

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.

(20 pontos) 1) Sobre algoritmos de ordenação:

- a) (5 pontos) É possível afirmar que o algoritmo QuickSort é superior aos algoritmos elementares (ex: BubbleSort, InsertionSort, SelectionSort) em todos os casos? Porquê?
- b) (10 pontos) Explique alguma estratégia para minimizar o problema apresentado no item (a).
- c) (5 pontos) A respeito da estabilidade de um algoritmo de ordenação, defina o conceito de estabilidade e cite 1 algoritmo estável e 1 não estável.

(30 pontos) 2) 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 ordenado 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 que as seguintes macros estão disponíveis para uso:

```
1 lesseq(A, B)
2 less(A, B)
```

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

(50 pontos) 3) Classificados

ATENÇÃO 1 - Além da solução deste problema você deverá explicar o motivo de sua solução.

ATENÇÃO 2 - Você não pode ordenar a entrada inteira neste problema

ATENÇÃO 3 - Cuidado com as sementes de nota iguais.

A famosa empresa agrônoma, EDA (Estudos e Desenvolvimento Agrônomos), está realizando a coleta de resultados da qualidade das sementes de todos os produtos que podem ser plantados no planeta terra. Um dos objetivos desta seleção está em descobrir se temos alguma semente que poderá ser plantada em um outro planeta, como marte. O problema que esta renomada empresa está enfrentando é como descobrir quais são as melhores sementes.

Laboratórios do mundo todo mandaram informações de muitas sementes, na casa dos bilhões, e a EDA precisa urgentemente da lista (ordenada) das k melhores sementes para realizar o experimento. O grupo técnico da EDA já tentou ordenar esse conjunto de dados e não obteve sucesso, nem mesmo o quicksort com a melhor estratégia consegue ordenar este vetor. E para conseguir resolver este problema, a EDA chamou você para auxiliar nesta difícil tarefa.

Entrada

A entrada possui um único caso de teste. A primeira linha do caso de teste possui um número k ($1 \leq k \leq 10^9$) representando quantas sementes você deverá selecionar (baseada no melhor Valor), a seguir existe um número indeterminado de linhas (sabemos que não passa de 10^9 entradas e você pode assumir que cabe na memória, pois a EDA disponibilizou uma máquina grande). Cada linha da entrada possui 2 números inteiros S ($0 \leq S \leq 2^{50}$) e N ($-10^9 \leq N \leq 10^9$), representando o código da Semente e a Nota da qualidade, respectivamente.

Saída

A saída deverá ser composta por um conjunto de linhas que representam as k melhores sementes. Quanto menor o valor de N melhor é a semente.

A saída deve estar ordenada pelo código da semente (da menor para a maior).

Havendo empate nas notas da semente o desempate será feito pelo menor código de semente.

Exemplo

Entrada:	Saída:
4	
30553 3265	2414 -30329
27183 26616	20027 5010
2414 -30329	26565 -22549
16682 23006	30553 3265
20027 5010	
10315 32560	
23488 17242	
26565 -22549	
2660 23760	
10568 27930	

Entrada:	Saída:
3	
10 232	56 -222
32 656	323 -222
4535 -222	767 -222
56 -222	
767 -222	
943 -222	
323 -222	