

Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.
R: A compreensão do enunciado faz parte da prova.

- P: O que será corrigido?
R: A lógica, a criatividade, a sintaxe, o uso correto dos comandos e dos tipos, os nomes das variáveis, a indentação, uso equilibrado de comentários no código e, evidentemente, a clareza. Nesta prova, você deverá sobretudo escrever códigos modulares, usando corretamente funções

e/ou procedimentos, conforme o caso, além de uso correto de variáveis locais ou globais e a passagem de parâmetros por referência ou por valor.

- P: Posso fazer a lápis?
R: Não. A prova deverá ser feita a caneta.
- P: Posso responder na folha de questões?
R: Não. A prova deverá ser respondida na folha de respostas.

(25 pontos) 1) Implemente uma versão modificada do MergeSort com que ele considere os números ímpares sempre maiores que os números pares.

Desta maneira o vetor, sem ordenação, abaixo:

30 44 1 93 44 77

Ficaria ordenado da seguinte maneira:

30 44 44 1 77 93

(25 pontos) 2) Um vetor $v[l..r]$ está separado se existe j em $l..r$ tal que $v[l..j-1] \leq v[j] < v[j+1..r]$. Escreva um algoritmo que decida se $v[l..r]$ está separado. Em caso afirmativo, o seu algoritmo deve devolver o valor de j .

(25 pontos) 3) Escreva uma função que imprima os n maiores elementos de vetor (ordenados) sem que o vetor inteiro seja ordenado, a sua função receberá os seguintes parâmetros: vetor de `Item`, os índices mais a esquerda, l , o mais a direita, r , e um número n , representando quantos maiores elementos devem ser impressos. O protótipo da função é:

```
1 void imprimetopN(Item *v, int l, int r, int n);
```

Por exemplo, para um vetor $\{10, 5, 2, 6, 8, 9, 5, 6\}$ e passando $n = 3$ o resultado deve ser a impressão dos valores $8, 9, 10$. *Lembre-se de não ordenar o vetor inteiro e a sua solução deve ser a mais eficiente possível.*

(25 pontos) 4) Um animado aluno está criando um sistema centralizado que recebe submissões de problemas, no estilo do MOJ, porém ele recebe submissões de diversas regiões do país.

O interessante é que o sistema processa as submissões a cada 5 minutos e por isso recebe muitos dados. Cada região envia um vetor ordenado pelo tempo decorrido da prova em que a submissão foi feita, ou seja, se uma submissão foi feita após 5 minutos de prova ela terá valor 5, se for 15 minutos, será 15.

O sistema deverá processar todas submissões, de todas as regiões. A ordem de processamento das submissões deverá ser ordenada pelo tempo decorrido da prova, ou seja, a submissão mais antiga (menor valor de tempo decorrido da prova), independente da região, é a que deverá ser feita primeiro.

O problema é relativamente simples, pois como todos os vetores recebidos já estão ordenados, basta criar um algoritmo que junte (intercale) todos os vetores em um único vetor.

Foi pedido a você que implemente 2 funções, uma chamada `intercala4` e outra `intercala8`, estas funções recebem, respectivamente, 4 e 8 conjuntos de um ponteiro do tipo `Item` e um inteiro do tipo `int`, sendo, cada um, o vetor ordenada pelo `tempo_de_submissao` e deverá retornar um único ponteiro do tipo `Item` contendo o vetor resultante da intercalação dos vetores recebidos como parâmetro. Você deverá implementar da forma mais eficiente possível.

Considere, nesta questão, que as macros `less(A,B)` e `lesseq(A,B)` existem e sabem comparar as chaves da estrutura deste problema.

Dica: Você pode achar conveniente implementar a função `intercala2`. É possível criar *wrappers*?