

Competidor(a): \_\_\_\_\_

Número de inscrição: \_\_\_\_\_ – \_\_\_\_\_ (opcional)



# OBI2021

## Caderno de Tarefas

Modalidade Programação • Nível Júnior • Fase 2 (Turno B)

1 de setembro de 2021

A PROVA TEM DURAÇÃO DE 2 HORAS

**Promoção:**



Sociedade Brasileira de Computação

**Apoio:**



# Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 8 páginas (não contando a 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; em particular, seu programa não deve escrever frases como “Digite o dado de entrada:” ou similares.
- 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 necessariamente 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*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python 3 devem ser arquivos com sufixo *.py*; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*.
- Na linguagem Java, **não** use o comando *package*, e note que o nome de sua classe principal deve usar somente letras minúsculas (o mesmo nome do arquivo indicado nas tarefas).
- Para tarefas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada tarefa.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou pen-drive, 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*.
  - em Java: qualquer classe ou função padrão, como por exemplo *Scanner, BufferedReader, BufferedWriter* e *System.out.println*
  - em Python: *read, readline, readlines, input, print, write*
  - em Javascript: *scanf, printf*
- Procure resolver a tarefa 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.

# Recorde

Nome do arquivo: “`recorde.x`”, onde  $x$  deve ser `c`, `cpp`, `pas`, `java`, `js` ou `py`

Atletas conseguem resultados cada vez melhores! O *recorde mundial* de uma determinada modalidade esportiva é o melhor resultado conseguido por um atleta nessa modalidade, em competições oficiais. Competições oficiais incluem campeonatos mundiais, como os campeonatos mundiais de ginástica, atletismo ou natação, e também as Olimpíadas.

Como as Olimpíadas acontecem a cada quatro anos e competições oficiais acontecem todos os anos, é possível que o melhor resultado obtido em Olimpíadas em uma dada modalidade seja um resultado pior do que o recorde mundial para aquela modalidade. Por isso, nas provas das Olimpíadas são sempre mencionados dois records: o recorde olímpico (melhor resultado que já foi obtido em Olimpíadas) e o recorde mundial (melhor resultado em qualquer competição oficial, incluindo as Olimpíadas).

Nesta tarefa, dados o resultado de uma prova nas Olimpíadas e os records mundial e olímpico para essa prova, sua tarefa é determinar se o resultado é um novo recorde mundial e/ou um novo recorde olímpico.

## Entrada

A entrada é composta por três linhas. A primeira linha é um inteiro  $R$ , o melhor resultado obtido por um atleta numa prova das Olimpíadas. A segunda linha é um inteiro  $M$ , o recorde mundial para essa prova. A terceira linha é um inteiro  $L$ , o recorde olímpico para essa prova. Para as provas desta tarefa, quanto menor o valor melhor o resultado.

## Saída

Seu programa deve produzir duas linhas. A primeira linha deve ser `RM` se o resultado é um recorde mundial, ou `*` (asterisco) caso contrário. A segunda linha deve ser `RO` se o resultado é um recorde olímpico, ou `*` (asterisco) caso contrário.

## Restrições

- $1 \leq R \leq 1000$
- $1 \leq M \leq 1000$
- $1 \leq L \leq 1000$

## Exemplos

<p><b>Exemplo de entrada 1</b></p> <p>97 95 98</p>	<p><b>Exemplo de saída 1</b></p> <p>* RO</p>
<p><b>Exemplo de entrada 2</b></p> <p>10 10 10</p>	<p><b>Exemplo de saída 2</b></p> <p>* *</p>

<b>Exemplo de entrada 3</b>	<b>Exemplo de saída 3</b>
78 80 81	RM RO

# Anagrama

Nome do arquivo: “anagrama.x”, onde  $x$  deve ser `c`, `cpp`, `pas`, `java`, `js` ou `py`

Uma palavra  $A$  é um *anagrama* de outra palavra  $B$  se podemos transformar a palavra  $A$  na palavra  $B$  apenas trocando de posição as letras da palavra  $A$ . Por exemplo, “iracema” é um anagrama de “america”, e “estudo” é um anagrama de “duetos”.

Podemos estender o conceito de anagramas para frases, desconsiderando caracteres que não são letras, apenas separam as palavras da frase. Assim, por exemplo, “porta coral” é um anagrama de “claro trapo”. Também não é necessário que a palavra exista em alguma língua: “aca aaa bb b” é um anagrama de “ba.ba,aab ac”.

Dadas duas frases, escreva um programa para determinar se elas são anagramas.

## Entrada

A primeira linha da entrada contém um inteiro  $N$ , indicando o número de letras e espaços das frases. As duas linhas seguintes contêm respectivamente a frase  $A$  e a frase  $B$ , cada linha contendo exatamente  $N$  caracteres, entre letras, espaços em branco, vírgulas e pontos.

## Saída

Seu programa deve produzir uma única linha, contendo um único caractere, que deve ser `S` se a frase for um anagrama ou `N` caso contrário.

## Restrições

- $1 \leq N \leq 200$
- Os únicos caracteres em  $A$  e  $B$  são letras minúsculas, espaços em branco, vírgulas e pontos.

## Exemplos

<p><b>Exemplo de entrada 1</b></p> <p>11 porta coral claro trapo</p>	<p><b>Exemplo de saída 1</b></p> <p>S</p>
<p><b>Exemplo de entrada 2</b></p> <p>1 a b</p>	<p><b>Exemplo de saída 2</b></p> <p>N</p>
<p><b>Exemplo de entrada 3</b></p> <p>12 aca aaa bb b ba.ba,aab ac</p>	<p><b>Exemplo de saída 3</b></p> <p>S</p>
<p><b>Exemplo de entrada 4</b></p> <p>2 a aa</p>	<p><b>Exemplo de saída 4</b></p> <p>N</p>

# Potência

Nome do arquivo: “potencia.x”, onde  $x$  deve ser `c`, `cpp`, `pas`, `java`, `js` ou `py`

A profa. Vilma preparou uma tarefa de programação sobre a *operação de potenciação*. Para lembrar, seja um número real  $n$  e um número inteiro  $p$  igual ou maior do que zero, então a operação de potenciação  $n^p$  tem o valor de  $n$  multiplicado por ele mesmo  $p$  vezes (se  $p = 0$  o resultado da operação de potenciação é 1). Por exemplo,  $2^3 = 2 \times 2 \times 2 = 8$  e  $102^0 = 1$ .

A tarefa preparada pela profa. Vilma foi a seguinte: *Escreva um programa para calcular o valor das seguintes expressões contendo operações de potenciação:*

$$\begin{array}{c} 2^4 + 12^3 \\ 300^3 + 15^2 + 4^2 \end{array}$$

Veja que cada termo das expressões tem a forma  $n^p$  onde  $n$  e  $p$  são números inteiros e  $p$  tem apenas um dígito decimal.

No entanto, quando a profa. Vilma colocou o enunciado da tarefa na Internet, a formatação do enunciado foi corrompida, fazendo com que as expressões aparecessem assim para os alunos:

$$\begin{array}{c} 24 + 123 \\ 3003 + 152 + 42 \end{array}$$

Note que por exemplo  $2^4$  virou 24,  $12^3$  virou 123,  $300^3$  virou 3003 e assim por diante, ou seja, as operações de potenciação desapareceram!

Nesta tarefa, você deve escrever um programa para calcular o valor das expressões da tarefa original da profa. Vilma, sabendo que a formatação do enunciado foi corrompida conforme explicado acima.

## Entrada

A primeira linha da entrada contém um número inteiro  $N$ , o número de termos da expressão. Cada uma das  $N$  linhas seguintes contém um inteiro  $T_i$ , indicando um termo da expressão com formatação corrompida.

## Saída

Seu programa deve produzir uma linha, contendo um único número inteiro, o valor da soma dos termos da expressão, sabendo que a formatação dos termos foi corrompida como explicado acima.

## Restrições

- $1 \leq N \leq 10$
- $10 \leq T_i \leq 9999$  para  $1 \leq i \leq N$
- O resultado é menor do que  $10^9$ .

## Exemplos

Exemplo de entrada 1	Exemplo de saída 1
2 24 123	1744

<b>Exemplo de entrada 2</b>	<b>Exemplo de saída 2</b>
3 303 152 42	27241

<b>Exemplo de entrada 3</b>	<b>Exemplo de saída 3</b>
6 12 13 24 20 31 43	86

# Pesquisa de preços

Nome do arquivo: “`pesquisa.x`”, onde  $x$  deve ser `c`, `cpp`, `pas`, `java`, `js` ou `py`

Uma jornalista está fazendo uma pesquisa de preços de combustíveis (álcool e gasolina), em vários estados do país, para uma reportagem sobre qual dos dois combustíveis é mais vantajoso para abastecer um carro.

Na reportagem ela vai usar a regra de que a utilização do álcool é vantajosa quando o preço por litro do álcool é no máximo igual a 70% do preço por litro da gasolina.

Ela compilou os dados da pesquisa em uma lista contendo o identificador do estado e os preços do litro de álcool e do litro de gasolina, e deseja computar em quais estados é mais vantajoso usar álcool ou gasolina. Você pode ajudá-la?

## Entrada

A primeira linha da entrada contém um número inteiro  $N$ , o número de estados em que a pesquisa foi realizada. Cada uma das  $N$  linhas seguintes contém o identificador do estado,  $E$ , seguido de dois números reais  $A$  e  $G$ , representando respectivamente o preço do litro de álcool e o preço do litro de gasolina.

## Saída

Para cada estado em que o álcool é vantajoso seu programa deve produzir uma linha, contendo somente o identificador do estado, na ordem em que os estados aparecem na entrada. Se em nenhum estado o álcool é vantajoso, seu programa deve imprimir uma linha contendo somente o caractere `*` (asterisco).

## Restrições

- $1 \leq N \leq 27$
- $E$  é uma cadeia de caracteres formada por duas letras maiúsculas sem acento; todos os  $E$  são distintos.
- $0.01 \leq A \leq 10.00$ , com precisão de dois dígitos.
- $0.01 \leq G \leq 10.00$ , com precisão de dois dígitos.

## Exemplos

<p><b>Exemplo de entrada 1</b></p> <p>2 AM 7.00 10.00 RS 7.01 10.00</p>	<p><b>Exemplo de saída 1</b></p> <p>AM</p>
<p><b>Exemplo de entrada 2</b></p> <p>3 SP 4.90 5.80 RJ 4.70 5.70 PR 4.60 5.60</p>	<p><b>Exemplo de saída 2</b></p> <p>*</p>



<b>Exemplo de entrada 3</b>	<b>Exemplo de saída 3</b>
4 SC 5.20 5.72 MT 4.22 6.10 AL 5.55 6.20 GO 4.30 6.25	MT GO