

14ª Maratona de Programação

30 de agosto de 2014

Caderno de Problemas

Regras: ACM *International Collegiate Programming Contest*

Brasil – **Desafio ALGAR TELECOM**

(Este caderno contém 11 problemas; as páginas estão numeradas de 1 a 13)

Participantes:

CESUC
FEIT-UEMG
FPU
IFTM Uberaba
IFTM Uberlândia
IFTM Ituiutaba
Pitágoras
UNIFEI
UFU
UNIPAM
UNITRI
UFG
UFV
ULBRA
UNIUBE

Apoio: <http://crbonilha.com/pt/>

Maratona de Programação – Instruções

Linhas de compilação:

Linguagem	Linha de comando de compilação
C	gcc -lm teste.c -oteste
C++	g++ -lm teste.cpp -oteste
Java	javac teste.java

Linhas de execução:

Linguagem	Linha de comando de execução
C	./teste < entrada.in
C++	./teste < entrada.in
Java	java teste < entrada.in

Exemplo de problema:

Problema Teste

Arquivo fonte: teste.c, teste.cpp ou teste.java.

Para um dado valor n , calcular a expressão $3n + 1$.

Entrada

A entrada contém várias linhas. Cada linha contém um inteiro n ($1 \leq n \leq 100$).

Saída

Para cada caso de teste imprima uma linha contendo o resultado da expressão.

Exemplo de entrada

```
8
3
12
2
```

Exemplo de saída

```
25
10
37
7
```

Exemplo de solução em C++ (teste.cpp):

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    while(cin >> n)
    {
        cout << 3*n+1 << "\n";
    }
}
```

Exemplo de solução em C (teste.c):

```
#include <stdio.h>
int main()
{
    int n;
    while(scanf("%d", &n) == 1)
    {
        printf("%d\n", 3*n+1);
    }
}
```

Exemplo de solução em Java (teste.java):

```
import java.util.Scanner;
public class teste{
    public static void main(String []
args){
        int n;
        Scanner s = new
Scanner(System.in);
        try{
            while(true){
                n = s.nextInt();
                System.out.println(3*n+1);
            }
        } catch(Exception e){}
    }
}
```

Exemplo do arquivo de entrada (entrada.in):

```
8
3
12
2
```

Obs.: para programas criados em Java, o nome da classe deve ser o mesmo que o nome do arquivo. No exemplo acima, como o arquivo fonte deve ser teste.java, a classe principal deve-se chamar teste.

Apoio: <http://crbonilha.com/pt/>

Problema A

A Linguagem

Arquivo fonte: linguagem.c, linguagem.cpp ou linguagem.java

Germanio um guerreiro intergaláctico e conquistador de planetas tem uma dificuldade para falar quando ele fica nervoso, ele fica um pouco gago. Então, palavras como QUERO, ele fala QQUERO, VAMOS, ele fala VVAMOS e assim por diante.

Aborrecido com esta situação, Germanio decidiu que toda nova conquista de um novo planeta ele iria inventar uma nova língua. Dado um alfabeto, onde não ocorre repetição de caracteres, por exemplo, QABCDEFG, todas as palavras da nova língua começariam com a letra Q duas vezes. Neste caso, a palavra QQABCDEFG seria válida.

Você analisou o caso, e ainda fez a sugestão para que os caracteres a serem repetidos possam ocorrer em qualquer parte da nova palavra desde que sejam na mesma ordem e sempre juntos. No exemplo dado, ABCDEFGQQ seria válido também.

O amado guerreiro Germanio gostou de sua ideia e pediu para você calcular quantas palavras estes novos idiomas terão. Mas, se o idioma tiver um número muito grande de palavras ele quer descartar este idioma.

Entrada

Haverá diversos casos de teste. Cada caso de teste inicia com dois números inteiros N e Q ($1 \leq N \leq 100000$, $1 \leq Q < N$), indicando o tamanho do alfabeto e o número de caracteres do alfabeto que será considerado na repetição que pode ocorrer em qualquer parte da palavra, respectivamente. A segunda linha é composta por um inteiro T ($1 \leq 10^{1000}$) indicando o número máximo de palavras permitido por idioma.

O último caso de teste é indicado quando $N = Q = 0$, o qual não deverá ser processado.

Saída

Para cada caso de teste, imprima uma linha contendo um inteiro, indicando o número de palavras distintas que esta nova língua terá. E imprimirá “descartado” caso o número de palavras ultrapasse o valor de T.

Exemplos de Entrada

```
4 1
100
5 2
30
6 3
10
0 0
```

Exemplos de Saída

```
24
24
descartado
```

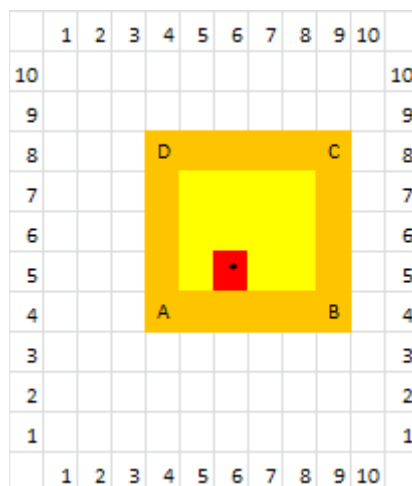
Apoio: <http://crbonilha.com/pt/>

Problema B

Colisão

arquivo: colisao.c , colisao.cpp, colisao.java

Você recebeu a missão de verificar se o robô invadiu uma área retangular formada por quatro pontos cardeais (A,B,C e D). Serão informados os quatro pontos de um plano cardinal conforme a figura. A área será formada pela ligação dos quatro pontos da seguinte forma A-B, B-C, C-D e D-A. Será informado ainda a coordenada X,Y do robô.



Entrada

A entrada é composta de vários casos de testes. A primeira linha é formada por um número N indicando o total de casos de testes. As próximas N linhas são constituídas por 10 números inteiros (A_x , A_y , B_x , B_y , C_x , C_y , D_x , D_y , R_x , R_y) representando cada um dos vértices A, B, C e D e pela posição X, Y do robô. Cada valor é separado por um espaço em branco.

Saída

A saída deverá imprimir para cada caso de testes o número 1, se o robô estiver dentro da área, e imprimir o número 0 caso contrário.

Exemplo de Entrada

```
5
3 6 6 6 6 5 3 5 5 4
1 1 7 1 7 7 1 7 4 2
1 4 7 4 7 6 1 6 5 5
6 2 9 2 9 6 6 6 1 7
4 3 9 3 9 5 4 5 10 7
```

Exemplo de Saída

```
0
1
1
0
0
```

Apoio: <http://crbonilha.com/pt/>

Problema C

Baile de Formatura

arquivo: baile.c , baile.cpp, baile.java

É final de ano, e finalmente Rafael está se formando em seu curso de Computação. O pessoal da sua sala resolveu comemorar a formatura organizando um baile, onde haveria música ao vivo, comida e bebida grátis. Como todo baile, o momento mais esperado é aquele em que todos começam a dançar em pares.

Os pares serão formados entre um garoto e uma garota, e como os alunos da sala de Rafael são muito tímidos, decidiram definir com antecedência quais seriam os pares. Há apenas um problema: há mais garotos do que garotas na sala. Isso implica que, para que todos consigam dançar ao menos uma vez, uma ou mais garotas terão que dançar com mais de um garoto.

Rafael pediu sua ajuda: de quantas maneiras os pares podem ser formados, de tal forma que todos os garotos dance exatamente uma vez, e que todas as garotas dance ao menos uma vez?

Entrada

Haverá diversos casos de teste. Cada caso de teste consiste de dois inteiros, B e G ($1 \leq G < B \leq 10^3$), indicando o número de garotos e garotas na sala, respectivamente.

O último caso de teste é indicado quando $B = G = 0$.

Saída

Para cada caso de teste imprima uma linha, contendo um inteiro, indicando quantas maneiras é possível que os pares sejam formados de tal modo que todos os garotos dance exatamente uma vez, e que todas as garotas dance ao menos uma vez.

Como o resultado pode ser muito alto, imprima o resultado com resto de divisão em 1000000007 (10^9+7).

Exemplo de Entrada

```
2 1
3 2
4 2
4 3
0 0
```

Exemplo de Saída

```
1
6
14
36
```

Apoio: <http://crbonilha.com/pt/>

Problema D

Labirinto

arquivo: labirinto.c , labirinto.cpp, labirinto.java

Labirinto de papel é o passatempo favorito de Rafael, mas ele anda reclamando que os labirintos que ele encontra para resolver são muito fáceis. Para ser mais específica, a distância entre o início do labirinto e a saída é sempre muito pequena.

A entrada de um labirinto é por onde o jogador deve começar a resolvê-lo, e a saída é por onde o jogador deve terminar o labirinto. O jogador pode dar passos nas quatro direções – cima, direita, baixo ou esquerda – e a distância entre a entrada e a saída do labirinto é dado pela soma de passos do menor caminho que pode ser feito.

Dado um labirinto de N linhas e M colunas, diga qual a distância máxima que pode ser definida se a entrada e a saída forem escolhidas de forma ótima.

Entrada

A entrada contém diversos casos de teste. Cada caso de teste inicia com dois inteiros N e M ($5 \leq N, M \leq 500$), representando o número de linhas e colunas do labirinto, respectivamente.

A seguir haverá N linhas contendo M caracteres cada, representando o labirinto a ser analisado. O caractere da i-ésima linha e da j-ésima coluna indica o que há na posição [i, j] do labirinto. Se o caractere for o "." (ponto), significa que aquele é um espaço vazio, por onde o jogador pode passar. Caso seja um "#", significa que aquele é um obstáculo, por onde o jogador não pode passar.

Haverá sempre ao menos dois espaços vazios no labirinto, e só há um caminho entre quaisquer dois espaços vazios. A entrada e a saída do labirinto não necessariamente precisam estar nas bordas.

O último caso de teste é indicado quando $N = M = 0$, o qual não deverá ser processado.

Saída

Para cada caso de teste imprima uma linha, contendo um inteiro, indicando a distância entre a entrada e a saída do labirinto se a localização da entrada e da saída do labirinto for escolhida de forma ótima.

Exemplo de Entrada

```
5 5
.#...
...##
.#...#
.##..
```

```
5 5
.....
####.
.....
.####
.....
```

```
0 0
```

Exemplo de Saída

```
8
16
```

Apoio: <http://crbonilha.com/pt/>

Problema E

Lâmpadas

arquivo: lamp.c , lamp.cpp, lamp.java

No último trabalho da escola, você desenvolveu um projeto um tanto quanto curioso. Trata-se de N lâmpadas dispostas uma ao lado da outra, enumeradas de 1 até N , da esquerda para a direita, e de um único interruptor. Quando este interruptor é pressionado, ele troca o estado de uma das lâmpadas (se está ligada, ela desliga, e vice-versa).

O diferencial do seu projeto está no comportamento desse interruptor. Em vez de trocar o estado de apenas uma lâmpada em particular, ele intercala entre as lâmpadas em que ele vai agir, funcionando da seguinte maneira: na primeira vez ele troca o estado da lâmpada número 1; na segunda, e nas próximas vezes, ele troca o estado da lâmpada que está K posições à direita da anterior. Se ele chegar no final da sequência, ele continua a contar da posição 1.

Ou seja, seja $N = 8$ e $K = 3$, se pressionarmos o interruptor 4 vezes ele vai agir nas lâmpadas 1, 4, 7 e 2, respectivamente.

Você ficou um pouco intrigado com sua própria invenção, e resolveu fazer alguns testes. Dado o estado inicial de cada uma das N lâmpadas (ligado ou desligado), o valor de K e um número M de vezes que o interruptor foi pressionado, diga o estado final das N lâmpadas.

Entrada

Haverá diversos casos de teste. Cada caso de teste inicia com dois inteiros N e Q ($3 \leq N \leq 100$, $1 \leq Q \leq 1000$), representando o número de lâmpadas e o número de consultas, respectivamente.

Em seguida haverá uma linha com N caracteres, representando o estado das N lâmpadas. O i -ésimo caractere indica o estado da i -ésima lâmpada, estando ela ligada (caractere 'o') ou desligada (caractere 'x'), para todo $1 \leq i \leq N$.

Em seguida haverá Q linhas, cada uma com dois inteiros K e M ($1 \leq K < N$, $1 \leq M \leq 10^6$), indicando o tamanho do "salto" que seu interruptor dá (conforme o enunciado), e o número de vezes que o mesmo foi pressionado, respectivamente. A cada consulta o interruptor inicia na posição 1.

O último caso de teste é indicado quando $N = Q = 0$, o qual não deverá ser processado.

Saída

Para cada consulta imprima uma linha, contendo N caracteres cada, onde o i -ésimo caractere indica o estado da i -ésima lâmpada, estado ela ligada (caractere 'o') ou desligada (caractere 'x').

Exemplo de Entrada

```
8 3
xxxxxxxx
3 3
3 4
3 7
11 3
xoxoxxxxxx
5 6
10 43
9 1000
0 0
```

Exemplo de Saída

```
oxoxxxx
oxoxxxx
ooooxxx
ooooxxxxx
xxoxxxxxx
oxoxxxxxx
```

Apoio: <http://crbonilha.com/pt/>

Problema F

Diferença entre Datas

arquivo: datas.c , datas.cpp, datas.java

Joana está trabalhando na nova versão do blog de sua empresa. Uma das coisas que ela quer mudar é a forma como as datas são mostradas no blog. Na versão atual, as datas são mostradas como ano-mês-dia (por exemplo, 2014-05-23).

Ela deseja que, em vez disso, seja mostrado o número de dias que se passaram desde a publicação do post até hoje (por exemplo, se hoje é 9 de agosto de 2014, então a data 2014-05-23 seria mostrada como "78 dias atrás" e a data 2014-08-07 seria "2 dias atrás").

Joana está ocupada com alguns problemas mais complexos relacionados ao blog e pediu que você a ajudasse com essa parte. Dadas duas datas, calcule o número de dias que se passaram entre elas.

Observações

Tome cuidado com anos bissextos (que possuem o dia 29 de fevereiro, e, portanto, 366 dias no total). Um ano é bissexto se seu número é um múltiplo de 400 ou se é um múltiplo de 4, mas não de 100.

Entrada

A entrada começa com uma linha contendo um único inteiro N , que representa o número de casos de teste ($0 < N \leq 10000$).

Em seguida, há N linhas, cada uma descrevendo um caso de teste.

Cada uma dessas linhas possui duas datas separadas por um espaço.

As datas estão no formato AAAA-MM-DD, onde AAAA é o ano, MM é o mês, e DD é o dia. Você pode supor que todas as datas são válidas (i.e., não existem datas como 2013-02-31 na entrada). Todas as datas estão entre 1970 e 2014 (inclusive). Mês e dia são sempre dados com dois dígitos; 3 de fevereiro de 2014 é representado como 2014-02-03.

Saída

Para cada caso de teste, imprima uma linha contendo apenas o valor absoluto do número de dias entre as duas datas.

Exemplo de Entrada

```
4
2014-07-27 2014-07-25
2014-08-09 2014-05-23
2012-01-01 2013-01-01
1970-01-01 1970-01-01
```

Exemplo de Saída

```
2
78
366
0
```


Apoio: <http://crbonilha.com/pt/>

Problema G

Caminho seguro

arquivo: caminho.c , caminho.cpp, caminho.java

Você, um Tenente especializado em computação, foi designado para ajudar o Coronel Rambo que por ser italiano, prefere ser chamado de Ramboni. Ramboni é o bravo comandante das tropas aliadas, que luta para manter a ordem na região das Algasias.

Para executar as missões, as tropas precisam se alimentar bem e com regularidade. Para isso, diariamente um caminhão sai do quartel, na cidade Origem e viaja alguns quilômetros passando por várias cidades até chegar ao destino, na cidade Destino, onde tem comida farta. Entretanto, nos últimos dias, começaram a ocorrer ataques ao caminhão para roubar o carregamento.

Diante deste cenário crítico, o Coronel Ramboni elaborou um plano. O caminhão deveria ir por um caminho e voltar por outro caminho totalmente diferente do caminho de ida. Sendo que o caminhão não pode passar pela mesma rodovia/estrada duas vezes. Caso isto não seja possível, o caminhão deve ficar no destino para voltar apenas no outro dia. O incansável Coronel Ramboni pediu uma coisinha a mais: temos que ser rápidos, pois a tropa não pode ficar com fome.

Entrada

A entrada conterà vários casos de testes. Cada caso de teste iniciará com um inteiro n ($2 \leq n \leq 100$) indicando o número de cidades. Origem é a cidade de número 1, e Destino é a cidade n . A próxima linha conterà um inteiro m representando o número de estradas/rodovias. As próximas m linhas descreverão as m estradas/rodovias. Cada linha conterà 3 inteiros, ou seja, as duas cidades conectados por uma estrada/rodovia e o tempo necessário para percorrer a distância entre elas (em minutos). Nenhuma estrada/rodovia levará mais do que 1000m ou menos que 1m. Cada estrada/rodovia se conectará a duas diferentes cidades. Nenhum par de cidades será diretamente conectado por mais do que uma estrada/rodovia. O último caso de teste será seguido por uma linha contendo o número 0.

Saída

Para cada caso de teste, a saída deverá ser uma linha contendo um único inteiro – o número de minutos que o caminhão precisará para ir de Origem até Destino e voltar. (Considere que o tempo que o caminhão fique em Destino seja desprezível). Se não houver solução, escreva "Pernoite."

Exemplo de Entrada

```
2
1
1 2 999
3
3
1 3 10
2 1 20
3 2 50
9
12
1 2 10
1 3 10
1 4 10
2 5 10
3 5 10
4 5 10
5 7 10
6 7 10
7 8 10
6 9 10
7 9 10
8 9 10
0
```

Exemplo de Saída

```
Pernoite.
80
Pernoite.
```

Apoio: <http://crbonilha.com/pt/>

Problema H

Promoção

Arquivo fonte: promocao.c, promocao.cpp ou promocao.java

Dr Luis Cláudio, um sujeito antenado com as promoções oferecidas pelo supermercado VemQueTem, o qual fica próximo à sua residência, anda muito sorridente ultimamente. Descobriu-se que ele foi sorteado em uma promoção oferecida pelo supermercado. Nesta promoção, a pessoa poderia entrar no supermercado, sozinho, e levar todos os produtos que pudesse carregar. Porém, algumas regras foram estabelecidas.

- 1) Entrar sozinho
- 2) Apenas um produto de cada tipo pode ser levado
- 3) Uma lista L contendo os preços e pesos dos produtos deve ser seguida
- 4) Um peso P máximo foi estabelecido

Você foi contratado pelo vizinho curioso do Dr Luis Cláudio para descobrir qual o valor total em mercadorias que ele conseguiu levar para casa.

Entrada

A entrada consiste de T casos de testes. Cada caso de teste começa com um inteiro N ($1 \leq N \leq 100$) que indica o número de produtos da lista L. As N linhas seguintes são formadas por 2 inteiros p e P. O primeiro inteiro, p ($1 \leq p \leq 1000$), representa o preço do produto. O segundo inteiro P ($1 \leq P \leq 30$) representa o peso do produto. A próxima linha contém um inteiro M, que indica o peso máximo permitido. O fim da entrada é representado por um 0.

Saída

Para cada caso de teste imprima um inteiro que representa o total dos produtos que Dr Luis Cláudio conseguir levar para casa.

Exemplo de entrada

```
4
72 17
44 23
31 24
22 2
26
3
72 17
44 23
31 24
25
0
```

Exemplo de saída

```
94
72
```

Apoio: <http://crbonilha.com/pt/>

Problema I

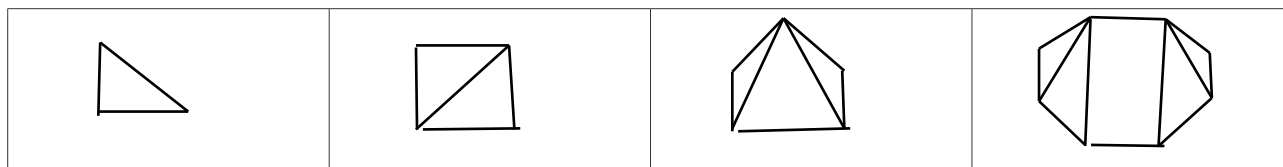
Triangulação de Delaunay

Arquivo fonte: delaunay.c, delaunay.cpp ou delaunay.java

Em matemática, uma Triangulação de Delaunay para um conjunto de pontos P no plano é uma triangulação DT(P) onde nenhum ponto em P está dentro da circunferência formada por qualquer triângulo na DT(P). A Triangulação de Delaunay maximiza o menor ângulo de todos os triângulos na triangulação; esta tende a evitar triângulos com ângulos internos muito pequenos.

A triangulação foi inventada por Boris Delaunay em 1934. Para um conjunto de pontos em uma mesma linha, não existe Triangulação de Delaunay (o conceito de triangulação é desfeito para este caso). Para quatro ou mais pontos em um mesmo círculo (isto é, os vértices de um retângulo) a Triangulação de Delaunay não é única: cada uma das duas possibilidades de triangulação que divide o quadrilátero em dois triângulos satisfaz a “condição Delaunay”, isto é, que as circunferências de todos os triângulos tenham interiores vazios. Considerando que as circunferências são esferas, a noção de Triangulação de Delaunay estende-se a três dimensões. Generalizações são possíveis para métricas diferentes das Euclidianas. Entretanto, nestes casos não se pode garantir a existência ou a unicidade da Triangulação de Delaunay.

O doutor Louco da Silva, em seu doutorado, resolveu verificar se a afirmativa sobre a triangulação de Delaunay anteriormente dita era verdadeira. Analisou as configurações de polígonos perfeito, como mostra a figura.



Ele verificou que é verdadeira e que a quantidade de arcos que criam a triangulação de Delaunay para a mesma quantidade de pontos era sempre a mesma. Por exemplo, para 3 pontos é sempre 3, para 4 pontos é sempre 5, para 5 pontos é sempre 7 e para 6 pontos é sempre 12 e assim sucessivamente.

Ele resolveu então criar um número real (X) determinado pela relação da quantidade de arco (I) com a quantidade de pontos (L) que é:

$$X = \frac{I - L}{L}$$

Ajude o doutor fazendo um programa que calcule o valor do número real X.

Entrada

A entrada é composta de um conjunto de teste, que contém uma única linha com um valor inteiro L ($3 \leq L \leq 10^{80}$). A entrada termina quando L = 0.

Saída

Para a entrada seu programa deve produzir um único resultado real X com precisão de seis casas decimais.

Exemplo de entrada

3
4
5
6
0

Exemplo de saída

0.000000
0.250000
0.400000
0.500000

Apoio: <http://crbonilha.com/pt/>

Problema J

Robocopia

Arquivo fonte: robocopia.c, robocopia.cpp ou robocopia.java

Robocopiadores são pequeninos drones que uma vez ativados copiam o movimento de rotação uns dos outros. Quando um drone é ativado junto com outros, eles trabalham em conjunto, como se fossem um só.

Recentemente Daniel comprou uma fábrica de robocopiadores. Um braço mecânico coloca cada robocopiador aleatoriamente em uma área, formando assim um conjunto de robocopiadores. Cada conjunto pode ser composto por números diferentes de robocopiadores. E para testá-los, eles são ativados. Os robocopiadores ativados devem passar por uma esteira para posteriormente serem desativados e armazenados. Vários conjuntos de robocopiadores podem passar pela mesma esteira. A largura da esteira deve ser sempre a menor possível, mas que comporte todos os conjuntos.

Como Daniel é um empresário inexperiente, não fez um planejamento adequado e então teve de contratar funcionários adicionais para verificar manualmente qual o tamanho da esteira que ele tem de configurar para suportar os diferentes conjuntos de robocopiadores. E claro, este processo é muito custoso e demorado.

Para diminuir os gastos e aumentar a eficiência, Daniel contratou você para calcular, de maneira automática, qual a menor largura da esteira para que todos os conjuntos de robocopiadores possam ser armazenados corretamente.

Fig 1

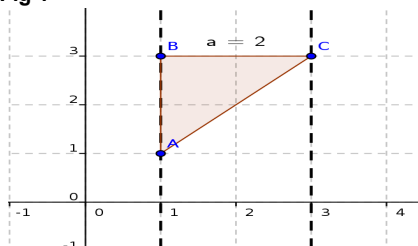
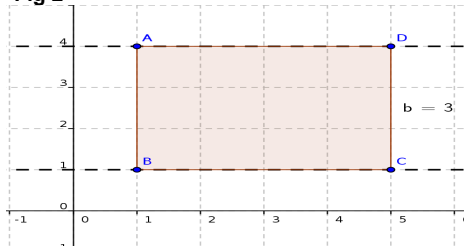


Fig 2



Na Figura 1, por exemplo, a máquina ativou 3 robocopiadores (A,B e C) e a menor distância é $a = 2$, entre BC. Quando a máquina fizer o outro conjunto de robocopiadores (A,B,C e D) da Figura 2, a menor distância é AB ou DC, $b = 3$, e neste caso, o conjunto tem de ser rotacionado 90 graus para passar na esteira, que tem tamanho 3. Logo, se estes conjuntos fossem passar pela esteira, esta teria que ter uma largura mínima de 3.

Entrada

A entrada é composta por vários casos de testes.

A primeira linha é composta de um inteiro N ($1 \leq N \leq 10000$) que representa o número de casos de testes.

Cada caso de teste é composto por um número inteiro C ($1 \leq C \leq 100$) indicando o número de conjuntos de robocopiadores fabricados. Cada conjunto é composto por um número inteiro c ($1 \leq c \leq 10000$) representando o número de robocopiadores do conjunto, seguido de c linhas de números inteiros, indicando a coordenada $-100000 \leq (x, y) \leq 100000$ de cada robocopiador do conjunto.

Saída

Em cada linha deverá ser impresso o tamanho da menor esteira para produzir todos os conjuntos de robocopiadores, com precisão de 10 casas decimais.

Exemplo de entrada

```
1
2
4
1 4
1 1
5 1
5 4
3
1 1
1 3
3 3
```

Exemplo de saída

```
3.0000000000
```

Apoio: <http://crbonilha.com/pt/>

Problema K

E agora Mamá ?

Arquivo fonte: mama.c, mama.cpp ou mama.java

Na velha China, o mestre ThaiTchunCamon, era conhecido pelos súditos apenas como Camon. Camon tinha um conselheiro muito inteligente e admirado por toda a região, seu nome era Mamá.

Camon convocou Mama para resolver uma questão: preciso saber de quantas maneiras diferentes eu consigo subir as escadarias da muralha da China, só que, indo de 1 em 1 de degraus, ou de 2 em 2. Para ajudar, mestre Camon disse brilhantemente que, para subir até o primeiro degrau tem somente uma possibilidade, e para chegar ao segundo degrau, tem duas possibilidades.

Você deve ajudar Mamá a resolver esta questão.

Entrada

A entrada é composta por vários casos de testes.

A primeira linha é composta de um inteiro N ($N \leq 10000$) que representa o número de casos de testes.

Cada caso de teste é composto por um número inteiro C ($3 \leq C \leq 10000$) indicando o número de degraus a serem escalados.

Saída

Em cada linha deverá ser impresso o número de possibilidades de subir a escadaria.

Exemplo de entrada

2
3
5

Exemplo de saída

3
8