

```

#include <locale.h> // para o idioma em português
#include <iostream> // textos no terminal
// #include <conio.h>
#include <windows.h> // para mudar as cores
using namespace std;

/* Luis Eduardo Ribeiro Guerra
   Mudar Cor UX kit de Dev C++ v0.1 */
/* feito no Dev-C++ 5.11 para livre uso creativo e distribuição */
/* Contato: luiseduardoribeiroguerra@gmail.com */
/* Github com códigos fontes e atualizações:
   https://github.com/luisrguerra/devcpp-ptbr-mini-kit-cpp
   */

enum cmd_cores {
    /*0*/cor_preto,          /*1*/cor_azul,              /*2*/cor_verde,            /*3*/cor_ciano,
    /*4*/cor_vermelho,       /*5*/cor_magenta,          /*6*/cor_marrom,          /*7*/cor_cinza_claro,
    /*8*/cor_cinza,          /*9*/cor_azul_claro,       /*10*/cor_verde_claro,    /*11*/cor_ciano_claro,
    /*12*/cor_vermelho_claro, /*13*/cor_magenta_claro,   /*14*/cor_amarelo,        /*15*/cor_branco,
    corteste};

void cor_texto (cmd_cores iColor){
    HANDLE h1 = GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_SCREEN_BUFFER_INFO bufferInfo;
    BOOL b = GetConsoleScreenBufferInfo(h1, &bufferInfo);
    bufferInfo.wAttributes &= 0x00F0;
    SetConsoleTextAttribute (h1, bufferInfo.wAttributes | iColor);
}

void cor_fundo (cmd_cores iColor){
    HANDLE h1 = GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_SCREEN_BUFFER_INFO bufferInfo;
    BOOL b = GetConsoleScreenBufferInfo(h1, &bufferInfo);
    bufferInfo.wAttributes &= 0x000F;
    SetConsoleTextAttribute (h1, bufferInfo.wAttributes | (iColor << 4));
}

void trocar_cor_padrao_cmd_rgb(int cor,int red,int green,int blue){
    CONSOLE_SCREEN_BUFFER_INFOEX info;
    info.cbSize = sizeof(CONSOLE_SCREEN_BUFFER_INFOEX);

    HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
    GetConsoleScreenBufferInfoEx(hConsole, &info);

    info.ColorTable[cor] = RGB (red,green,blue);

    SetConsoleScreenBufferInfoEx(hConsole, &info);
}

void desabilitar_cursor(bool habilitado){
    HANDLE out = GetStdHandle(STD_OUTPUT_HANDLE);

    CONSOLE_CURSOR_INFO    cursorInfo;

    GetConsoleCursorInfo(out, &cursorInfo);
    cursorInfo.bVisible = habilitado -1;
    SetConsoleCursorInfo(out, &cursorInfo);
}

void imprimir_texto_colorido(cmd_cores cor,std::string texto){
    cor_texto (cor);
    cor_fundo (cor_preto);
    std::cout << texto;
    cor_texto (cor_cinza_claro);
}

void imprimir_texto_colorido(cmd_cores cor,cmd_cores fundo,std::string texto){
    cor_texto (cor);
    cor_fundo (fundo);
    std::cout << texto;
    cor_texto (cor_cinza_claro);
    cor_fundo (cor_preto);
}

int main () {

```

```

setlocale(LC_ALL, "Portuguese");

cout << "Kit UX para Mudar Cor do CMD no Dev C++ V0.1" << endl;
cout << "Por Luís Eduardo Ribeiro Guerra" << endl;

cor_texto(cor_azul);
cout << "Texto Azul" << endl;

cor_texto(cor_azul_claro);
cout << "Texto Azul Claro" << endl;

cor_texto(cor_ciano);
cout << "Texto Ciano" << endl;

cor_texto(cor_ciano_claro);
cout << "Texto Ciano Claro" << endl;

cor_texto(cor_vermelho);
cout << "Texto Vermelho" << endl;

cor_texto(cor_vermelho_claro);
cout << "Texto Vermelho Claro" << endl;

cor_texto(cor_magenta);
cout << "Texto Magenta" << endl;

cor_texto(cor_magenta_claro);
cout << "Texto Magenta Claro" << endl;

cor_texto(cor_verde);
cout << "Texto Verde" << endl;

cor_texto(cor_verde_claro);
cout << "Texto Verde Claro" << endl;

cor_texto(cor_amarelo);
cout << "Texto Amarelo" << endl;

cor_texto(cor_marrom);
cout << "Texto Marrom" << endl;

cor_texto(cor_cinza);
cout << "Texto Cinza Escuro" << endl;

cor_texto(cor_cinza_claro);
cout << "Texto Cinza Claro" << endl;

cor_texto(cor_branco);
cout << "Texto Branco" << endl;

cor_texto(cor_preto);
cor_fundo(cor_branco);
cout << "Texto Preto, Fundo Branco " << endl;

cor_texto(cor_preto);
cor_fundo(cor_azul);
cout << "Texto Preto, Fundo Azul " << endl;

cor_texto(cor_preto);
cor_fundo(cor_azul_claro);
cout << "Texto Preto, Fundo Azul Claro " << endl;

cor_texto(cor_preto);
cor_fundo(cor_ciano);
cout << "Texto Preto, Fundo Ciano " << endl;

cor_texto(cor_preto);
cor_fundo(cor_ciano_claro);
cout << "Texto Preto, Fundo Ciano Claro " << endl;

cor_texto(cor_preto);
cor_fundo(cor_vermelho);
cout << "Texto Preto, Fundo Vermelho " << endl;

```

```

cor_texto(cor_preto);
cor_fundo(cor_vermelho_claro);
cout << "Texto Preto, Fundo Vermelho Claro " << endl;

cor_texto(cor_preto);
cor_fundo(cor_magenta);
cout << "Texto Preto, Fundo Magenta " << endl;

cor_texto(cor_preto);
cor_fundo(cor_magenta_claro);
cout << "Texto Preto, Fundo Magenta Claro " << endl;

cor_texto(cor_preto);
cor_fundo(cor_verde);
cout << "Texto Preto, Fundo Verde " << endl;

cor_texto(cor_preto);
cor_fundo(cor_verde_claro);
cout << "Texto Preto, Fundo Verde Claro " << endl;

cor_texto(cor_preto);
cor_fundo(cor_amarelo);
cout << "Texto Preto, Fundo Amarelo " << endl;

cor_texto(cor_preto);
cor_fundo(cor_marrom);
cout << "Texto Preto, Fundo Marrom " << endl;

cor_texto(cor_preto);
cor_fundo(cor_cinza);
cout << "Texto Preto, Fundo Cinza Escuro " << endl;

cor_texto(cor_preto);
cor_fundo(cor_cinza_claro);
cout << "Texto Preto, Fundo Cinza Claro " << endl;

cor_texto(cor_branco);
cor_fundo(cor_preto);
cout << "Texto Branco, Fundo Preto " << endl;

cout << "Arco-íris:" << endl;
cor_fundo(cor_vermelho);
cout << " ";
cor_fundo(cor_vermelho_claro);
cout << " ";
cor_fundo(cor_marrom);
cout << " ";
cor_fundo(cor_verde);
cout << " ";
cor_fundo(cor_azul_claro);
cout << " ";
cor_fundo(cor_azul);
cout << " ";
cor_fundo(cor_magenta);
cout << " " << endl;

trocar_cor_padrao_cmd_rgb(/*cor:*/cor_azul,/*red:*/26,/*green:*/35,/*blue:*/126);
cor_fundo(/*cor:*/cor_azul);
cout << "Azul RGB(26,35,126)" << endl;

trocar_cor_padrao_cmd_rgb(cor_vermelho_claro,255,82,82);
cor_fundo(cor_vermelho_claro);
cout << "Vermelho claro RGB(255,82,82)" << endl;

trocar_cor_padrao_cmd_rgb(cor_verde,51,105,30);
cor_fundo(cor_verde);
cout << "Verde RGB(51,105,30)" << endl;

imprimir_texto_colorido(/*cor texto:*/cor_vermelho_claro,/*texto:*/"Texto\n");

imprimir_texto_colorido(/*cor texto:*/cor_verde_claro,/*cor fundo:*/cor_azul ,/*texto:*/"Texto\n");

cout << "Desativando o cursor que pisca" << endl;

```

```
desabilita_cursor(true);

system("PAUSE");
//link para o GitHub
system("start https://github.com/luisrguerra/devcpp-ptbr-mini-kit-cpp");

return 0;
}
```