



OBI2009

Caderno de Tarefas

Modalidade **Programação** • Nível 1, Fase 1

A PROVA TEM DURAÇÃO DE QUATRO HORAS

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 8 páginas (não contando esta folha de rosto), numeradas de 1 a 8. 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.

Promoção:

Sociedade Brasileira de Computação

Patrocínio:

Fundação Carlos Chagas

Aviões de papel

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

Para descontrair os alunos após as provas da OBI, a Diretora da escola organizou um campeonato de aviões de papel. Cada aluno participante receberá uma certa quantidade de folhas de um papel especial para fazer os seus modelos de aviões. A quantidade de folhas que cada aluno deverá receber ainda não foi determinada: ela será decidida pelos juízes do campeonato.

A diretora convidou, para atuarem como juízes, engenheiros da Embraer, uma das mais bem sucedidas empresas brasileiras, que vende aviões com tecnologia brasileira no mundo todo. O campeonato está programado para começar logo após a prova da OBI, mas os juízes ainda não chegaram à escola. A diretora está aflita, pois comprou uma boa quantidade de folhas de papel especial, mas não sabe se a quantidade comprada vai ser suficiente.

Considere, por exemplo, que a Diretora comprou 100 folhas de papel especial, e que há 33 competidores. Se os juízes decidirem que cada competidor tem direito a três folhas de papel, a quantidade comprada pela diretora é suficiente. Mas se os juízes decidirem que cada competidor tem direito a quatro folhas, a quantidade comprada pela diretora não seria suficiente.

Tarefa

Você deve escrever um programa que, dados o número de competidores, o número de folhas de papel especial compradas pela Diretora e o número de folhas que cada competidor deve receber, determine se o número de folhas comprado pela Diretora é suficiente.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). O arquivo de entrada contém três números inteiros C ($1 \leq C \leq 1000$), P ($1 \leq P \leq 1000$) e F ($1 \leq F \leq 1000$) representando respectivamente o número de competidores, a quantidade de folhas de papel especial compradas pela Diretora e a quantidade de folhas de papel especial que cada competidor deve receber.

Saída

Seu programa deve imprimir, na *saída padrão*, o caractere ‘S’ se a quantidade de folhas compradas pela Diretora é suficiente, ou o caractere ‘N’ caso contrário. Note que os caracteres devem ser letras maiúsculas.

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 30 pontos, $C \leq 10$, $P \leq 10$ e $F \leq 10$.
- Em um conjunto de casos de teste que totaliza 80 pontos, $C \leq 100$, $P \leq 100$ e $F \leq 100$.

Exemplos

Entrada	Saída
10 100 10	S

Entrada	Saída
10 90 10	N

Entrada	Saída
5 40 2	S

Overflow

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

Os computadores foram inventados para realizar cálculos muito rapidamente, e atendem a esse requisito de maneira extraordinária. Porém, nem toda conta pode ser feita num computador, pois ele não consegue representar todos os números dentro de sua memória. Em um computador pessoal atual, por exemplo, o maior inteiro que é possível representar em sua memória é 4.294.967.295. Caso alguma conta executada pelo computador dê um resultado acima desse número, ocorrerá o que chamamos de *overflow*, que é quando o computador faz uma conta e o resultado não pode ser representado, por ser maior do que o valor máximo permitido (em inglês *overflow* significa transbordar).

Por exemplo, se um computador só pode representar números menores do que 1023 e mandamos ele executar a conta $1022 + 5$, vai ocorrer *overflow*.

Tarefa

Dados o maior número que um computador consegue representar e uma expressão de soma ou multiplicação entre dois inteiros, determine se ocorrerá *overflow*.

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 inteiro N ($1 \leq N \leq 500.000$) representando o maior número que o computador consegue representar. A segunda linha contém um inteiro P ($0 \leq P \leq 1000$), seguido de um espaço em branco, seguido de um caractere C (que pode ser '+' ou '*', representando os operadores de adição e multiplicação, respectivamente), seguido de um espaço em branco, seguido de um outro inteiro Q ($0 \leq Q \leq 1000$). Essa linha representa a expressão $P + Q$, se o caractere C for '+', ou $P \times Q$, se o caractere C for '*'.

Saída

Seu programa deve imprimir, na *saída padrão*, a palavra 'OVERFLOW' se o resultado da expressão causar um *overflow*, ou a palavra 'OK' caso contrário. Ambas as palavras devem ser escritas com letras maiúsculas.

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 30 pontos, $N \leq 10$, $P \leq 10$ e $Q \leq 10$.
- Em um conjunto de casos de teste que totaliza 80 pontos, $N \leq 100$, $P \leq 100$ e $Q \leq 100$.

Exemplos

Entrada	Saída
10 5 + 5	OK

Entrada	Saída
44 23 * 2	OVERFLOW

Entrada	Saída
323500 42 * 35	OK

Número de Envelopes

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

Aldo é um garoto muito esperto que adora promoções e sorteios. Como já participou de muitas promoções da forma “para participar, envie n rótulos de produtos ...”, Aldo tem o costume de guardar o rótulo de todos os produtos que compra. Dessa forma, sempre que uma empresa faz uma promoção ele já tem um monte de rótulos para mandar.

A SBC (Super Balas e Caramelos) está fazendo uma nova promoção, e, como era de se esperar, Aldo quer participar. Para participar da promoção é preciso enviar um envelope contendo um rótulo de cada tipo de bala que a SBC produz. Por exemplo, se a SBC produz 3 tipos de balas, A, B, C, e uma pessoa tem 3 rótulos de A, 3 de B e 2 de C, ela pode enviar no máximo 2 envelopes, já que falta um rótulo de C para compor o terceiro envelope. Não há limite para o número de envelopes que uma pessoa pode enviar.

Balas são a segunda coisa de que Aldo mais gosta (a primeira como você sabe são promoções). Por causa disso a quantidade de rótulos de balas que ele tem é muito grande, e ele não está conseguindo determinar a quantidade máxima de envelopes que ele pode enviar.

Como você é o melhor amigo de Aldo ele pediu sua ajuda para fazer o cálculo, de modo que ele compre o número exato de envelopes.

Tarefa

Você deve escrever um programa que, a partir da lista de rótulos de Aldo, calcula o número máximo de envelopes válidos que ele pode enviar.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado).

A primeira linha contém dois números inteiros N ($1 \leq N \leq 1000000$) e K ($1 \leq K \leq 1000$) representando respectivamente a quantidade de rótulos de balas que Aldo possui e o número de tipos diferentes de bala que a SBC produz. Os tipos de balas são identificados por inteiros de 1 a K . A segunda linha contém N números inteiros X_i , cada um representando um rótulo de bala que Aldo possui ($1 \leq X_i \leq K$, para $1 \leq i \leq N$).

Saída

Seu programa deve imprimir, na *saída padrão*, o número máximo de envelopes válidos que Aldo pode enviar.

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 30 pontos, $N \leq 10$ e $X_i \leq 10$.
- Em um conjunto de casos de teste que totaliza 80 pontos, $N \leq 100$ e $X_i \leq 100$.

Exemplos

Entrada	Saída
10 2 1 1 1 1 1 2 2 2 2 2	5

Entrada	Saída
20 5 1 2 3 4 1 2 3 4 1 2 3 4 5 1 2 3 4 5 4 4	2

Feira de Bactérias

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

Bruno é um biólogo apaixonado por sua profissão. Sua especialidade é estudar o comportamento de bactérias. Por isso, ele possui em seu laboratório centenas de colônias de diferentes tipos desses microorganismos.

Nesta semana ele viu o anúncio de um evento inusitado: uma feira de bactérias. Nessa feira, vários fornecedores estarão vendendo diferentes tipos de bactérias. Cada tipo de bactéria é vendido em uma placa de vidro, já preparada para a formação de uma colônia de bactérias. Cada placa de vidro é vendida com apenas uma bactéria inicialmente.

Bruno deu uma olhada no catálogo com os tipos de bactérias que estarão à venda na feira, e notou algumas coisas interessantes:

- Todos os tipos de bactérias à venda terão o mesmo preço.
- Todas as bactérias (de todos os tipos) se subdividem todas as noites para gerar outras bactérias. Por exemplo, a bactéria da colônia de tipo X se subdivide em 2 outras bactérias todas as noites. Assim, no primeiro dia teremos só uma bactéria na colônia. No dia seguinte, teremos 2, e no próximo, 4. A quantidade de divisões de uma bactéria depende do seu tipo.
- O crescimento da colônia cessa após um determinado número de dias, por causa da escassez de alimento. A quantidade de dias em que uma colônia cresce depende do tipo de bactéria.

É final de mês e Bruno já gastou quase todo o seu dinheiro. Assim, resolveu que irá comprar apenas uma colônia de bactérias. No entanto, ele pretende comprar a colônia que forneça a maior quantidade de bactérias ao final do período de crescimento da mesma.

Ele tem um catálogo mostrando os tipos de bactérias à venda. Para cada tipo de bactéria, o catálogo informa a quantidade de bactérias geradas por uma bactéria desse tipo a cada divisão e por quantos dias a população da colônia crescerá. Porém, a calculadora que ele tem em casa não é suficiente para que ele faça os cálculos necessários para decidir qual é a melhor colônia a comprar.

Tarefa

Bruno pediu sua ajuda para decidir qual é o melhor tipo de bactéria para a compra. Lembre que para Bruno o melhor tipo de bactéria é aquele cuja colônia, ao final do período de crescimento, terá a maior quantidade de bactérias.

Você deve supor que não haverá duas colônias com a mesma população final de bactérias.

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 inteiro N ($1 \leq N \leq 50.000$) representando a quantidade tipos de bactérias no catálogo. Cada uma das N linhas seguintes contém informações sobre um tipo de bactéria: a primeira dessas linhas contém a informação da bactéria de tipo 0, a segunda dessas linhas contém a informação sobre a bactéria de tipo 1, e assim por diante. A última dessas linhas contém a informação da bactéria de tipo $N - 1$.

A informação para cada tipo de bactéria é composta por dois números inteiros D e C ($1 \leq D \leq 2.000$ e $1 \leq C \leq 5.000$), onde D é quantidade de bactérias que cada bactéria deste tipo gera ao se dividir numa noite, e C é a quantidade de dias que a população de bactérias crescerá.

Saída

Seu programa deve imprimir, na *saída padrão*, um número inteiro entre 0 e $N - 1$ representando o tipo da bactéria que Bruno deverá comprar.

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 20 pontos, $N \leq 1.000$, $D^C \leq 1.000.000$.
- Em um conjunto de casos de teste que totaliza 80 pontos, $N \leq 2.000$, $D \leq 2.000$, $C \leq 5.000$.

Exemplos

Entrada	Saída
2	1
2 5	
3 4	

Entrada	Saída
5	4
2 1	
4 5	
30 4	
20 6	
2 154	

Entrada	Saída
4	2
145 15	
2 4999	
3 3211	
135 20	