << "Definindo a matriz..." << endl;
char array_char_matriz [3] [4] =
{{' ',' ',' ',' '},
{'|','o','i','|'},
{' ',' ',' ',' '}};

cout << "Renderizando a matriz linha por linha:" << endl;
int tamanho_horizontal = 4;
int tamanho_vertical = 3;
for (int contagem_vertical = 0; contagem_vertical < tamanho_vertical; contagem_vertical++) {
    for (int contagem_horizontal = 0; contagem_horizontal < tamanho_horizontal; contagem_horizontal++){

```

```

    cout << array_char_matriz[contagem_vertical][contagem_horizantal];

}
    pular_linha(1); //pulando a linha depois de escrito a respectiva linha
}
pular_linha(1);

cout << "Uma string funciona parecido com um array:" << endl;
cout << "Defini uma string escrito (cavalo branco)" << endl;
string exemplo_string = "cavalo branco";
cout << "A primeira letra (caractere) da string, na posição 0, é: " << exemplo_string[0] << endl;
cout << "O tamanho da string é: " << exemplo_string.size() << " caracteres" << endl;
cout << "A última letra (caractere) da string, na posição 0, é: " << exemplo_string[ exemplo_string.size() - 1 ] << endl;
cout << "Soletrando:" << endl;
for (int contagem = 0; contagem < exemplo_string.size(); contagem++){
    cout << exemplo_string[contagem] << "_";
}
pular_linha(1);

limpeza_terminal();

cout << "|Página 7|" << endl;
cout << endl << "Vetor e iterador:" << endl;
cout << "Declarando vetor de inteiros (vetor_exemplo)" << endl;
vector<int> vetor_exemplo;
cout << "Testando se o vetor está vazio (1 = verdadeiro 0 = falso):" << vetor_exemplo.empty() << endl;
cout << "Adicionando o valor 2 ao vetor" << endl;
vetor_exemplo.push_back(2);
cout << "Adicionando o valor 4 ao vetor" << endl;
vetor_exemplo.push_back(4);
cout << "Adicionando o valor 8 ao vetor" << endl;
vetor_exemplo.push_back(8);
cout << "Primeiro valor do vetor: " << vetor_exemplo[0] << endl;;
cout << "Tamanho do vetor:" << vetor_exemplo.size() << endl;
cout << "Último valor do vetor: " << vetor_exemplo[ vetor_exemplo.size() - 1 ] << endl;
cout << "Removendo o último valor do vetor" << endl;
vetor_exemplo.pop_back();
cout << "Tamanho do vetor:" << vetor_exemplo.size() << endl;
cout << "Último valor do vetor: " << vetor_exemplo[ vetor_exemplo.size() - 1 ] << endl;
cout << "Testando se o vetor está vazio (1 = verdadeiro 0 = falso):" << vetor_exemplo.empty() << endl;
cout << "Declarando um iterador (iterador_exemplo)" << endl;
vector<int>::iterator iterador_exemplo;
cout << "Colocando o iterador no início do vetor" << endl;
iterador_exemplo = vetor_exemplo.begin();
cout << "Valor do vetor onde o iterador se encontra: "<< *iterador_exemplo << endl;
cout << "Movendo o iterador uma posição para frente" << endl;
iterador_exemplo++;
cout << "Valor do vetor onde o iterador se encontra: "<< *iterador_exemplo << endl;
cout << "Movendo o iterador uma posição para trás" << endl;
iterador_exemplo--;
cout << "Valor do vetor onde o iterador se encontra: "<< *iterador_exemplo << endl;
cout << "Colocando o iterador no final do vetor" << endl;
iterador_exemplo = vetor_exemplo.end();
cout << "Valor do vetor onde o iterador se encontra: "<< *(iterador_exemplo-1) << endl;

limpeza_terminal();

cout << "|Página 8|" << endl;

/// a classe estudante voce encontra no topo do codigo acima do int main
cout << "Orientação a objetos:" << endl;
cout << "Declarando o aluno Miguel da classe estudante" << endl;
estudante Miguel;
cout << "Declarando a aluna Beatriz da classe estudante" << endl;
estudante Beatriz;

cout << "Definindo as notas do Miguel" << endl;
Miguel.prova = 80;
Miguel.teste = 70;
Miguel.trabalho = 60;
cout << "Definindo as notas da Beatriz com o método set" << endl;
Beatriz.notas_set(/*prova_set:*/ 90 , /*teste_set:*/ 80 , /*trabalho_set:*/ 70);

cout << "Nota final Miguel: " << Miguel .nota_final_get() << endl;
cout << "Nota final Beatriz: " << Beatriz.nota_final_get() << endl;

limpeza_terminal();

cout << "|Página 9|" << endl;

cout << "Executando caixas de aviso..." << endl;
/////https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-messagebox
MessageBox( 0, "Atenção!\nComi o bolo de quem está lendo", "Atenção titulo", 0 | MB_ICONEXCLAMATION ); //icone de "!" em um triangulo amarelo
MessageBox( 0, "Linha 1\nLinha 2", "janela_erro titulo", 0 | MB_ICONERROR ); //icone de "x" em uma bola vermelha
MessageBox( 0, "Linha 1\nLinha 2", "janela_interrogação titulo", 0 | MB_ICONQUESTION ); //icone de "?" em uma bola azul
MessageBox( 0, "Linha 1\nLinha 2", "janela_informação titulo", 0 | MB_ICONINFORMATION ); //icone de "i" em uma bola azul
MessageBox( 0, "Linha 1\nLinha 2", "janela_sem_icone titulo", 0); // sem icone

/// janela que responde aos comandos

```

```

int janela_sim_nao_cancelar = MessageBox( 0, "Linha 1\nLinha 2", "janela_sim_nao_cancelar titulo", MB_YESNOCANCEL );
switch (janela_sim_nao_cancelar){
case IDYES:
    //codigo
    cout << "janela_sim_nao_cancelar: Você selecionou sim" << endl;
    break;
case IDNO:
    //codigo
    cout << "janela_sim_nao_cancelar: Você selecionou não" << endl;
    break;
case IDCANCEL:
    //codigo
    cout << "janela_sim_nao_cancelar: Você selecionou cancelar" << endl;
    break;
}

int janela_anular_repetir_ignorar = MessageBox( 0, "Linha 1\nLinha 2", "janela_anular_repetir_ignorar titulo", MB_ABORTRETRYIGNORE );
switch (janela_anular_repetir_ignorar){
case IDABORT:
    //codigo
    cout << "janela_anular_repetir_ignorar: Você selecionou anular" << endl;
    break;
case IDRETRY:
    //codigo
    cout << "janela_anular_repetir_ignorar: Você selecionou repetir" << endl;
    break;
case IDIGNORE:
    //codigo
    cout << "janela_anular_repetir_ignorar: Você selecionou ignorar" << endl;
    break;
}

int janela_ok_cancelar = MessageBox( 0, "Linha 1\nLinha 2", "janela_ok_cancelar titulo", MB_OKCANCEL );
switch (janela_ok_cancelar){
case IDOK:
    //codigo
    cout << "janela_ok_cancelar: Você selecionou ok" << endl;
    break;
case IDCANCEL:
    //codigo
    cout << "janela_ok_cancelar: Você selecionou cancelar" << endl;
    break;
}

int janela_repetir_cancelar = MessageBox( 0, "Linha 1\nLinha 2", "janela_repetir_cancelar titulo", MB_RETRYCANCEL );
switch (janela_repetir_cancelar){
case IDRETRY:
    cout << "janela_repetir_cancelar: Você selecionou repetir" << endl;
    break;
case IDCANCEL:
    cout << "janela_repetir_cancelar: Você selecionou cancelar" << endl;
    break;
}

int janela_cancelar_tentar_continuar = MessageBox( 0, "Linha 1\nLinha 2", "janela_cancelar_tentar_continuar titulo", MB_CANCELTRYCONTINUE );
switch (janela_cancelar_tentar_continuar){
case IDCANCEL:
    //codigo
    cout << "janela_cancelar_tentar_continuar: Você selecionou cancelar" << endl;
    break;
case IDTRYAGAIN:
    //codigo
    cout << "janela_cancelar_tentar_continuar: Você selecionou tentar" << endl;
    break;
case IDCONTINUE:
    //codigo
    cout << "janela_cancelar_tentar_continuar: Você selecionou continuar" << endl;
    break;
}

limpeza_terminal();

/// animacao
bool rodando_animacao = true;
int intervalo_animacao = 100; // em milisegundos
int animacao_ciclos = 0;
while ( rodando_animacao == true /*true ou false ou alguma condicao*/ ) {

    cout << "|" << "\n";
    Sleep(intervalo_animacao);
    system("cls"); //apagar tela
    cout << "/" << "\n";
    Sleep(intervalo_animacao);
    system("cls"); //apagar tela
    cout << "-" << "\n";
    Sleep(intervalo_animacao);
    system("cls"); //apagar tela
    cout << "\"\" << "\n";
    Sleep(intervalo_animacao);
    system("cls"); //apagar tela
    animacao_ciclos++;
    if (animacao_ciclos >= 4){
        rodando_animacao = false;
        //break; //interrompe o while
    }
}
// fim do codigo da animacao

cout << "|Página 10|" << endl;

```

```

cout << "Som de aviso do windows" << endl;
cout << '\a'; //som de aviso

Sleep(2000); /// necessita do #include <windows.h>

cout << "Beep de frequencia" << endl;
//Beep(frequência,duração);

printf("Dó\n"); ; // printf (""); funciona parecido com o cout << "";
Beep (261, 500);

printf("Ré\n");
Beep (293, 500);

printf("Mi\n");
Beep (329, 500);

printf("Fá\n");
Beep (349, 500);

printf("Sol\n");
Beep (392, 500);

printf("Lá\n");
Beep (440, 500);

printf("Si\n");
Beep (493, 500);

int janela_sim_nao = MessageBox( 0, "Deseja abrir o link no seu navegador padrão?", "Link", MB_YESNO );
switch (janela_sim_nao){
    case IDYES:
        /// o system executa um comando de terminal cmd, isso é algo bem util
        system("start https://github.com/luisrguerra/devcpp-ptbr-mini-kit-cpp");
        break;
    case IDNO:
        cout << "Você não quis abrir o link no seu navegador padrão" << endl;
        break;
}

int janelafinal = MessageBox( 0, "Deseja reiniciar o programa?", "Reinicialização", MB_RETRYCANCEL );
switch (janelafinal){
    case IDRETRY:
        cout << "Reiniciando..." << endl;
        Sleep(3000); /// necessita do #include <windows.h>
        system("cls"); //apagar tela
        main();
        /*chamar a funcao main dentro do main faz o
        programa reiniciar, dependendo da situacao, deixando
        ele em looping */
        break;
    case IDCANCEL:
        cout << "Finalizando o programa..." << endl;
        return 0; //finalizando o programa
        break;
}

return 0;
}

```