

1. Pretende-se uma classe que simule o buffer de documentos que se encontram em espera para imprimir numa impressora de rede. Os documentos são impressos obedecendo a um regime FIFO. A impressora tem uma memória limitada para o buffer de documentos, só podendo receber documentos cujo tamanho seja inferior à diferença entre a memória máxima da impressora e a memória consumida pelos documentos que já se encontram em buffer.
 - a) Apresente a definição da classe **BufferPrinter**, os construtores, destrutor, e o método *AddsDoc*, que permita adicionar ao buffer um novo documento a imprimir, desde que exista memória suficiente para o guardar.
 - b) Acrescente à classe o método *ElimDoc*, que permita eliminar do buffer da impressora um documento com um determinado nome e pertencente a um determinado autor.
 - c) Elabore um método para remover do buffer da impressora todos os documentos com tamanho superior a um determinado valor.
 - d) Faça a sobrecarga do operador stream out << para visualização do buffer da impressora
2. Uma **MultiStack** é um vetor de stacks s_0, s_1, \dots, s_n na qual a stack s_0 funciona como buffer e as restantes stacks s_i $i=1..n$ podem acomodar no máximo $3 \times i$ elementos, estando as stacks preenchidas por ordem crescente do índice i , de tal modo que se a MultiStack possui n stacks, as $n-1$ primeiras stacks estão totalmente preenchidas, só a última poderá não estar. Este sistema MultiStack é usado numa empresa fabril para coleta de peças individuais e peças de montagem. Simplificadamente, para ambos os tipos de peças é registado o código da peça, para as peças individuais o tipo de embalagem necessário para acomodar a peça e para as peças de montagem o peso da peça.
 - a) Defina as classes indicando apenas os atributos e a assinatura dos métodos necessários à modelação de uma aplicação que permita gerir o sistema MultiStack desta empresa fabril, com capacidade máxima $n=50$ stacks.
 - b) Elabore o método de inserção de peças na MultiStack. Note que a inserção de peças numa MultiStack com n stacks faz-se sempre na stack s_0 , com capacidade para k peças, a qual ficando cheia todas as suas peças são transferidas para as demais stacks da MultiStack.
 - c) A remoção de um código de peça na MultiStack pode envolver a remoção de várias peças na mesma stack e/ou em várias stacks, pelo que após a remoção é necessário transferir peças da(s) última(s) stack(s) preenchida(s) para as stacks envolvidas na remoção, de modo a garantir que todas as stacks estão preenchidas por ordem crescente do índice i . Escreva o método de remoção na MultiStack.
 - d) Escreva uma função que calcule o peso total das peças de montagem existentes na MultiStack.

3. Pretende-se definir uma Lista Telefónica com pares número de telefone, nome. A lista deve distinguir entre números de telefone fixo e móveis e validar se os números de telefone só contêm dígitos 0..9, se os dígitos iniciais são válidos e se o número é único.
- a) Usando a classe map apresente a definição da classe **LisTelef** os construtores e o destrutor.
 - b) Elabore um método que insira um par nome/telefone.
 - c) Analise o método anterior quanto à sua complexidade temporal, justifique.
 - d) Elabore um método para apresentar todos os números associados ao mesmo nome. Faça a análise de complexidade temporal do método, justifique.

Recursividade

4.

<pre>void exemplo (int a[], int li, int ls) { if (li < ls) { processa (a,li,ls) ; ls=ls/2 ; exemplo (a,li,ls) ; } }</pre>	<pre>void processa (int a[], int liminf, int limsup) { int i=liminf ; int j=limsup-1 ; while (i < j) { int temp=a[i] ; a[i]=a[j] ; a[j]=temp ; i++ ; j-- ; } }</pre>
--	--

- a) Explique o que faz o código acima e apresente o resultado aplicado ao vetor $a[8]=\{ 6,1,4,2,7,3,1,5\}$, $li=0$, $ls=8$, exemplo (a,0,8)
- b) Analise a sua complexidade temporal, notação BigOh. Justifique.

5. Elabore métodos recursivos que permitam:

- a) Calcular a soma de dois números positivos.
- b) Converter um inteiro decimal n na base b.
- c) Verificar se um número inteiro é primo
- d) Verificar se uma palavra é palíndroma, isto é, se a sua leitura é a mesma da esquerda para a direita ou da direita para a esquerda exemplos: ANA, SOPAPOS
- e) Verificar se um número é capicua, exemplo 1221