

Lista de exercícios STL

Exercícios:

1. Escreva um programa explorando STL que declara um vetor de inteiros. Armazene no vetor os 5 valores informados e então imprima o vetor de elementos com o comando `cout`.
2. Escreva um programa que explora STL, o qual captura uma sequência arbitrária de dígitos binários com `cin` e armazena em um container. Enquanto o valor recebido é diferente de 1 ou 0. Considere que o primeiro valor inserido é o bit menos significativos. Ao final, apresente a representação binária informada e o valor decimal sem sinal desta representação binária.
3. Implemente um programa que lê `n` palavras da entrada-padrão, e mais uma palavra-chave. Você deve localizar e imprimir as palavras que foram digitadas que possuem a palavra-chave como substring.
 - a. Dica: o método `::find()` retorna o valor específico `string::npos` se não encontrar nada (use-o para teste!)
4. Escreva um programa que recebe de entrada nomes e telefones correspondentes.
 - a. A entrada deve ser `n` nomes e `n` telefones
 - i. Defina uma classe e seus métodos para armazenamento de tais informações
 - b. Na saída, imprima a lista [`<nome> <telefone>`] ordenada pelos nomes
 - i. Crie uma função para comparação
 - ii. Explore o algoritmo `sort` de STL
 - iii. Veja exemplo em:
 1. <http://www.inf.pucrs.br/~pinho/PRGSWB/STL/stl.html>
 - c. Após isso, o programa deve ler um nome da entrada padrão e imprimir seu telefone correspondente, se este existir. Se não existir, imprima uma mensagem apropriada.
5. Uma pilha (stack) é um tipo abstracto de dados que tem associadas as seguintes operações:
 - `push()` – coloca um novo elemento no topo da pilha
 - `pop()` – retira um elemento do topo da pilha
 - `top()` – devolve o elemento no topo da pilha (sem a alterar!)
 - `clear()` – esvazia a pilha
 - a. Defina uma classe genérica em C++ que implemente uma pilha com todas as operações mencionadas e utilize como estrutura de armazenamento um array. (Sugestão: utilize a classe `vector` da STL.)
 - b. Defina uma classe genérica em C++ que implementa uma pilha com todas as operações mencionadas e utiliza como estrutura de armazenamento uma lista ligada. (Sugestão: Utilize a classe `list` da STL.)
6. Considere um sistema de impressão que utiliza duas políticas de atendimento dos trabalhos de impressão:
 - P1 – imprime pela ordem de chegada;
 - P2 – imprime o menor trabalho primeiro.

Assuma que o tempo de impressão por página é uma constante e que o número máximo de páginas é 50.

- a. Gere 10 sessões de impressão e meça o tempo médio que um utilizador espera pelo seu trabalho se utilizar a política P1 em cada sessão. Para cada sessão de impressão gere aleatoriamente T trabalhos, cada um com um número de páginas também aleatório. Para o armazenamento dos trabalhos, utilize a classe queue da STL.
- b. Repita a alínea a) utilizando, agora, a política P2.
- c. Identifique a melhor política, no sentido em que minimiza o tempo médio de espera dos utilizadores.

Referências:

Geral → <http://en.cppreference.com/w/cpp/container>

Vector → <http://en.cppreference.com/w/cpp/container/vector>

List → <http://en.cppreference.com/w/cpp/container/list>