

```

#include <iostream>
#include <locale.h>
#include <windows.h> //usar sleep
//#include <string>
#include <list> // necessário para usar listas
using namespace std;

/*Luis Eduardo Ribeiro Guerra
Mini kit (Exemplo biblioteca list) 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
*/

/* este exemplo é um exemplo que faz parte de um kit de c++, procure também fazer download
do kit principal no GitHub para ter acesso a um conteúdo mais completo */

void ler_lista(list<int> lista, string nome){
    cout<< nome + ": ";
    for (list<int>::iterator i = lista.begin(); i != lista.end(); i++){
        cout << *i << " ";
    }
    cout<< endl;
}

int main(){
    setlocale(LC_ALL, "Portuguese");
    cout << "Abrindo GitHub..." << endl;
    Sleep(1400);
    //Abrir no browser o link do GitHub do Mini Kit de c++
    system("start https://github.com/luisrguerra/devcpp-ptbr-mini-kit-cpp");

    cout << "|Mini kit de Dev C++ (Exemplos da biblioteca list) v0.1|" << " Por Luís Eduardo Ribeiro Guerra" << endl;

    cout << endl << "Lista exemplo 1" << endl;

    list<int> lista_exemplo;
    cout << "Adicionando 1 ao fim da lista" << endl;
    lista_exemplo.push_back(1);
    cout << "Adicionando 2 ao fim da lista" << endl;
    lista_exemplo.push_back(2);
    cout << "Adicionando 3 ao fim da lista" << endl;
    lista_exemplo.push_back(3);
    cout << "Adicionando 6 ao fim da lista" << endl;
    lista_exemplo.push_back(6);
    cout << "Adicionando 5 ao fim da lista" << endl;
    lista_exemplo.push_back(5);
    cout << "Adicionando 4 ao fim da lista" << endl;
    lista_exemplo.push_back(4);
    ///
    ler_lista(lista_exemplo, "Lista exemplo");

    cout << "Ordenando a Lista" << endl;
    lista_exemplo.sort();
    ler_lista(lista_exemplo, "Lista exemplo");

    cout << "Removendo o último item da Lista" << endl;
    lista_exemplo.pop_back();
    ler_lista(lista_exemplo, "Lista exemplo");

    cout << "Removendo o primeiro item da Lista" << endl;
    lista_exemplo.pop_front();
    ler_lista(lista_exemplo, "Lista exemplo");

    cout << endl << "Lista exemplo 2" << endl;

    list<int> lista_exemplo2;
    cout << "Adicionando 5 \"0\" na lista 2" << endl;
    lista_exemplo2.assign(5, 0);
    ler_lista(lista_exemplo2, "Lista exemplo 2");

    cout << endl << "Lista exemplo 3" << endl;

    list<int> lista_exemplo3;
    cout << "Adicionando 1 ao inicio da lista 3" << endl;
    lista_exemplo3.push_front(1);
    cout << "Adicionando 2 ao inicio da lista 3" << endl;
    lista_exemplo3.push_front(2);
    cout << "Adicionando 3 ao inicio da lista 3" << endl;
    lista_exemplo3.push_front(3);
    cout << "Adicionando 4 ao inicio da lista 3" << endl;
    lista_exemplo3.push_front(4);
    cout << "Adicionando 5 ao inicio da lista 3" << endl;
    lista_exemplo3.push_front(5);
    ler_lista(lista_exemplo3, "Lista exemplo 3");

```

```

cout << "Invertendo a Lista exemplo 3" << endl;
lista_exemplo3.reverse();
ler_lista(lista_exemplo3, "Lista exemplo 3");

cout << "Redimensionando a Lista exemplo 3 para \"2 casas\"" << endl;
lista_exemplo3.resize(2);
ler_lista(lista_exemplo3, "Lista exemplo 3");

cout << "Apagando a Lista exemplo 3" << endl;
lista_exemplo3.clear();

cout << endl << "Lista exemplo 4" << endl;

list<int> lista_exemplo4;
cout << "Adicionando 2 ao inicio da lista 4" << endl;
lista_exemplo4.push_front(2);
cout << "Adicionando 4 ao inicio da lista 4" << endl;
lista_exemplo4.push_front(4);
cout << "Adicionando 6 ao inicio da lista 4" << endl;
lista_exemplo4.push_front(6);
cout << "Adicionando 8 ao inicio da lista 4" << endl;
lista_exemplo4.push_front(8);
cout << "Adicionando 10 ao inicio da lista 4" << endl;
lista_exemplo4.push_front(10);
ler_lista(lista_exemplo4, "Lista exemplo 4");

cout << "Declarando ponteiro_exemplo4" << endl;
list<int>::iterator ponteiro_exemplo4;

cout << endl << "Colocando o ponteiro_exemplo4 no inicio da lista 4" << endl;
ponteiro_exemplo4 = lista_exemplo4.begin();
cout << "Valor no ponteiro ponteiro_exemplo4: " << *ponteiro_exemplo4 << endl;

cout << endl << "Movendo ponteiro_exemplo4 +1 casa" << endl;
ponteiro_exemplo4++;
cout << "Valor no ponteiro ponteiro_exemplo4: " << *ponteiro_exemplo4 << endl;

cout << endl << "Movendo ponteiro_exemplo4 -1 casa" << endl;
ponteiro_exemplo4--;
cout << "Valor no ponteiro ponteiro_exemplo4: " << *ponteiro_exemplo4 << endl;

cout << endl << "Colocando o ponteiro_exemplo4 no final da lista 4" << endl;
ponteiro_exemplo4 = lista_exemplo4.end();
ponteiro_exemplo4--; //necessário voltar uma casa
cout << "Valor no ponteiro ponteiro_exemplo4: " << *ponteiro_exemplo4 << endl;

cout << endl << "Movendo ponteiro_exemplo4 -1 casa" << endl;
ponteiro_exemplo4--;
cout << "Valor no ponteiro ponteiro_exemplo4: " << *ponteiro_exemplo4 << endl;

cout << endl << "Inserindo o valor na posição do ponteiro_exemplo4" << endl;
lista_exemplo4.insert(ponteiro_exemplo4, 5);
ler_lista(lista_exemplo4, "Lista exemplo 4");
cout << "Valor no ponteiro ponteiro_exemplo4: " << *ponteiro_exemplo4 << endl;

cout << endl << "Removendo o valor na posição do ponteiro_exemplo4" << endl;
lista_exemplo4.erase(ponteiro_exemplo4);
ler_lista(lista_exemplo4, "Lista exemplo 4");
cout << "Valor no ponteiro ponteiro_exemplo4: " << *ponteiro_exemplo4 << endl;

cout << endl << "Lista exemplo 5" << endl;

list<int> lista_exemplo5;

cout << "Adicionando 2 ao inicio da lista 5" << endl;
lista_exemplo5.push_front(2);
cout << "Adicionando 0 ao inicio da lista 5" << endl;
lista_exemplo5.push_front(0);
cout << "Adicionando 6 ao inicio da lista 5" << endl;
lista_exemplo5.push_front(6);
cout << "Adicionando 0 ao inicio da lista 5" << endl;
lista_exemplo5.push_front(0);
cout << "Adicionando 10 ao inicio da lista 5" << endl;
lista_exemplo5.push_front(10);
ler_lista(lista_exemplo5, "Lista exemplo 5");

cout << "Removendo os valores \"0\" da lista 5" << endl;
lista_exemplo5.remove(0);
ler_lista(lista_exemplo5, "Lista exemplo 5");

system("PAUSE");
return 0;

```

