

Problema 1 (digitos)

Trung está entediado com seus deveres de matemática. Em um dos deveres, ele deve escrever uma sequência de inteiros consecutivos começando com 1 a N ($1 < N < 10000$). Por exemplo,

Para $N=3$, sequência é:

123

Para $N=13$, a sequência é:

12345678910111213

Depois disso, ele deve contar o número de vezes que cada dígito (0 a 9) aparece na sequência. Por exemplo, com $N = 13$, o dígito 0 aparece uma vez, o dígito 1 aparece 6 vezes, o dígito 2 aparece 2 vezes, o dígito 3 aparece 3 vezes e cada dígito de 4 a 9 aparece uma vez. Depois de jogar um pouco, Trung ficou entediado novamente. Ele agora quer escrever um programa para fazer isso por ele. Sua tarefa é ajudá-lo a escrever este programa.

Entrada:

Seu programa deve ler da entrada padrão. A entrada consiste em vários conjuntos de dados. A primeira linha do arquivo de entrada contém o número de casos que é um número inteiro positivo e não maior que 20. As linhas a seguir descrevem os conjuntos de dados. Para cada caso de teste, há uma única linha contendo o número N.

Saída

Para cada caso de teste, escreva sequencialmente em uma linha o número de ocorrências de cada dígito 0, 1, . . . 9 separados por um espaço.

Exemplo de entrada

2
3
13

Exemplo de Saída

0 1 1 1 0 0 0 0 0 0
1 6 2 2 1 1 1 1 1 1

Problema 2 (massa)

Um composto orgânico é qualquer membro de uma grande classe de compostos químicos cujas moléculas contêm carbono. A *massa molar* de um composto orgânico é a massa de um mol do composto orgânico. A massa molar de um composto orgânico pode ser calculada a partir dos pesos atômicos padrão dos elementos.

Dr. CHON deseja encontrar sua massa molar de compostos orgânicos a partir de *fórmulas moleculares*. A fórmula molecular mostra quais são os elementos que formam determinada substância e o número exato de átomos de cada elemento que está presente em uma molécula dessa substância.. Por exemplo, na *fórmula molecular* $C_3H_4O_3$ temos 3 átomos de carbono (C), 4 átomos de Hidrogênio (H) e 3 átomos de oxigênio (O).

Neste problema, assumimos que a fórmula molecular é representada por apenas quatro elementos, 'C' (Carbono), 'H' (Hidrogênio), 'O' (Oxigênio) e 'N' (Nitrogênio) sem parênteses. A tabela a seguir mostra os pesos atômicos padrão para 'C', 'H', 'O' e 'N'.

Nome Atômico	Carbono	Hidrogênio	Oxigênio	Nitrogênio
Peso atômico padrão	12.01 g/mol	1.008 g/mol	16.00 g/mol	14.01 g/mol

Por exemplo, a massa molar de uma fórmula molecular C_6H_5OH é 94,108 g/mol que é calculada por $6 \times (12,01 \text{ g/mol}) + 6 \times (1,008 \text{ g/mol}) + 1 \times (16,00 \text{ g/mol})$. Dada uma *fórmula molecular*, escreva um programa para calcular a massa molar da fórmula.

Entrada:

Seu programa deve ler da entrada padrão. A entrada consiste em T casos de teste. O número de casos de teste T é dado na primeira linha da entrada. Cada caso de teste é dado em uma única linha, que contém uma fórmula molecular como uma *string*. O símbolo químico é dado por uma letra maiúscula e o comprimento da *string* é maior que 0 e menor que 80. O número de quantidade n que é representado após o símbolo químico seria omitido quando o número for 1 ($2 \leq n \leq 99$).

Saída:

Seu programa deve escrever na saída padrão. Imprima exatamente uma linha para cada caso de teste. A linha deve conter a massa molar da fórmula molecular dada.

Exemplo de Entrada:

```
4
C
C6H5OH
NH2CH2COOH
C12H22O11
```

Exemplo de saída:

```
12.010
94.108
75.070
342.296
```

Problema 3 (chocolate)

Mohammad visitou recentemente a Suíça. Como ele ama muito seus amigos, ele decidiu comprar um pouco de chocolate para eles, mas como chocolates eram muito caros, ele só podia comprar um chocolate, embora muito grande (parte dele pode ser vista na figura 1) para todos eles como lembrança. Agora, ele quer dar a cada um de seus amigos exatamente uma parte desse chocolate e como acredita que todos os seres humanos são iguais (!), quer dividi-lo em partes iguais.



O chocolate é um retângulo $M \times N$ construído a partir de quadrados de tamanho unitário $M \times N$. Você pode supor que Mohammad também tem amigos $M \times N$ esperando para receber seu pedaço de chocolate. Para dividir o chocolate, Mohammad pode cortá-lo no sentido vertical ou horizontal (através das linhas que separam os quadrados). Em seguida, ele deve fazer o mesmo com cada parte separadamente até atingir o tamanho unitário $M \times N$ de pedaços de chocolate. Infelizmente, por ser um pouco preguiçoso, ele quer usar o número mínimo de cortes necessários para realizar essa tarefa.

Seu objetivo é dizer a ele o número mínimo de cortes necessários para dividir todos os quadrados de chocolate separado.

Entrada

A entrada consiste em vários casos de teste. Em cada linha de entrada, há dois inteiros $1 \leq M \leq 300$, o número de linhas no chocolate e $1 \leq N \leq 300$, o número de colunas no chocolate. A entrada deve ser processada até que o final do arquivo seja encontrado.

Saída

Para cada linha de entrada, seu programa deve produzir uma linha de saída contendo um inteiro indicando o número mínimo de cortes necessários para dividir todo o chocolate em pedaços de tamanho unitário.

Exemplo entrada

```
2 2
1 1
1 5
```

Exemplo Saída

```
3
0
4
```

Problema 4 (codigo)

Brivaldo participa de um grupo de amigos no whatsapp. Um dia ele observou que seus amigos começaram a trocar mensagens com caracteres que não faziam sentido para ele. Ele perguntou no grupo do que se tratava, mas nenhum dos amigos dele respondeu. Após algumas horas no grupo, ele recebeu um fragmento de uma mensagem:

VOCES NAO VAO RESPONDER O BRIVALDO?
JVJLZ'UHV']HV'YLZWVUKLY'V'IYP]HSKVF

Brivaldo ficou com raiva e te procurou para escrever um programa para decodificar as mensagens do grupo. Você consegue perceber que a codificação é simples e que a chave para quebrar o código é uma simples operação aritmética envolvendo caracteres ASCII.

Entrada:

A entrada é composta por múltiplos textos codificados, um em cada linha. Cada texto possui entre 1 em 300 caracteres e a entrada termina com EOF.

Saída:

Para cada linha codificada, você deve imprimir uma linha na saída com o texto decodificado

Exemplo de Entrada

Iyp}hskv'lo'y|pt'kl'ivsh5
L|'wln|lp|t'jovjvsh{l'kh'tvjopsh'klsl5
Lzzl'wyvislth'lo'|t'kvz'thpb'nhjlpz'kh'wyv}h5
Iyp}hskv'wlykl|uv'tvy{hs'rvtih{5
zljvtw979:

Exemplo de Saída

Brivaldo eh ruim de bola.
Eu peguei um chocolate da mochila dele.
Esse problema eh um dos mais facis da prova.
Brivaldo perdeu no mortal kombat.
secomp2023

Problema 5 (labirinto)

Sua missão, se você decidir aceitá-la, é criar um programa de desenho de labirinto. Um labirinto consistirá nos caracteres alfabéticos A-Z, * (asterisco) e espaços.

Entrada

Seu programa obterá as informações dos labirintos do arquivo de entrada. Este arquivo conterá linhas de caracteres que seu programa deve interpretar para desenhar um labirinto. Cada linha do labirinto será descrita por uma série de números e caracteres, onde os números antes de um caractere informam quantas vezes esse caractere será usado. Se houver vários dígitos em um número antes de um caractere, o número de vezes para repetir o caractere será a soma dos dígitos antes desse caractere. A letra minúscula 'b' será usada no arquivo de entrada para representar espaços no labirinto. As descrições de diferentes linhas no labirinto serão separadas por um ponto de exclamação (!), ou por um final de linha. Descrições para diferentes labirintos serão separadas por uma linha em branco. O arquivo de entrada será encerrado por um fim de arquivo.

Saída

Para cada descrição no arquivo de entrada, desenhe o labirinto correspondente conforme mostrado no exemplo de saída abaixo. Não há limite para o número de linhas em um labirinto ou para o número de labirintos em um arquivo, embora nenhuma linha contenha mais de 132 caracteres. Imprima uma linha em branco entre dois labirintos consecutivos.

Feliz labirinto!

Exemplo de Entrada

```
1T1b5T!1T2b1T1b2T!1T1b1T2b2T!1T3b1T1b1T!3T3b1T!1T3b1T1b1T!5T1*1T
```

```
11X21b1X
```

```
4X1b1X
```

Exemplo de Saída

```
T TTTT
T  T TT
T T  TT
T   T T
TTT  T
T   T T
TTTTT*T
```

```
XX   X
XXXX X
```

Problema 6 (gene)

Uma maneira que os cientistas têm para tentar medir como uma espécie evoluiu para outra é investigando como o genoma do ancestral se modificou para se transformar nesta outra espécie. Espécies intimamente relacionadas têm vários genes em comum e verifica-se que uma boa maneira de compará-las é através da comparação de como os genes comuns mudaram de lugar.

Uma das mutações mais comuns que alteram a ordem dos genes de genomas é a inversão. Se modelarmos um genoma como uma sequência de N genes sendo cada gene um número inteiro de 1 a N , então uma inversão é uma mutação que altera o genoma revertendo a ordem de um bloco de genes consecutivos. A inversão pode ser descrita por dois índices (i, j) , $(1 \leq i \leq j \leq N)$, indicando que ela inverte a ordem dos genes dentro de índices de i até j .

Assim, quando isto é aplicado para um genoma $[g_1, \dots, g_{i-1}, g_i, g_{i+1}, \dots, g_{j-1}, g_j, g_{j+1}, \dots, g_N]$, obtém-se o genoma $[g_1, \dots, g_{i-1}, g_j, g_{j-1}, \dots, g_{i+1}, g_i, g_{i+1}, \dots, g_N]$. Como um exemplo, a inversão de $(3, 6)$, aplicado à genoma $[1, 2, 3, 4, 5, 6, 7]$ dá $[1, 2, 6, 5, 4, 3, 7]$. Se depois que a inversão $(1, 3)$ é aplicada, obtém-se o genoma $[6, 2, 1, 5, 4, 3, 7]$.

Um cientista que está estudando a evolução de uma espécie deseja tentar uma série de inversões no genoma desta espécie. Em seguida, ele quer consultar a posição final de vários genes. Será que você aceita o desafio de ajudá-lo?

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um inteiro N indicando o número de genes no genoma ($1 \leq N \leq 50000$). Você pode supor que o ordem inicial dos genes é a sequência de números inteiros de 1 a N em ordem crescente. A segunda linha de um caso de teste contém um inteiro R ($0 \leq R \leq 1000$) que indica o número de inversões a serem aplicadas ao genoma. Então, R linhas seguem, cada uma contendo dois inteiros i, j ($1 \leq i \leq j \leq N$), separados por um único espaço, o qual indicam os dois índices que definem a inversão correspondente. Após a descrição das inversões há uma linha contendo um inteiro Q ($0 \leq Q \leq 100$), que indica o número de consultas para os genes, seguido de Q linhas, onde cada linha contém um inteiro representando um gene cuja posição final você deve determinar.

O final da entrada é indicada por $N = 0$.

Saída

Para cada caso de teste da entrada seu programa deve produzir $Q + 1$ linhas de saída. A primeira linha deve conter a string "Genome", seguido do número do caso de teste. As seguintes Q linhas devem conter um número inteiro, cada um representando as respostas das consultas.

Exemplo de Entrada

9
1
3 6
4
1
3
5
1
5
2
1 2
1 5
2
5
2
0

Exemplo de Saída

Genome 1
1
6
4
1
Genome 2
1
5

Problema 7 (feedback)

Muitos alunos de diversas universidades conhecem o portal de programação IRU. Este portal possui milhares de problemas de programação disponíveis. Diariamente a equipe do IRU recebe diversos feedbacks (elogios, bugs, dúvidas, sugestões, ...) que precisam primeiramente ser atribuídos para membros da equipe resolver.

Como a equipe é muito ocupada e não tem tempo para classificar estes feedbacks, você foi convidado a escrever um programa que faça isso e mostre quem será o membro responsável por resolver e responder o feedback.

Os membros responsáveis em cada setor são:

1. Elogios: Rolien
2. Bugs: Naej
3. Dúvidas: Elehchim
4. Sugestões: Odranoel

Entrada

O primeiro valor a ser lido é o número de casos de teste N ($1 < N < 100$). Cada caso de teste representa um dia de trabalho respondendo feedbacks. Cada caso de teste inicia com K ($1 < K < 50$), indicando o número de feedbacks recebidos naquela data. Seguem K linhas indicando a categoria de cada um dos feedbacks, conforme mostrado acima (1, 2, 3 ou 4).

Saída

Para cada caso de teste você deve imprimir o nome do membro da equipe responsável por responder o feedback.

Exemplo de entrada

```
2
4
1
1
3
4
3
3
3
3
2
```

Exemplo de saída

```
Rolien
Rolien
Elehchim
Odranoel
Elehchim
Elehchim
Naej
```