

Centro de Educação Superior a Distância do
Estado do Rio de Janeiro – CEDERJ

Curso de Tecnologia em Sistemas de Computação – TSC

EAD-05.009 Fundamentos de Programação

Caderno de Exercícios

Aula 4

(Funções, Passagem de Parâmetros e Recursividade)

Professores

Dante Corbucci Filho
Leandro A. F. Fernandes

Instruções

- Utilize Python 3 e a IDE PyCharm na elaboração de soluções para os problemas propostos;
- A entrada de cada problema deve ser lida da entrada padrão (teclado);
- A saída de cada problema deve ser escrita na saída padrão (tela);
- Siga o formato apresentado na descrição da saída, caso contrário não é garantido que a saída emitida será considerada correta;
- Na saída, toda linha deve terminar com o caractere `'\n'` ;
- Utilize o URI Online Judge (<http://www.urionlinejudge.com.br>) e submeta sua solução para correção automática.

Referências Autorais

Os exercícios apresentados nesta lista foram extraídos do URI Online Judge (<http://www.urionlinejudge.com.br>). Acesse a URL apresentada abaixo do título de cada problema para proceder com a correção automática de sua solução e, também, para consultar a autoria do enunciado.

Problema A: Fila do Bandeirão

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1524>

Um fenômeno muito comum na fila da bandeirão (também conhecido como restaurante universitário) é ver uma pessoa recém-chegada entrar no interior na fila em vez de no final. Isso ocorre sempre que tal pessoa encontra alguém de seu grupo já na fila.

Interessado em estudar esse fenômeno, um amigo pediu para você escrever um programa para estudar os grupos presentes na fila. Podemos supor que existem K grupos diferentes e toda pessoa pertence a exatamente um desses grupos. O tamanho de um grupo é definido pela distância entre as duas pessoas mais distantes dentro do grupo. Se o grupo consiste de apenas uma pessoa, seu tamanho é zero. Considerando que os grupos se organizam de forma que a soma dos tamanhos dos K grupos seja mínima, seu programa deve determinar qual é o valor dessa soma.

Entrada

A entrada é composta por diversas instâncias e termina com o final de arquivo (EOF). A primeira linha de cada instância contém os inteiros N , indicando o número de pessoas na fila, e K , indicando o número de grupos ($1 \leq K < N \leq 1.000$). Na linha seguinte são apresentados $N - 1$ inteiros, a_2, \dots, a_N , ($0 \leq a_2 \leq \dots \leq a_N \leq 1.000.000$) indicando as posições de cada pessoa em relação à primeira pessoa da fila. A posição da primeira pessoa é omitido, pois é sempre zero.

Saída

Para cada instância, imprima uma única linha contendo o valor mínimo que a soma dos tamanhos dos K grupos pode ter.

Exemplo

Entrada	Saída
5 2	3
1 2 5 6	0
4 3	
0 1 2	

Problema B: Jogo de Varetas

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1366>

Há muitos jogos divertidos que usam pequenas varetas coloridas. A variante usada neste problema envolve a construção de retângulos. O jogo consiste em, dado um conjunto de varetas de comprimentos variados, desenhar retângulos no chão, utilizando as varetas como lados dos retângulos, sendo que cada vareta pode ser utilizada em apenas um retângulo, e cada lado de um retângulo é formado por uma única vareta. Nesse jogo, duas crianças recebem dois conjuntos iguais de varetas. Ganha o jogo a criança que desenhar o maior número de retângulos com o conjunto de varetas.

Dado um conjunto de varetas de comprimentos inteiros, você deve escrever um programa para determinar o maior número de retângulos que é possível desenhar.

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um inteiro N que indica o número de diferentes comprimentos de varetas ($1 \leq N \leq 1.000$) no conjunto. Cada uma das N linhas seguintes contém dois números inteiros C_i e V_i , representando respectivamente um comprimento ($1 \leq C_i \leq 10.000$) e o número de varetas com esse comprimento ($1 \leq V_i \leq 1.000$). Cada comprimento de vareta aparece no máximo uma vez em um conjunto de teste (ou seja, os valores C_i são distintos). O final da entrada é indicado por $N = 0$.

Saída

Para cada caso de teste da entrada seu programa deve produzir uma única linha na saída, contendo um número inteiro, indicando o número máximo de retângulos que podem ser formados com o conjunto de varetas dado.

Exemplo

Entrada	Saída
1	1
10 7	3
4	2
50 2	
40 2	
30 4	
60 4	
5	
15 3	
6 3	
12 3	
70 5	
71 1	
0	

Problema C: O jogo Matemático de Paula

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1192>

Paula simplesmente adora matemática. Seu maior passatempo é ficar inventando jogos ou atividades que a envolvam para brincar com seus amiguinhos. Obviamente, nem todos eles não são tão apaixonados assim por matemática e têm muita dificuldade para resolver as brincadeiras propostas por ela. Agora Paula inventou um pequeno passatempo que envolve 3 caracteres: um dígito numérico, uma letra e outro dígito numérico.

Se a letra for maiúscula, deve-se subtrair o primeiro dígito do segundo. Se a letra for minúscula, deve-se somar ambos os dígitos e se os DÍGITOS forem iguais, deve-se desconsiderar a letra e mostrar o produto entre os dois dígitos. Ela pediu para seu amigo Marcelo, que é bom em programação, para criar um programa para que encontre a solução para cada uma das sequências que Paula lhe apresentar.

Entrada

A entrada contém vários casos de teste. A primeira linha da entrada contém um inteiro N , indicando o número de casos de teste que virão a seguir. Cada caso de teste é uma sequência de três caracteres criada por Paula. Esta sequência contém na primeira posição um caractere de '0' a '9', na segunda posição uma letra maiúscula ou minúscula do alfabeto e na terceira posição outro caractere de '0' a '9'.

Saída

Para cada caso de teste, deve ser impressa uma linha com um valor inteiro que representa a solução da sequência proposta por Paula.

Exemplo

Entrada	Saída
5	1
4A5	9
3A3	6
4f2	2
2G4	-6
7Z1	

Problema D: Loteria de Fim de Semana

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1407>

Algumas pessoas são contra loterias por motivos morais, alguns governos as proíbem, mas com a criação da internet essa popular forma de aposta, que teve início na China e ajudou a financiar a Grande Muralha, está prosperando.

Mas as chances de se ganhar em uma loteria nacional são pequenas, por conta disso seus colegas de classe decidiram organizar uma loteria particular, cujo sorteio se realiza toda sexta-feira. A loteria é baseada em um estilo popular: um estudante que quer apostar escolhe C números distintos entre 1 e K e paga US\$ 1.00 (note que as loterias tradicionais como a US National Lotto usam $C=6$ e $K=49$). Na sexta-feira durante o almoço, C números (também de 1 a K) são sorteados. O estudante que acertar a maior quantidade de números sorteados recebe o montante coletado nas apostas. O montante é dividido no caso de empates e acumulado para a próxima semana se ninguém acertar qualquer um dos números sorteados.

Alguns de seus colegas não acreditam nas leis da probabilidade e pediram para você para escrever um programa que determine os números que foram sorteados o menor número de vezes considerando todos os sorteios prévios, para que eles possam apostar nesses números.

Entrada

A entrada contém diversos casos de teste. A primeira linha de um caso de teste contém três inteiros N , C e K que indicam, respectivamente, o número de sorteios que já aconteceram ($1 \leq N \leq 10000$), quantos números compõem uma aposta ($1 \leq C \leq 10$) e o valor máximo que pode ser escolhido numa aposta ($C < K \leq 100$). Cada uma das próximas N linhas contém C inteiros distintos X_i indicando os números sorteados em cada concurso prévio ($1 \leq X_i \leq K$, para $1 \leq i \leq C$). O fim da entrada é indicado por $N = C = K = 0$.

Saída

Para cada caso de teste, seu programa deve escrever uma linha de saída, contendo o conjunto de números que foram sorteados o menor número de vezes. Este conjunto deve

ser impresso como uma lista em ordem crescente. Deixe um espaço em branco entre dois números consecutivos na lista.

Exemplo

Entrada	Saída
5 4 6 6 2 3 4 3 4 6 5 2 3 6 5 4 5 2 6 2 3 6 4 4 3 4 3 2 1 2 1 4 4 3 2 1 4 3 0 0 0	1 1 2 3 4

Problema E: Movimentos de Cavalo

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1100>

Pedro está fazendo uma pesquisa sobre o problema do movimento do cavalo em um tabuleiro de xadrez e incumbiu você da tarefa de encontrar o menor conjunto de movimentos possíveis, podendo sair de qualquer quadrado a e podendo chegar em qualquer quadrado b dentro do tabuleiro, sendo que a e b são quadrados diferentes. Ele pensa que a parte mais difícil do problema é determinar o menor número de movimentos do cavalo entre 2 quadrados fornecidos e que uma vez que você está comprometido com esta tarefa, encontrar a sequência de movimentos entre estes 2 quadrados será uma tarefa muito fácil.

É claro que você sabe que o movimento é vice-versa. Portanto você deve fornecer a Pedro um programa que resolva esta questão.

Seu trabalho então será escrever um programa que, pegando dois quadrados a e b como entrada, determine o número de movimentos para encontrar a rota mais curta de a até b.



Entrada

A entrada contém um ou mais casos de teste. Cada caso de teste consiste de uma linha contendo dois quadrados separados por um espaço. Um quadrado será uma string

consistindo de uma letra (a-h) representando a coluna e um dígito (1-8) representando a linha do tabuleiro de xadrez (veja figura acima).

Saída

para cada caso de teste imprima uma linha dizendo "To get from *xx* to *yy* takes *n* knight moves.". No caso *xx* é a origem, *yy* é o destino e *n* é a quantidade de movimentos necessários para ir de *xx* até *yy*.

Exemplo

Entrada	Saída
e2 e4	To get from e2 to e4 takes 2 knight moves.
a1 b2	To get from a1 to b2 takes 4 knight moves.
b2 c3	To get from b2 to c3 takes 2 knight moves.
a1 h8	To get from a1 to h8 takes 6 knight moves.
a1 h7	To get from a1 to h7 takes 5 knight moves.
h8 a1	To get from h8 to a1 takes 6 knight moves.
b1 c3	To get from b1 to c3 takes 1 knight moves.
f6 f6	To get from f6 to f6 takes 0 knight moves.