

CLASSES. APONTADORES.

1) Construa uma classe de esferas que contenha métodos para calcular o volume e a área da esfera. Use os respectivos métodos num programa teste.

2) Construa um programa em C++ que efectue as operações básicas com números complexos : adição, subtração, multiplicação e divisão. Para este efeito redefina os operadores $+$, $-$, $*$ e $/$. Acrescente construtores à esta classe para que os objectos `complex` possam ser inicializados de seguinte modo : `complex z(1., -2.) ;`

3) Considera-se um conjunto $C = \{x_1, \dots, x_n\}$ com n elementos e uma relação α definida sobre C . A relação é definida através duma matriz M : os elementos x_i e x_j estão α relacionados (escreve-se $x_i \alpha x_j$) se e só se $M(i, j) = 1$, caso contrário $M(i, j) = 0$. Escreva um programa em C++ que verifique se a relação α é reflexiva, simétrica e transitiva.

4) Construa um programa em C++ que calcule somas de vectores e produtos internos entre vectores. Para este efeito defina os operadores $+$ e $*$ dentro da definição da classe dos vectores.

5) Construa um programa em C++ que calcule somas e produtos de matrizes. Para este efeito defina os operadores $+$ e $*$ dentro da definição da classe das matrizes.

6) É dada uma tabela triangular em que na horizontal e na vertical figuram as localidades dum distrito. A informação é organizada de seguinte modo : se entre duas localidades existir uma estrada de ligação directa, então na tabela, na respectiva entrada está a distância entre as duas localidades, em kilometros. Se não existir estrada de ligação directa entre duas localidades então a respectiva entrada da tabela está preenchida com 0. Escreva um programa em C++ que produza a seguinte informação : uma tabela triangular em que na horizontal e na vertical figuram as localidades do distrito e cujas entradas sejam as mais curtas distâncias entre as respectivas localidades, tomando em conta todos os trajectos possíveis.

7) Implemente em C++ uma classe cujos objectos possam guardar listas de números. Sendo `l` um objecto desta classe, devem ser implementados os seguintes métodos :

a) `l.elem(i)` deve retornar o elemento na posição `i` na lista ou deve exibir uma mensagem de erro se a lista tiver menos de `i` elementos.

b) `l.remove(i)` deve eliminar o elemento na posição `i` na lista ou deve exibir uma mensagem de erro se a lista tiver menos de `i` elementos.

c) `l.card()` deve retornar o número de elementos da lista.

d) `l.inserir(i,x)` deve inserir o elemento `x` na posição `i` na lista ou deve exibir uma mensagem de erro se a lista tiver menos de `i-1` elementos.

8) Implemente em C++ uma classe cujos objectos possam guardar pilhas (stack) de números. Sendo `p` um objecto desta classe, devem ser implementados os seguintes métodos :

a) `p.tirar()` deve retornar o último elemento da pilha (o que foi colocado mais recentemente), ou deve exibir uma mensagem de erro se a pilha estiver vazia.

b) `p.colocar(x)` deve acrescentar o número `x` na pilha `p`.

c) `p.vazio()` deve retornar o valor 1 se pilha estiver vazia e o valor 0 no caso contrário.