



OBI2008

Caderno de Tarefas

Modalidade Programação • Nível Júnior, Fase 1
29 de Março de 2008

A PROVA TEM DURAÇÃO DE TRÊS HORAS

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 3 páginas (não contando esta folha de rosto), numeradas de 1 a 3. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Pascal devem ser arquivos com sufixo *.pas*. Para problemas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada problema.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou disquete, conforme especificado pelo seu professor.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
 - em Pascal: *readln*, *read*, *writeln*, *write*;
 - em C: *scanf*, *getchar*, *printf*, *putchar*;
 - em C++: as mesmas de C ou os objetos *cout* e *cin*.
- Procure resolver o problema de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

Sociedade Brasileira de Computação

www.sbc.org.br

Fundação Carlos Chagas

www.fcc.org.br

OBI

Nome do arquivo fonte: `obi.c`, `obi.cpp` ou `obi.pas`

O principal prêmio da Olimpíada Brasileira de Informática é o convite para os cursos de programação oferecidos no Instituto de Computação da Unicamp, com todas as despesas pagas pela Fundação Carlos Chagas, patrocinadora da OBI. São convidados apenas os competidores que atingem um certo número mínimo de pontos, consideradas as duas fases de provas.

Você foi contratado pela Coordenação da OBI para fazer um programa que, dados os números de pontos obtidos por cada competidor em cada uma das fases, e o número mínimo de pontos para ser convidado, determine quantos competidores serão convidados para o curso na Unicamp. Você deve considerar que

- todos os competidores participaram das duas fases;
- o total de pontos de um competidor é a soma dos pontos obtidos nas duas fases;

Por exemplo, se a pontuação mínima para ser convidado é 435 pontos, um competidor que tenha obtido 200 pontos na primeira fase e 235 pontos na segunda fase será convidado para o curso na Unicamp. Já um competidor que tenha obtido 200 pontos na primeira fase e 234 pontos na segunda fase não será convidado.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha da entrada contém dois números inteiros N e P , representando respectivamente o número de competidores ($1 \leq N \leq 1000$) e o número mínimo de pontos para ser convidado ($1 \leq P \leq 1000$). Cada uma das N linhas seguintes contém dois números inteiros X e Y indicando a pontuação de um competidor em cada uma das fases ($0 \leq X \leq 400$) e ($0 \leq Y \leq 400$).

Saída

Seu programa deve imprimir na *saída padrão* uma única linha contendo um único inteiro, indicando o número de competidores que serão convidados a participar do curso na Unicamp.

Exemplo de entrada	Exemplo de saída
3 100 50 50 100 0 49 50	2

Exemplo de entrada	Exemplo de saída
4 235 100 134 0 0 200 200 150 150	2

Telefone

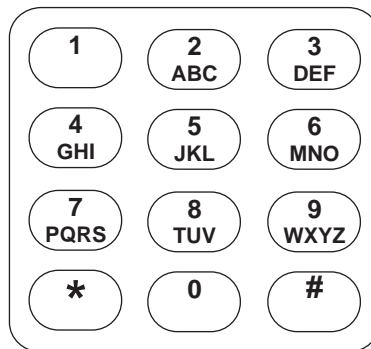
Nome do arquivo fonte: `telefone.c`, `telefone.cpp` ou `telefone.pas`

As primeiras redes públicas de telefonia foram construídas pela AT&T no começo do século XX. Elas permitiam que seus assinantes conversassem com a ajuda de uma *telefonista*, que conectava as linhas dos assinantes com um cabo especial.

Essas redes evoluíram muito desde então, com a ajuda de vários avanços tecnológicos. Hoje em dia, essas redes atendem centenas de milhões de assinantes; ao invés de falar diretamente com uma telefonista, você pode simplesmente discar o número da pessoa desejada no telefone.

Cada assinante recebe um número de telefone — por exemplo, 55-98-234-5678. Qualquer pessoa que discar esse número consegue então falar com a pessoa do outro lado da linha. Os hífens no número de telefone são só para facilitar a leitura, e não são discados no telefone.

Para que fique mais fácil de se lembrar de um número de telefone, muitas companhias divulgam números que contém letras no lugar de dígitos. Para convertê-los de volta para dígitos, a maioria dos telefones tem letras nas suas teclas:



Ao invés de discar uma letra, disca-se a tecla que contém aquela letra. Por exemplo, se você quiser discar o número 0800-FALE-SBC, você na realidade discaria 0800-3253-722.

A sua avó tem reclamado de problemas de vista — em particular, ela não consegue mais enxergar as letrinhas nas teclas do telefone, e por isso queria que você fizesse um programa que convertesse as letras em um número de telefone para dígitos.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A entrada é composta de apenas uma linha, contendo o número de telefone que deve ser traduzido. O número de telefone contém entre 1 e 15 caracteres, que podem ser letras de ‘A’ a ‘Y’ e hífens (‘-’).

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo o número de telefone com as letras convertidas para dígitos. Hífens no número telefone devem ser mantidos no número de telefone de saída.

Exemplo de entrada	Exemplo de saída
PIPOCA	747622

Exemplo de entrada	Exemplo de saída
FALE-SBC	3253-722

Vestibular

Nome do arquivo fonte: `vestib.c`, `vestib.cpp` ou `vestib.pas`

A maioria das universidades brasileiras usa o *vestibular* para selecionar seus alunos. O vestibular consiste de uma ou mais provas sobre as matérias do Ensino Médio, visando avaliar os conhecimentos dos candidatos.

Um formato popular de prova de vestibular é a *prova objetiva*. Neste formato de prova, cada candidato deve escolher uma das cinco alternativas apresentadas pela questão como sendo a correta. Durante a correção dos cartões, cada questão onde a alternativa escolhida pelo candidato é a mesma do gabarito, ele ganha um ponto.

Alguns dos vestibulares mais concorridos do Brasil são disputados por dezenas de milhares de candidatos, e, por isso, geralmente usa-se uma folha de leitura óptica e um programa de computador para corrigir as provas de todos os candidatos e gerar a lista com suas pontuações.

Você trabalha no comitê responsável pelo vestibular em uma faculdade e deve escrever um programa que, dado o gabarito e as respostas de um dos candidatos, determina o número de acertos daquele candidato.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha da entrada contém um único inteiro N ($1 \leq N \leq 80$), representando o número de questões na prova. A segunda linha da entrada contém uma cadeia de N caracteres, indicando o gabarito da prova. A terceira linha da entrada contém outra cadeia de N caracteres, indicando as opções marcadas pelo candidato. Ambas as cadeias contêm apenas os caracteres 'A', 'B', 'C', 'D' e 'E' (sempre em letra maiúscula).

Saída

Seu programa deve imprimir na *saída padrão* uma única linha contendo um único inteiro, indicando o número de acertos do candidato.

Exemplo de entrada	Exemplo de saída
7 AEDBCCE ADDCCBE	4

Exemplo de entrada	Exemplo de saída
5 ABCDE ABCDE	5

Exemplo de entrada	Exemplo de saída
10 ABCDEABCDE BCDEABCDEA	0