

Competidor(a): _____

Número de inscrição: _____ – _____ (opcional)

Este Caderno de Tarefas não pode ser levado para casa após a prova. Após a prova entregue este Caderno de Tarefas para seu professor guardar. Os professores poderão devolver os Cadernos de Tarefas aos competidores após o término do período de aplicação das provas (13 de setembro de 2024).



Olimpíada Brasileira de Informática
Competição Feminina - OBI2024

Caderno de Tarefas
Modalidade Programação • Nível Júnior • Fase Única

13 de setembro de 2024

A PROVA TEM DURAÇÃO DE 3 HORAS

Promoção:



Sociedade Brasileira de Computação

Apoio:



Coordenação:



Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 13 páginas (não contando a folha de rosto), numeradas de 1 a 13. 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.

Altura da árvore

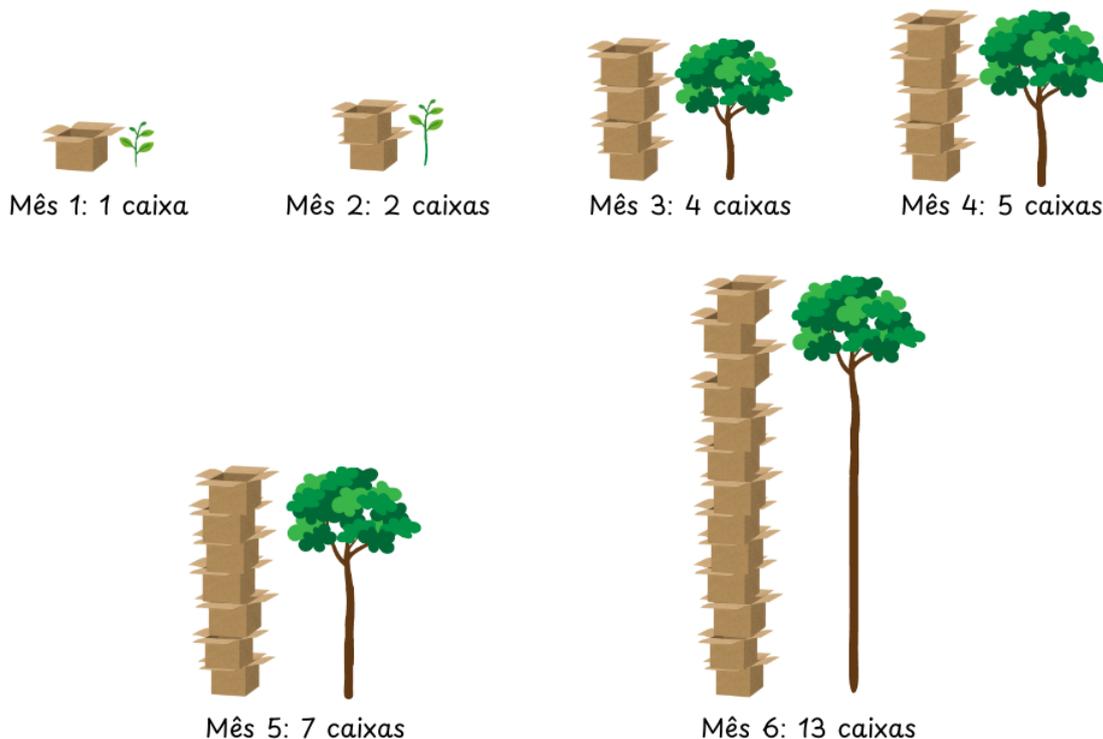
Nome do arquivo: `altura.c`, `altura.cpp`, `altura.java`, `altura.js` ou `altura.py`

Bibi é uma menina apaixonada por natureza. Influenciada por sua mãe, que possui um lindo jardim botânico, ela começou desde cedo a plantar e cuidar de diversos tipos de plantas.

Bibi é extremamente curiosa e sonha em um dia ser uma grande cientista. Por isso, ela registra e acompanha, de forma independente, o crescimento de todas as plantas do jardim, anotando tudo em um livro. Ela também é bastante engenhosa com as ferramentas que possui à sua disposição: como ela não possui uma fita métrica, ela mede a altura das plantas usando caixinhas de papelão que estão prestes a ir para a reciclagem.

Num belo dia, sua mãe lhe trouxe de presente uma semente de *Abratibum*, uma árvore que supostamente vive até 100.000 anos (1.200.000 meses) e que seria a de maior altura já registrada no livro de Bibi. *Uau!*

Bibi começou imediatamente a registrar o crescimento mensal da árvore. Nos 6 primeiros meses, ela obteve os seguintes resultados:



Depois do sexto mês, Bibi observou que o crescimento da árvore parecia ter ficado fixo, e logo ela imaginou que isso poderia ser verdade para todos os meses seguintes. Deste modo, ela fez a seguinte anotação em seu caderno:

Quando a Abratibum chega aos 5 meses de vida, ela atinge a vida adulta. Por isso, a partir de seu sexto mês de vida, a cada mês a Abratibum crescerá em altura a mesma quantidade de caixinhas que cresceu entre o seu quinto e o seu sexto mês de vida.

Bibi notou que não terá caixinhas o suficiente para medir a Abratibum por muitos meses. Por isso, decidiu começar a registrar previsões de crescimento da árvore.

Sua tarefa é, dado um inteiro X , determinar a altura da árvore (em quantidade de caixinhas de papelão) no X -ésimo mês de vida dela.

Entrada

A entrada contém um único inteiro X , o mês de vida da Abratibum a ser consultado.

Saída

Seu programa deve produzir uma única linha contendo um único inteiro, a altura (em caixinhas de papelão) da Abratibum no seu X -ésimo mês de vida.

Restrições

- $5 \leq X \leq 1\,200\,000$

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (23 pontos):** $X = 7$.
- **Subtarefa 3 (27 pontos):** $5 \leq X \leq 12\,000$.
- **Subtarefa 4 (31 pontos):** $5 \leq X \leq 120\,000$.
- **Subtarefa 5 (19 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1 7	Exemplo de saída 1 19
Exemplo de entrada 2 5	Exemplo de saída 2 7

Vale-presente

Nome do arquivo: `vale.c`, `vale.cpp`, `vale.java`, `vale.js` ou `vale.py`

Ana, uma estudante do ensino médio, fez 15 anos recentemente e recebeu diversos vales-presentes de amigos e familiares para usar em suas lojas favoritas, cada um com um valor em reais diferente. Ao usar um vale-presente, Ana pode descontar o valor dele do custo total das compras.

Além disso, as lojas favoritas de Ana possuem uma política de desconto especial, descrito na figura a seguir:



**Desconto adicional de 15 reais
para vales-presentes, iguais ou
acima de 100 reais.**

Ana está muito animada para ir às compras e deseja saber qual o desconto total que ela terá se usar um vale-presente de valor específico. Escreva um programa para ajudar Ana: determine o total de desconto que ela terá ao usar o vale-presente e, se possível, o desconto adicional da loja.

Entrada

A entrada consiste de uma única linha contendo um único inteiro N , o valor do vale-presente em reais.

Saída

Seu programa deverá imprimir uma única linha contendo um único inteiro, o valor total do desconto em reais que Ana terá se usar o vale-presente de N reais.

Restrições

- $1 \leq N \leq 100\,000$

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (40 pontos):** $1 \leq N \leq 100$.
- **Subtarefa 3 (60 pontos):** $100 \leq N \leq 100\,000$.

Exemplos

Exemplo de entrada 1 100	Exemplo de saída 1 115
------------------------------------	----------------------------------

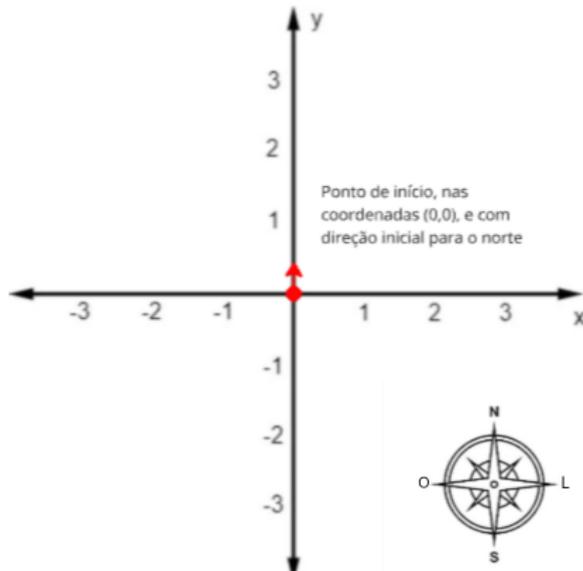
Exemplo de entrada 2 200	Exemplo de saída 2 215
------------------------------------	----------------------------------

Exemplo de entrada 3 50	Exemplo de saída 3 50
-----------------------------------	---------------------------------

Aventura na floresta mágica

Nome do arquivo: `aventura.c`, `aventura.cpp`, `aventura.java`, `aventura.js` ou `aventura.py`

Você é uma aventureira corajosa que se perdeu na Floresta Mágica. A Floresta Mágica é um terreno plano no qual a sua posição é representada por um par de inteiros (x, y) , onde x indica a coordenada no eixo Leste-Oeste e y indica a coordenada no eixo Norte-Sul. Inicialmente, você está no ponto $(0, 0)$ olhando para a direção Norte, mas não sabe onde está a sua casa.



Por sorte, você encontrou um pergaminho secreto milenar que contém uma série de instruções mágicas descrevendo o seu caminho de volta para casa. O pergaminho contém uma lista de comandos, onde cada comando é de um dos dois tipos abaixo:

- *Mova N*: Mova N passos em linha reta na direção para a qual você está olhando.
- *Gire P*: Gire P graus no sentido horário (onde P pode ser 90, 180 ou 270).

Por exemplo, suponha que o pergaminho contém dois comandos, “Gire 90” e “Mova 5”. Após o primeiro comando, você continua na posição $(0, 0)$, porém olhando para a direção Leste. No segundo comando, você anda 5 passos na direção Leste, terminando na posição $(5, 0)$.

Seu objetivo é determinar sua posição final após seguir todas as instruções do pergaminho.

Entrada

A primeira linha de entrada contém um inteiro T , que representa o número de comandos no pergaminho secreto.

As próximas $2 \cdot T$ linhas contêm os comandos, na ordem do pergaminho. Cada par de linhas consecutivas representa um comando. Para cada comando:

- A primeira das duas linhas contém um único caractere K , que pode ser a letra maiúscula ‘M’ ou a letra maiúscula ‘G’. Se $K = \text{‘M’}$, este é um comando Mova. Se $K = \text{‘G’}$, este é um comando Gire.
- A segunda das duas linhas contém um único inteiro. Se este é um comando Mova, o inteiro é N e indica o número de passos que você deve andar na direção atual. Se este é um comando Gire, o inteiro é P e indica quantos graus você deve girar no sentido horário.

Saída

Seu programa deverá imprimir uma única linha contendo suas coordenadas finais X e Y , separadas por um único espaço em branco.

Restrições

- $1 \leq T \leq 10\,000$
- Para todos os comandos, $K = \text{'M'}$ ou $K = \text{'G'}$
- Para um comando com $K = \text{'M'}$: $1 \leq N \leq 10\,000$
- Para um comando com $K = \text{'G'}$: $P = 90$, $P = 180$ ou $P = 270$

Informações sobre a pontuação

A tarefa vale 100 pontos.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
2 G 90 M 5	5 0

Exemplo de entrada 2	Exemplo de saída 2
7 M 10 G 90 M 5 G 90 M 10 G 90 M 5	0 0

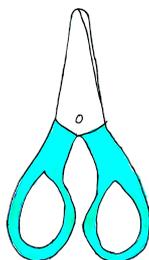
Fábrica de tesouras

Nome do arquivo: `tesoura.c`, `tesoura.cpp`, `tesoura.java`, `tesoura.js` ou `tesoura.py`

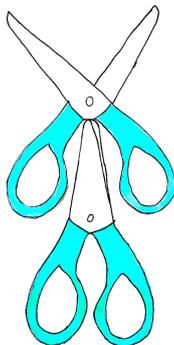
Cecília trabalha em uma fábrica de tesouras. Após o fim da super-promoção de volta às aulas, sobraram muitas tesouras, então ela decidiu usar algumas delas em um quadro para decorar a parede da fábrica. Cecília ama a natureza, e por isso decidiu que seu quadro seguirá o formato de uma árvore.

Cecília fará a montagem de acordo os passos abaixo.

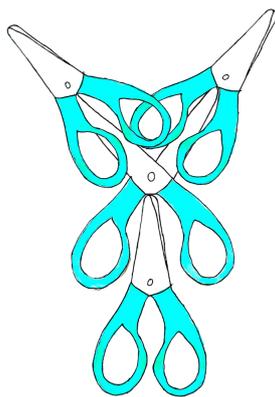
- O primeiro passo é colar uma tesoura fechada em pé na base do quadro. Esta tesoura representa o tronco da árvore:



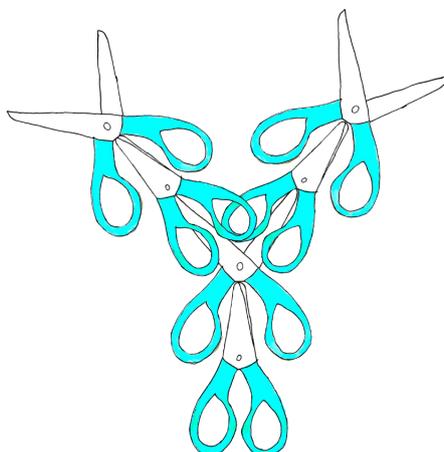
- O segundo passo é colar, na ponta das lâminas da tesoura fechada, uma tesoura aberta. Esta tesoura aberta representa os primeiros galhos da árvore:



- O terceiro passo é colar tesouras fechadas em cima de cada lâmina da tesoura posicionada no passo anterior:



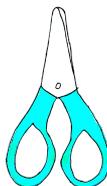
- O quarto passo é colar uma tesoura aberta na ponta das lâminas de cada tesoura colada no passo anterior:



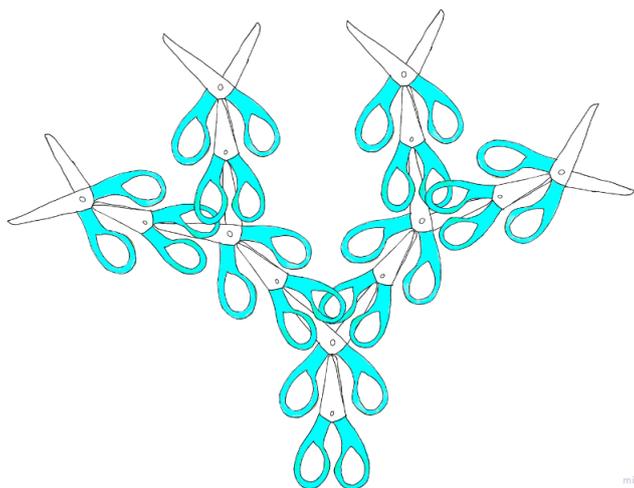
- O quinto passo segue a mesma regra do terceiro passo: colar tesouras fechadas em cima de cada lâmina das tesouras abertas coladas no quarto passo.
- O sexto passo segue a mesma regra do quarto passo: colar tesouras abertas em cima de cada lâmina das tesouras fechadas coladas no quinto passo.
- E assim em diante: Cecília vai alternando os passos entre colar tesouras fechadas e abertas até ficar satisfeita com a altura do quadro.

Cecília chama de *altura* do quadro o número de passos usados para criá-lo.

Por exemplo, o seguinte quadro tem altura 1:



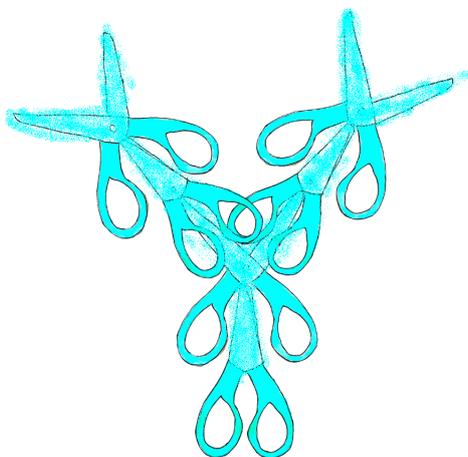
E o seguinte quadro tem altura 6:



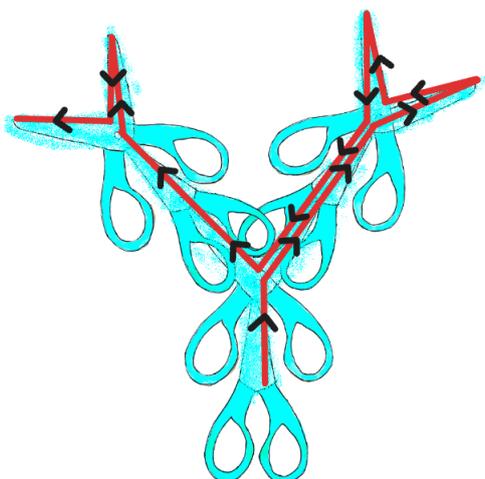
Cecília já fez a colagem de altura K , porém ela ainda não está satisfeita com o quadro: ela está incomodada porque as lâminas das tesouras não possuem a mesma cor que os seus cabos.

Sua amiga, Laura, irá emprestar um tubo de spray de tinta para Cecília pintar a árvore com a cor dos cabos das tesouras. Porém, Laura só permitiu que Cecília use o spray uma única vez. Ou seja, Cecília não pode usar o spray para pintar uma parte da árvore, parar de pintar e recomeçar em uma outra parte da árvore.

Por exemplo, Cecília pode usar o spray para pintar a seguinte árvore, de altura 4:



Para isso, ela pode fazer o seguinte caminho com o spray:



Note que, por só poder usar o spray uma vez, Cecília teve que pintar algumas lâminas de tesouras duas vezes, gastando mais tinta.

Além disso, observe que, quando uma tesoura está fechada, uma passada de spray basta para pintar as lâminas e gastamos apenas 1 ml de tinta:

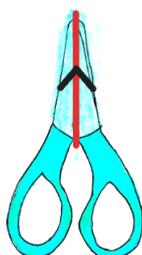


Figura 1: pintamos as lâminas em uma passada, gastando 1 ml

Por outro lado, quando uma tesoura está aberta, temos que passar o spray ao menos uma vez em cada lâmina e gastamos 1 ml de tinta em cada passada:



Figura 2: pintamos a lâmina da direita em duas passadas, gastando 3 ml no total

Cecília sempre começa pintando a árvore a partir de seu tronco (de baixo para cima) e tenta sempre

gastar o mínimo de tinta possível. Na árvore de altura 4, por exemplo, ela gastou 14 ml de tinta.

Sua tarefa é, dada a altura K da árvore que Cecília pintou, determinar a quantidade M de tinta (em ml) que ela usou. Observe que M pode ser muito grande. Se M for maior ou igual a 1 000 000 007, você deve encontrar o resto da divisão de M por 1 000 000 007.

Entrada

A única linha da entrada contém um único inteiro K , a altura da árvore montada por Cecília.

Saída

Seu programa deverá produzir uma única linha contendo um único inteiro, o resto da divisão de M por 1 000 000 007.

Restrições

- $1 \leq K \leq 50\,000$

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (7 pontos):** $K \leq 3$.
- **Subtarefa 3 (4 pontos):** $K \leq 6$ e K é par.
- **Subtarefa 4 (6 pontos):** $K \leq 6$.
- **Subtarefa 5 (19 pontos):** $K \leq 36$.
- **Subtarefa 6 (16 pontos):** $K \leq 54$ e K é par.
- **Subtarefa 7 (20 pontos):** $K \leq 54$.
- **Subtarefa 8 (15 pontos):** $K < 120$.
- **Subtarefa 9 (13 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1 1	Exemplo de saída 1 1
Exemplo de entrada 2 4	Exemplo de saída 2 14

Observação

O “resto da divisão” é o valor que sobra quando um número é dividido por outro. Por exemplo, o resto da divisão de 10 por 4 é 2. Este resto pode ser calculado utilizando o operador % entre o dividendo e o divisor. Por exemplo:

- O resto da divisão de 10 por 4 é calculado por $10 \% 4$
- O resto da divisão de M por 1 000 000 007 é calculado por $M \% 1000000007$