Correção: a questão está dividida em casos triviais, completos, visíveis e ocultos. O aluno ganhará os pontos conforme a tabela abaixo. Para ganhar os pontos referentes à modularização (implementação dos tipos complexos e suas funções conforme pedido), o aluno deverá acertar pelo menos os casos visíveis triviais.

Casos	**Pontos**
Visíveis triviais	0,5
Visíveis completos	0,5
Ocultos completos	0,5
Modularização	1,0

- (BOCA: P2_Q4) Problema: Incremente o programa da questão anterior para considerar também o tamanho dos arquivos e um limite de tamanho para as pastas. Após a quantidade de arquivos de cada pasta, será fornecido um número inteiro representando o tamanho máximo que os arquivos dentro daquela pasta podem somar. Além disso, para cada arquivo, será informado um número inteiro indicando seu tamanho.

  O código do programa deve manter-se modularizado. Os tipos da questão anterior devem ser incrementados com os devidos atributos, por exemplo, o tipo *tPasta* deve armazenar também o tamanho máximo permitido. Assim como as funções para resolver este problema também devem ser alteradas, por exemplo, adaptar as funções de leitura para os tipos *tPasta* e *tArquivo*. O uso e acesso dos tipos devem continuar sendo feitos por meio de suas funções.

  Pelo menos, a seguinte função de *tPasta* deverá ser implementada adicionalmente:

  - **int tPasta_cabeArquivo(tPasta pasta, tArquivo arquivo);** Recebe uma pasta e um arquivo como parâmetros. Retorna verdadeiro caso a pasta suporte a inserção do arquivo e falso, caso contrário. Para o arquivo caber na pasta, a soma de todos os arquivos da pasta deve ser menor ou igual ao seu tamanho máximo;

Já para o tipo tArquivo, deverá ser implementada:

- **int tArquivo_qualTamanho(tArquivo arquivo);** Recebe um arquivo como parâmetro e retorna o seu tamanho;

- Entrada: A entrada começa da mesma forma que a questão anterior, exceto que agora, após o número de arquivos de cada pasta, separado por um espaço, será informado um inteiro representando o tamanho máximo suportado pela pasta. Após cada nome de arquivo, também será informado um inteiro separado por espaço indicando o tamanho daquele arquivo. Além disso, no lugar das leituras de filtro, será informado um número Q, significando quantas inserções serão feitas, onde as Q linhas seguintes terão um nome de pasta seguido por todas as informações de um arquivo (extensão, nome do arquivo e tamanho, nessa ordem), todas informações separadas por um espaço. Caso o arquivo informado caiba dentro da pasta requisitada, ele deve ser adicionado a ela, caso contrário, deverá ser impresso um erro.

- Saída: A saída será composta por mensagens de erro. Se um arquivo não couber na pasta especificada, o programa deve imprimir "ERRO: Arquivo nao cabe na pasta X." e se for tentada a inserção de um arquivo em uma pasta que não existe, será impresso "ERRO: Pasta X nao encontrada.", onde, em ambos os casos, X é o nome da pasta. Ao final do programa todas as pastas, serão impressas na mesma ordem fornecida na entrada padrão, assim como todos os seus arquivos. Nos casos simples, as operações de inserção sempre recebem pastas que existem.

  Ver exemplos de formato de entrada e saída nos arquivos fornecidos com a questão.