Trabalho de PAA - Módulo 03:

Escolher 04 dos 05 para resolver. Entrega: 09/06/2019

Valor: 05 Pontos.

Recomendado C++, podendo ser Java ou Python.

URI Online Judge | 1310

Lucro

Por TopCoder* EUA

Timelimit: 1

George é dono de um circo e traz seu circo de cidade em cidade. Ele sabe o quanto de receita ele pode obter em qualquer dia de uma série de dias em uma cidade. Ele também sabe o custo constante diário para manter o seu circo. George quer trazer seu circo à cidade para a série de dias que resulta em maior lucro.

Por exemplo, se em uma determinada cidade o custo for de $ 20 por dia em um exemplo com 6 dias, sendo que as receitas previstas por dia são {$ 18, $ 35, $ 6, $ 80, $ 15, $ 21}, George pode obter o máximo de lucro trazendo o seu circo para esta cidade do dia 2 ao dia 4. Desta forma ele pode lucrar (35 + 80 + 6) - (3 * 20) = $ 61.

Nota: A série de dias que George traz seu circo para a cidade pode ser entre 0 e o número máximo de dias, inclusive. Obviamente, se George traz seu circo para a cidade por 0 dias, ele obtém $ 0 de lucro.

Entrada

A entrada contém muitos casos de teste. A primeira linha de cada caso de teste contém um inteiro **N** ($1 \leq N \leq 50$) que representa o número de dias que George pode trazer o seu circo para a cidade. A segunda linha do caso de teste contém um número inteiro **custoPorDia** ($0 \leq custoPorDia < 1000$) que representa o custo em manter o circo na cidade. Segue **N** linhas (uma por cada dia), contendo cada um um inteiro **receita** ($0 \leq receita < 1000$) representa a receita que o circo obtem em cada dia. O final da entrada é indicado por EOF (fim de arquivo).

Saída

Para cada caso de teste imprima o máximo de dinheiro que George pode ganhar trazendo o seu circo para a cidade de acordo com o exemplo abaixo.

Exemplo de Entrada	Exemplo de Saída
6 20 18 35 6 80 15 21 4 40 30 20 10 38	61 0

URI Online Judge | 1487

Six Flags

IX Maratona de Programação IME-USP Brasil

Timelimit: 1

O Six Flags Fiesta Texas é um dos maiores parques de diversão do mundo, e fica em San Antonio. Sabendo que as finais do ACM-ICPC de 2006 serão naquela cidade, três colegas começaram a planejar em quais dos famosos brinquedos eles iriam, caso seu time se classificasse para as finais mundiais.

Para isso, estabeleceram notas para cada uma das atrações de acordo com o quanto eles gostariam de brincar lá. Por exemplo, a montanha russa "Superman Krypton Coaster" (que tem 800m de giros, loops e quedas com o carrinho indo a mais de 100km/h) recebeu a maior pontuação possível entre os colegas.

O problema é que é impossível visitar todas as atrações em um mesmo dia. Assim, os colegas pesquisaram, para cada uma delas, quanto tempo durava o brinquedo (e quanto tempo de fila teriam de enfrentar até chegar a ele...). Sua tarefa neste problema é encontrar, dado o tempo disponível pelos colegas no Six Flags, uma coleção (pode haver repetições) de atrações que dá a maior pontuação dentro deste período.

Entrada

Seu programa deve estar preparado para processar diversas instâncias. Na primeira linha são dados dois inteiros $0 \leq \mathbf{N} \leq 100$ e $0 \leq \mathbf{T} \leq 600$, em que **N** é o número de atrações nas quais os colegas gostariam de brincar, e **T** é o tempo (em minutos) que eles terão disponível para isso. Nas próximas **N** linhas, são dados dois inteiros $0 \leq \mathbf{D} \leq 600$ e $0 \leq \mathbf{P} \leq 100$ (em cada linha). O primeiro deles, **D**, representa a duração do brinquedo (incluído aí o tempo de fila e uma estimativa do tempo de traslado entre os brinquedos). O segundo, **P**, representa a pontuaçãao atribuída ao brinquedo pelos colegas. Um valor **N** = 0 indica o final das instâncias e não deverá ser processado.

Saída

Para cada instância solucionada, você deverá imprimir um identificador *Instancia **H*** em que **H** é um número inteiro, sequencial e crescente a partir de 1. Na linha seguinte, deve ser impressa a pontuação total conseguida com a coleção determinada por seu programa. Com relação a quais são as atrações da coleção determinada, os colegas decidiram que iriam perguntar para você pessoalmente no futuro, já que eles não querem que outras pessoas saibam e venham a utilizá-la. Uma linha em branco deve ser impressa após cada caso de teste.

Exemplo de Entrada	Exemplo de Saída
5 60 10 30 20 32 5 4 50 90 22 45 5 60 10 10 20 32 5 4 50 90 22 45 0 0	Instancia 1 180 Instancia 2 104

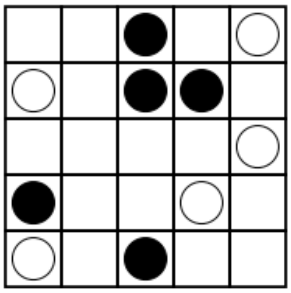
IX Maratona de Programação IME-USP 2005.

Go--

Por Maratona de Programação da SBC – 2016 Brazil

Timelimit: 1

Go-- é até parecido com o tradicional jogo de Go, mas é bem mais fácil! Ele é jogado em um tabuleiro quadrado de dimensão **N**, inicialmente vazio, no qual dois jogadores, um jogando com as pedras pretas e o outro com as brancas, se alternam colocando uma pedra por vez dentro de qualquer célula que ainda não esteja ocupada. A partida termina depois que cada jogador colocou **P** pedras no tabuleiro. Considere todas as possíveis sub-áreas quadradas de dimensão de 1 a **N**. Uma sub-área pertence ao jogador que joga com as pedras pretas se ela contém pelo menos uma pedra preta e nenhuma pedra branca. Da mesma forma, uma sub-área quadrada pertence ao jogador que joga com as pedras brancas se contém ao menos uma pedra branca e nenhuma pedra preta. Note que as áreas que não contenham nenhuma pedra, ou que contenham tanto pedras pretas quanto brancas, não pertencem a nenhum jogador.



Neste problema, dada a posição final do tabuleiro, seu programa deve computar quantas sub-áreas quadradas pertencem a cada jogador, para descobrir quem ganhou a partida. Na figura, as pretas possuem 12 sub-áreas (cinco de dimensão 1, seis de dimensão 2 e uma de dimensão 3). As brancas, que perderam a partida, possuem apenas 10.

Entrada

A primeira linha da entrada contém dois inteiros **N** e **P**, $2 \leq N \leq 500$, $1 \leq P \leq 500$ e $P \leq N^2/2$, representando, respectivamente, a dimensão do tabuleiro e o número de pedras que cada jogador coloca. Cada uma das **P** linhas seguintes contém dois inteiros **L** e **C** ($1 \leq L, C \leq N$) definindo as coordenadas (linha, coluna) das pedras pretas. Depois, cada uma das próximas **P** linhas contém dois inteiros **L** e **C** ($1 \leq L, C \leq N$) definindo as coordenadas (linha, coluna) das pedras brancas. Todas as pedras são colocadas em células distintas.

Saída

Imprima uma linha contendo dois inteiros separados por um espaço: quantas áreas distintas pertencentes às pretas e às brancas.

Exemplos de Entrada	Exemplos de Saída
2 1 1 1 2 2	1 1
5 5 1 3 2 3 2 4 4 1 5 3 1 5 2 1 3 5 4 4 5 1	12 10
500 3 500 498 500 499 500 500 120 124 251 269 499 498	4 12463784

Ginástica

Por Maratona de Programção da SBC, ACM ICPC 2017 Brazil

Timelimit: 1

Vinícius gosta muito de se exercitar na academia de ginástica. Ele fez um acordo com o seu treinador para ter programas de exercícios diferentes a cada vez que usar a bicicleta ergométrica. Um programa, na linguagem das academias, é uma sequência de níveis de dificuldade do exercício. Os programas de Vinícius para a bicicleta ergométrica devem ter a mesma duração em minutos e os níveis de dificuldade devem mudar a cada minuto, para um nível imediatamente acima ou um nível imediatamente abaixo. Os níveis de dificuldade não podem estar abaixo de um mínimo e nem acima de um máximo previamente estipulados.

Seu problema é calcular o número de programas diferentes que o treinador pode construir, obedecidas as restrições acima.

Entrada

A entrada consiste de uma única linha que contém três inteiros, **T**, **M**, **N** ($1 \leq \mathbf{T} \leq 50$, $1 \leq \mathbf{M} < \mathbf{N} \leq 10^5$) em que **T** é o número de minutos do exercício, **M** é o valor mínimo de dificuldade permitido e **N** é o valor máximo de dificuldade permitido.

Saída

Seu programa deve produzir uma única linha com um inteiro representando o número de programas diferentes que o treinador pode construir. Como esse número pode ser grande, a resposta deve ser esse número módulo $10^9 + 7$.

Exemplos de Entrada	Exemplos de Saída
3 2 5	10
30 2 5	4356618
50 1 100000	738072143

Maratona de Programção da SBC-ACM ICPC 2017

URI Online Judge | 2796

Planta da Casa

Por Lucas Maciel Brazil

Timelimit: 1

Isabel reformou sua casa no bairro Ibituruna depois do aumento de faturamento de sua loja. Agora ela quer comprar uma grande mesa de jantar para comemorar. Ela deseja a maior mesa possível que caiba dentro de sua nova casa. E adivinhe quem irá realizar essa tarefa para ela?

Não se desespere, Isabel forneceu tudo para que você resolva essa tarefa. Ela forneceu uma planta da casa informando quais espaços já estão ocupados e que, portanto, não podem conter a mesa. Além disso, ela fez uma lista com o comprimento e a largura de várias opções de mesa. Basta agora descobrir qual é a mesa de maior área que pode ser comprada para que Isabel coloque em sua casa!

Isabel pode rotacionar uma mesa em 90 graus, caso precise.

Entrada

A entrada começa com uma linha contendo dois inteiros N e M separados por espaço, que são as dimensões da planta da casa. Em seguida, há N linhas contendo M caracteres descrevendo a planta. Um caractere . (ponto) representa um espaço vazio, enquanto que o caractere # representa um espaço preenchido (parede ou outro móvel).

A linha seguinte contém um inteiro K que representa o tamanho da lista de opções de mesa que Isabel pode comprar. As próximas K linhas contém, cada uma, dois inteiros C_i e L_i separados por espaço, representando o comprimento e largura da i-ésima mesa.

$1 \leq N, M \leq 1000$

$1 \leq K \leq 10^6$

$1 \leq C_i \leq \min(500, \max(N, M))$

$1 \leq L_i \leq \min(500, \max(N, M))$

Saída

Escreva na saída uma linha contendo dois inteiros separados por espaço: o comprimento e a largura da da mesa de maior área que cabe na casa de Isabel. Em caso de empate, imprima as dimensões da mesa de maior largura. É garantido que sempre há pelo menos uma mesa que cabe na casa de Isabel.

Exemplos de Entrada	Exemplos de Saída
13 32 ################################ #......##.....##.....# #......##.....##.....# #......##.....##.....# #......##....#############..# #..######.....#..............# #..............................# #..######.....###.....# #......####.....###.....# #......##.....#.....###.....## #......##.....#..............## #......##.....#.....##### ################.############## 4 3 3 3 6 15 2 7 7	7 7
13 32 ################################ #......##.....##.....# #......##.....##.....# #......##.....##.....# #......##....#############..# #..######.....##..............# #..............................# #..######.....##...###.....# #......####.....###.....# #......##.....#.....#####.....## #......##.....#..............## #......##.....#.....##### ################.############## 4 3 3 3 6 15 2 7 7	15 2