ECOP13A-Lab 11

STL - I

Guia de Laboratório

Prof. André Bernardi
andrebernardi@unifei.edu.br

11º Laboratório ECOP13A
27 de junho 2025

1ª Questão

1. (ex01.cpp) Utilizando a STL, escreva um programa em C++ para a demonstração do funcionamento de uma ***Pilha***, incluída através do cabeçalho `<stack>`. Faça um programa que deve mostrar repetidamente um menu com as opções que podem ser escolhidas pelo usuário. Ele deve funcionar de maneira semelhante ao exemplo a seguir:

```
---------------------------
Programa de Pilha STL
---------------------------

1.Insira um elemento na pilha
2.Remova um elemento da pilha
3.Tamanho da pilha
4.Primeiro elemento da pilha
5.Sair
Escolha (1-5):  1 (cin)
Entre com o valor a ser inserido:  56 (cin)

1.Insira um elemento na pilha
2.Remova um elemento da pilha
3.Tamanho da pilha
4.Primeiro elemento da pilha
5.Sair
Escolha (1-5):  1 (cin)
Entre com o valor a ser inserido:  26 (cin)

1.Insira um elemento na pilha
2.Remova um elemento da pilha
3.Tamanho da pilha
4.Primeiro elemento da pilha
5.Sair
Escolha (1-5):  1 (cin)
Entre com o valor a ser inserido:  12 (cin)
```

```
1.Insira um elemento na pilha
2.Remova um elemento da pilha
3.Tamanho da pilha
4.Primeiro elemento da pilha
5.Sair
Escolha (1-5):  2 (cin)
Elemento 12 removido

1.Insira um elemento na pilha
2.Remova um elemento da pilha
3.Tamanho da pilha
4.Primeiro elemento da pilha
5.Sair
Escolha (1-5):  3 (cin)
Tamanho da pilha: 2

1.Insira um elemento na pilha
2.Remova um elemento da pilha
3.Tamanho da pilha
4.Primeiro elemento da pilha
5.Sair
Escolha (1-5):  4 (cin)
Primeiro elemento da pilha: 26

1.Insira um elemento na pilha
2.Remova um elemento da pilha
3.Tamanho da pilha
4.Primeiro elemento da pilha
5.Sair
Escolha (1-5):  5(cin)

Programa finalizado!
```

1ª questão
Exemplo de solução

```
#include <stack>
#include <iostream>
using namespace std;

stack<int> pilha;

int menu () {
   cout << "\n1.Insira um elemento na pilha\n";
   cout << "2.Remova um elemento da pilha\n";
   cout << "3.Tamanho da pilha\n";
   cout << "4.Primeiro elemento da pilha\n";
   cout << "5.Sair\n";

   int aux = -1;

   while (!(1 <= aux && aux <= 5)) {
      cout << "Escolha (1-5): ";
      cin >> aux;
   }
   return aux;
}
```

main

```
int main () {
   cout << "-----------------------------\n";
   cout << "   Programa de Pilha STL   \n";
   cout << "-----------------------------\n";

   int op = -1;
   while (op != 5) {
      op = menu();

      switch (op) {
         case 1:
            cout << "Entre com o valor a ser inserido: ";
            int val;
            cin >> val;
            pilha.push(val);
            break;

         case 2:
            if (!pilha.empty()) {
               cout << "Elemento " << pilha.top() << " removido\n";
               pilha.pop();
            }
            else
               cout << "Pilha vazia\n";
            break;
```

```
            case 3:
                cout << "Tamanho da pilha: " << pilha.size() << "\n";
                break;

            case 4:
                if (!pilha.empty())
                    cout << "Primeiro elemento da pilha: " << pilha.top() << "\n";
                else
                    cout << "Pilha vazia\n";
        }

        cout << "\n";
    }

    cout << "Programa finalizado!\n";

    return 0;
}
```

2ª Questão

2. (ex02.cpp) Utilizando a STL, escreva um programa em C++ para a demonstração do funcionamento de uma ***Lista*** Encadeada, incluída através do cabeçalho `<list>`. Faça um programa que deve mostrar repetidamente um menu com as opções que podem ser escolhidas pelo usuário. Ele deve funcionar de maneira semelhante ao exemplo a seguir:

2ª Questão

```
==============================
 Implementação de Lista no STL
==============================
1.Inserir elemento na frente
2.Inserir elemento no final
3.Excluir elemento na frente
4.Excluir elemento no final
5.Exibir primeiro elemento da lista
6.Exibir último elemento da lista
7.Tamanho da lista
8.Redimensionar lista
9.Remover elementos com valores específicos
10.Remover valores duplicados
11.Reverter a ordem dos elementos
12.Ordenar a lista
13.Sair

Escolha uma opção: (Usuário entra com cin)
```

```cpp
//header file para classe list
#include <list>
#include <iostream>

using namespace std;

list<int> lista;

bool find (int v) {
   for (auto x : lista) {
      if (x == v)
         return true;
   }
   return false;
}
```

2ª questão
Exemplo de solução

```cpp
int menu () {
   cout << "1. Inserir elemento na frente\n";
   cout << "2. Inserir elemento no final\n";
   cout << "3. Excluir elemento na frente\n";
   cout << "4. Excluir elemento no final\n";
   cout << "5. Exibir primeiro elemento da lista\n";
   cout << "6. Exibir ultimo elemnto da lista\n";
   cout << "7. Tamanho da lista\n";
   cout << "8. Redimensionar lista\n";
   cout << "9. Remover elementos com valores especificos\n";
   cout << "10.Remover valores duplicados\n";
   cout << "11.Reverter a oredem dos elementos\n";
   cout << "12.Ordenar a lista\n";
   cout << "13.Sair\n";

   int aux = -1;
   while (!(1 <= aux && aux <= 13)) {
      cout << "Escolha uma opcao: ";
      cin >> aux;
   }
   return aux;
}
```

```
int main () {

   cout << "================================\n";
   cout << " Implementacao de Lista no STL \n";
   cout << "================================\n";

   int op = -1, val = -1;

   while (op != 13) {
      op = menu();

      switch (op) {
         case 1:
            cout << "Elemento a ser inserido: ";
            cin >> val;
            lista.push_front(val);
            break;
         case 2:
            cout << "Elemento a ser inserido: ";
            cin >> val;
            lista.push_back(val);
            break;
```

```
        case 3:
           if (!lista.empty()) {
              cout << "Elemento " << lista.front() << " excluido\n";
              lista.pop_front();
           }
           else
              cout << "Lista vazia\n";
           break;
        case 4:
           if (!lista.empty()) {
           cout << "Elemento " << lista.back() << " excluido\n";
           lista.pop_back();
           }
           else
              cout << "Lista vazia\n";
           break;
        case 5:
           if (!lista.empty()) {
              cout << "Primeiro elemento: " << lista.front() << "\n";
           }
           else
              cout << "Lista vazia\n";
           break;
```

```
        case 6:
           if (!lista.empty()) {
              cout << "Ultimo elemento: " << lista.back() << "\n";
           }
           else
              cout << "Lista vazia\n";
           break;

        case 7:
           cout << "Tamanho da lista: " << lista.size() << "\n";
           break;

        case 8:
           cout << "Novo tamanho: ";
           cin >> val;
           if (val >= 0) {
              lista.resize(val);
           }
           else
              cout << "Tamanho invalido\n";
           break;
```

```
        case 9:
            cout << "Elemento a ser removido: ";
            cin >> val;
            if (find(val)) {
                lista.remove(val);
            } else {
                cout << "Elemento nao esta na lista\n";
            }
            break;

        case 10:
            cout << "Lista Antiga:";
            for (auto x : lista) {
                cout << " " << x;
            }
            cout << "\n";

            lista.sort();

            lista.unique();
            break;
```

```
        case 11:
            cout << "Lista Antiga:";
            for (auto x : lista) {
                cout << " " << x;
            }
            cout << "\n";
            lista.reverse();
            break;

        case 12:
            cout << "Lista Antiga:";
            for (auto x : lista) {
                cout << " " << x;
            }
            cout << "\n";
            lista.sort();
            break;
    }
```

```
        // imprimir a lista resultante
        cout << "Lista Atual:";
        for (auto x : lista) {
            cout << " " << x;
        }
        cout << "\n";

        cout << "\n";
    }

    cout << "Programa finalizado!\n";

    return 0;
}
```