

Prof. Hebert Coelho
Profa Nádia Félix

Conteúdo

1	Imprimir números naturais usando Recursão	2
2	Programa C para encontrar o número N-Fibonacci usando Recursão	3
3	Batalha Naval	4
4	Altas Aventuras	6
5	O Banco Inteligente	7
6	Setas	8
7	Famílias de Tróia	10

1 Imprimir números naturais usando Recursão



(+)

Escreva um programa em C para imprimir os n primeiros números naturais usando recursão.

Entrada

A única linha da entrada contém um único inteiro n, indicando os n primeiros números naturais.

Saída

Seu programa deve imprimir uma única linha, contendo os n primeiros números naturais separados por um espaço.

Exemplo

Entrada	Saída
50	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

2 Programa C para encontrar o número N-Fibonacci usando Recursão



(+)

Dado um número n , usando recursão, imprimir o n -ésimo número da série fibonacci.

Entrada

A única linha da entrada contém um número inteiro n , indicando a posição do n -ésimo número da série fibonacci.

Saída

Seu programa deve imprimir uma única linha, contendo o n -ésimo número da série fibonacci

Exemplo

Entrada	Saída
8	21

3 Batalha Naval



(++++)

Pedro e Paulo gostam muito de jogar batalha naval; apesar de serem grandes amigos, Pedro desconfia que Paulo não esteja jogando honestamente. Para tirar essa dúvida, Pedro decidiu usar um programa de computador para verificar o resultado do jogo, mas Pedro não sabe programar e por isso pediu a sua ajuda.

O jogo de batalha naval é jogado em um tabuleiro retangular com N linhas e M colunas. Cada posição deste tabuleiro é um quadrado que pode conter água ou uma parte de um navio. Dizemos que dois quadrados são vizinhos se estes possuem um lado em comum. Se duas partes de navio estão em posições vizinhas, então essas duas partes pertencem ao mesmo navio. A regra do jogo proíbe que os quadrados de duas partes de navios distintos tenham um canto em comum (em outras palavras, que quadrados de duas partes de navios distintos compartilhem um vértice).

Cada disparo que um jogador faz deve ser feito em um dos quadrados do tabuleiro do outro jogador. Um jogador informa ao outro a coluna e a linha do quadrado alvo do disparo. Para que um navio seja destruído, o jogador deve acertar todas as partes deste navio. O jogador não pode atirar no mesmo lugar mais de uma vez.

Escreva um programa que, dadas a configuração do tabuleiro e uma sequência de disparos feitos por um jogador, determina o número de navios do outro jogador que foram destruídos.

Entrada

A primeira linha da entrada contém dois números inteiros N e M ($1 \leq N \leq 100$ e $M \leq 100$) representando respectivamente o número de linhas e de colunas do tabuleiro. As N seguintes linhas correspondem ao tabuleiro do jogo. Cada uma dessas linhas contém M caracteres. Cada caractere indica o conteúdo da posição correspondente no tabuleiro. Se esse caractere for '.', essa posição contém água; se for '#', essa posição contém uma parte de um navio. A próxima linha contém um número K que é o número de disparos feitos pelo jogador ($1 \leq K \leq NXM$). As próximas K linhas indicam os disparos feitos pelo jogador. Cada linha contém dois inteiros L e C , indicando a linha e a coluna do disparo feito pelo outro jogador ($1 \leq L \leq N$ e $1 \leq C \leq M$).

Saída

Seu programa deve imprimir uma única linha contendo um único número inteiro, o número de navios destruídos.

Exemplo

Entrada	Saída
5 5 ..#.# #.... ...#. #.... ...#. 5 1 3 1 4 1 5 2 1 3 4	4

Entrada	Saída
5 5 ..### ##### #.#.#. 5 5 1 5 2 1 3 1 4 1 5	2

Entrada	Saída
7 7 .#.....# ###..## .#.....##.# .#...#.# .####.# 8 1 1 1 2 2 1 2 2 2 3 3 2 5 2 6 2	1

4 Altas Aventuras



(+++++)

Incentivado por um filme de animação recente, vovô resolveu realizar seu sonho de criança, fazendo sua pequena casa voar amarrada a balões de hélio. Comprou alguns balões coloridos de boa qualidade, para fazer alguns testes, e começou a planejar a grande aventura. A primeira tarefa é determinar qual a quantidade de hélio máxima que pode ser injetada em cada balão de maneira que ele não estoure. Suponha que os valores possíveis de quantidade de hélio em cada balão variem entre os valores 1 e N. Claro que vovô poderia testar todas as possibilidades, mas esse tipo de solução ineficiente não é apropriada, ainda mais considerando que vovô comprou apenas K balões para os testes. Por exemplo, suponha que $N = 5$ e $K = 2$. Nesse caso, a melhor solução seria testar primeiro em 3. Caso o balão estoure, vovô só teria mais um balão, então teria de testar 1 e 2 no pior caso, somando ao todo 3 testes. Caso o balão não estoure, vovô poderia testar 4 e depois 5 (ou 5 e depois 4), também somando 3 ao todo.

Tarefa

Dados a capacidade máxima da bomba e o número de balões, indicar o número mínimo de testes que devem ser feitos, no pior caso, para determinar o ponto em que um balão estoura.

Entrada

A única linha da entrada contém dois inteiros, N e K, separados por espaço em branco ($1 \leq K \leq N \leq 1.000.000.000$).

Saída

Seu programa deve imprimir uma única linha, contendo um inteiro que representa o número mínimo de testes que devem ser feitos no pior caso para determinar o ponto em que o balão estoura.

Exemplo

Entrada	Saída
5 2	3
20 2	6
11 5	4

5 O Banco Inteligente



(+++)

Caixas automáticos nos bancos são uma invenção ótima mas, às vezes, a gente precisa de dinheiro trocado e a máquina entrega notas de R\$100,00. Outras vezes, a gente quer sacar um valor um pouco maior e por questões de segurança gostaria de receber tudo em notas de R\$100,00, mas a máquina entrega um monte de notas de R\$20,00. O Banco Inteligente está tentando minimizar esse problema dando aos clientes a possibilidade de escolher o valor das notas na hora do saque. Para isso, eles precisam da sua ajuda para saber, dado o valor S do saque e quantas notas de cada valor a máquina tem, quantas formas distintas existem de entregar o valor S . O banco disponibiliza notas de 2, 5, 10, 20, 50 e 100. Por exemplo, se $S = 22$ e o número de notas de cada valor é $N_2=5$, $N_5=4$, $N_{10}=3$, $N_{20} = 10$, $N_{50} = 0$ e $N_{100}=10$, então há 4 formas distintas da máquina entregar o valor do saque: $20+2$, $10+10+2$, $10+5+5+2$ e $5+5+5+5+2$.

Entrada

A primeira linha da entrada contém um inteiro S , o valor do saque. A segunda linha contém seis inteiros N_2 , N_5 , N_{10} , N_{20} , N_{50} e N_{100} , respectivamente, o número de notas de valores 2,5,10,20,50 e 100, disponíveis na máquina.

Saída

Seu programa deve imprimir um inteiro, o número de formas distintas da máquina entregar o saque.

Restrições

- $0 \leq S \leq 5000$ e $N_i \leq 500$

Exemplo

Entrada	Saída
22 5 4 3 10 0 10	4
Entrada	Saída
1000 20 20 20 20 20 20	34201

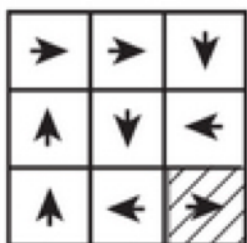
6 Setas



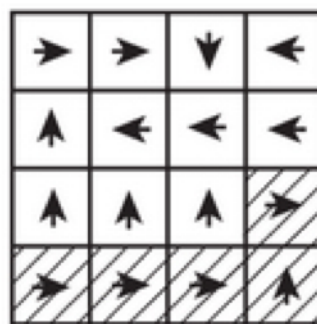
(++++)

Gabriel é um garoto que gosta muito de um jogo onde há várias letras em um tabuleiro e o jogador precisa rapidamente pisar nas letras corretas, de acordo com as instruções na tela, seguindo uma música. Cansado de vencer, Gabriel inventou um novo jogo: agora temos um tabuleiro quadrado, com N células de cada lado, em que cada célula possui uma seta que aponta para uma das quatro posições vizinhas. O jogador primeiro escolhe uma célula inicial para se posicionar e, quando a música começa, ele deve caminhar na direção para onde a seta em que ele está aponta. Ganha o jogo quem pisar em mais setas corretas durante um período de tempo.

O problema é que Gabriel joga tão rápido que quando a seta atual manda ele sair do tabuleiro, ele segue a orientação, muitas vezes quebrando alguns objetos próximos. Quando isso acontece, dizemos que a célula inicial deste jogo não é segura, pois leva a um caminho que termina fora do tabuleiro. A figura abaixo mostra dois tabuleiros.



Tabuleiro 3x3 com oito células seguras



Tabuleiro 4x4 com onze células seguras

Figura 1: Legenda

Ajude Gabriel: dada a configuração do tabuleiro, determine quantas células são seguras para ele iniciar o jogo.

Entrada

A primeira linha da entrada contém um inteiro N , o tamanho do tabuleiro. Cada uma das N linhas seguintes contém N caracteres, com as direções das setas. As direções válidas são:

- ‘V’ Aponta para a célula da linha abaixo, na mesma coluna
- ‘<’ (sinal menor-que) aponta para a célula à esquerda, na mesma linha
- ‘>’ (sinal maior-que) aponta para a célula à direita, na mesma linha
- ‘A’ Aponta para a célula da linha acima, na mesma coluna

Saída

Seu programa deve produzir um único inteiro, o número de células seguras.

Restrições

- $1 \leq N \leq 500$

Exemplo

Entrada	Saída
3 >>V AV< A<>	8

Entrada	Saída
4 >>V< A<<< AAA> >>>A	11

7 Famílias de Tróia



(+++)

A Guerra de Troia pode ter sido um grande conflito bélico entre gregos e troianos, possivelmente ocorrido entre 1300 a.C. e 1200 a.C. (fim da Idade do Bronze no Mediterrâneo). Recentemente foram encontradas inscrições numa caverna a respeito de sobreviventes. Após um trabalho árduo, arqueólogos descobriram que as inscrições descreviam relações de parentesco numa certa população. Cada item da inscrição indicava duas pessoas que pertenciam a uma mesma família. Seu problema é determinar quantas famílias distintas existem.

Entrada

O arquivo de entrada consiste de $M + 1$ linhas. A primeira linha do arquivo de entrada contém um inteiro positivo N , que indica o número de elementos da comunidade, numerados de 1 a N . As demais M linhas do arquivo de entrada contêm, cada uma, dois inteiros. Cada inteiro identifica um elemento da comunidade. Cada linha indica que os dois indivíduos pertencem a uma mesma família.

Saída

A saída deve conter apenas uma linha contendo um único inteiro, que é o número de famílias.

Restrições

- $1 \leq N \leq 5 \times 10^4$
- $1 \leq M \leq 10^5$

Exemplo

Entrada	Saída
4 4 1 2 2 3 3 4 4 1	1

Entrada		Saída
8 10		1
1 2		
2 3		
3 6		
6 5		
5 4		
4 3		
6 7		
7 8		
8 1		
1 5		