



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

3er Concurso de Programación Competitiva MAC 2015

Indicaciones

- La entrada del programa se debe leer de la entrada estándar `stdin`.
- La salida de programa se debe imprimir a la salida estándar `stdout`.
- No imprimir cadenas como: “Dame la entrada”, “La salida es:”
- No usar librerías no estándar

Ejemplo:

```
#include <conio.h> /*Esta libreria no es estandar*/

int main()
{
    system("pause");//No usar esto
    getch();//Ni esto

    return 0;
}
```

- Para los programas hechos en Java el nombre de la clase será la letra del problema en mayúscula seguido de la palabra “main” por ejemplo `Amain.java`.



Facultad de Estudios Superiores
Acatlán
Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema Ejemplo

Límite de tiempo: 1 segundos

Problema

Dados 2 enteros A y B debes imprimir la suma de ellos.

Entrada

Una línea con un entero T que indica el número de casos. Las siguientes T líneas aparecerán 2 enteros A y B .

Salida

T líneas con la suma que se pide.

Entrada Ejemplo

```
2
2 3
5 8
```

Salida Ejemplo

```
5
13
```

Solución

Los siguientes códigos resuelven el problema anterior:

■ C

```
#include <stdio.h>

int main()
{
    int casos,a,b;

    scanf("%d",&casos);

    while(casos--)
    {
        scanf("%d %d",&a,&b);
        printf("%d\n",a+b);
    }

    return 0;
}
```

■ C++

```
#include <iostream>
using namespace std;

int main()
{
    int casos,a,b;

    cin >> casos;

    while(casos--)
    {
        cin >> a >> b;
        cout << a+b << endl;
    }

    return 0;
}
```

■ Java

```
import java.util.*;

class Bmain{//Ejemplo del nombre de la clase
//para el problema B

    public static void main(String []args)
    {
        Scanner sc = new Scanner(System.in);

        int casos,a,b;

        casos = sc.nextInt();

        while(casos-- > 0)
        {
            a = sc.nextInt();
            b = sc.nextInt();

            System.out.println(a+b);
        }
    }
}
```



Problema A - Armando elecciones

Límite de tiempo: 5 segundos

Problema

En Maclandia se llevarán a cabo elecciones presidenciales muy pronto, pero dado que conllevan un costo muy elevado, su Instituto Electoral te ha encargado que les digas si es posible determinar quién ganará las elecciones presidenciales, ya que si esto es posible, les ahorrarías mucho dinero.

En Maclandia ocurren muchas situaciones que no ocurren en los demás países de su planeta Namek, por ejemplo: debido a ciertas influencias (dinero, dinero, dinero, nepotismo, etc.), el voto de algunas personas vale más que el de otras; además, si una persona vota por cierto candidato todos sus amigos votarán por el mismo candidato; aquellas personas que no tengan amigos con un voto definido, el día de las elecciones votarán por algún candidato según su sano juicio.

Las personas en Maclandia tienen agregados a todos sus amigos en su red social, y si una persona es amiga de otra, entonces esta otra también es amiga de esa persona. Es bien sabido que los candidatos votarán por ellos mismos.

Sin embargo en este país existen dos genios: uno malvado y uno bueno. Si alguien es amigo del genio malvado entonces el candidato por el que votaría perderá las elecciones, sin importar el valor de los votos que haya obtenido. En caso contrario si alguien es amigo del genio bueno, el candidato por el que votaría ganará las elecciones.

Si los genios intentan hacer que el mismo candidato gane y pierda las elecciones entonces se anularán sus poderes y no tendrán poder sobre las elecciones. Si algún genio no tiene amigos, entonces este no podrá ejercer su poder para influir en las elecciones. Los genios sólo pueden ser amigos (directa o indirectamente) de personas que ya tienen predeterminado su voto. Ninguna persona votará por dos candidatos, puesto que ninguna persona será amiga indirecta de dos candidatos al mismo tiempo. Un genio es considerado como una persona en sus relaciones de amistad.

Entrada

La primera línea será un entero $1 \leq T \leq 100$, indicando la cantidad de casos de prueba. Cada caso de prueba se compondrá de varias líneas: la primera tendrá tres números n, c y r ($2 \leq n \leq c \leq 500$, $0 \leq r \leq (n + 2 - c)^2$) indicando el número de votantes, el número de candidatos y el número de relaciones de amistad respectivamente. Los votantes son identificados con los números 1 hasta n inclusive. La siguiente línea tendrá n números enteros ($1 \leq a_j \leq 10000$), el j -ésimo número indicará la cantidad de puntos que vale el voto de la j -ésima persona. En la siguiente línea vendrán c números, los cuales indican el número de cada candidato. r líneas vendrán después, cada una con dos números $-1 \leq a, b \leq n$, indicando que las personas a y b son amigas. La persona 0 es el genio bueno, y la persona -1 es el genio malvado. Los casos de prueba están separados por una línea en blanco.

Salida

Para cada caso, indica en una línea distinta: el número del candidato que ganará las elecciones, si es posible determinar quien las ganara; si no es posible determinar esto, imprime “Es mas facil salir de la friendzone que saber quien ganara” sin las comillas.

Entrada Ejemplo

```
2
4 3 3
1 10 30 1
1 2 3
1 4
2 0
3 -1

5 3 2
1 50 10 2 20
1 2 3
1 4
2 -1
```

Salida Ejemplo

```
2
Es mas facil salir de la friendzone que saber quien ganara
```

Explicación de la entrada ejemplo

En el segundo caso, la persona 2 pierde las elecciones automaticamente porque es amiga del genio malvado, la persona 1 puede juntar 3 puntos resultado de la suma de sus puntos y los de sus amistades, la persona 3 puede juntar 10 puntos. Sin embargo no es posible determinar quien ganará la elección, a pesar de que la persona 3 tiene más puntos, porque la persona 5, que no es amiga directa/indirecta de algún candidato, cuando vote puede hacerlo por alguno de los dos (1 ó 3) inclinando la balanza a favor de cualquiera.

Silverio Flores Moroni - Grupo de Algoritmia



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema B - Buscando amigos

Límite de tiempo: 3 segundos

Problema

Después de una intensa jornada de actividades en la semana de MAC, algunos alumnos han encontrado a quien consideran “el amor de su vida”. Sin embargo muchos estudiantes de MAC se consideran a sí mismos matemáticos puros y no le hablan a cualquier persona. Los alumnos que se inscribieron a la semana de MAC recibieron un id (numero de identificación) para poder entrar a las conferencias. Los matemáticos puros sólo le hablarán a la persona cuyo id a sea amigo de su id b . Un número a es amigo de b si la suma de los divisores de a (exceptuando sólo a a) es igual a b , nota que si a es amigo de b , b no necesariamente es amigo de a . Algunos alumnos de MAC han recurrido a ti debido a tus grandes conocimientos en estas áreas del saber para saber si “el amor de su vida” les hablará. Por causas fortuitas todos estos amores son matemáticos puros. Tu tarea es determinar si los alumnos tienen oportunidad de hablar con quienes consideran el amor de su vida.

Entrada

La primer línea de la entrada contendrá un entero $T \leq 200$, el número de casos de prueba, a continuación se darán T renglones, cada uno con un caso de prueba. Cada renglón tendrá dos números a, b , tales que $1 \leq a, b \leq 10000$

Salida

Para cada caso de prueba imprime “Quiza y sin quiza”, sin las comillas si a es amigo de b ; imprime “Llora desconsoladamente” en caso contrario.

Entrada Ejemplo

```
3
6 5
3 10
1184 1210
```

Salida Ejemplo

```
Llora desconsoladamente
Llora desconsoladamente
Quiza y sin quiza
```



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema C - Caballo Sin Mecate

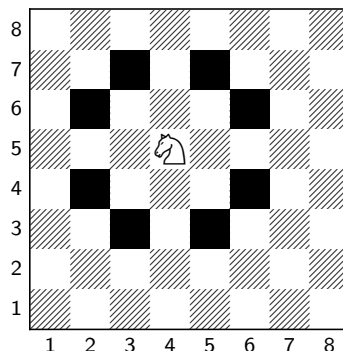
Límite de tiempo: 1 segundos

Problema

Últimamente los integrantes del Grupo de Algoritmia se han obsesionado con el milenario juego de ajedrez, pero tienen un inconveniente, puesto que Moro siempre les gana, han decidido inventar un nuevo juego al que llamaron “Caballo sin Mecate”.

El juego consiste en un tablero de ajedrez de tamaño $n \times m$ en el que interactúan dos jugadores. Dada una configuración de piezas tienes que llegar en la menor cantidad de pasos con el caballo de un punto a otro del tablero; gana el jugador que pueda dar la menor cantidad de movimientos. Además, se estipula que en el tablero sólo hay un caballo y múltiples piezas de otros tipos de las cuales no puede ocupar su lugar, pero sí saltarlas.

El caballo no se desplaza en línea recta, sino que tiene un movimiento característico llamado salto. El salto del caballo se parece a una L, y se compone de un desplazamiento de dos casillas en dirección horizontal o vertical, y otra casilla más en ángulo recto (en el diagrama siguiente, los cuadros marcados son los posibles lugares hacia donde puede desplazarse el caballo).



Tu tarea es ayudarnos a encontrar el número mínimo de movimientos que le tomará al caballo llegar a cierto punto, dada una configuración, para así poderle ganar a Moro.

Entrada

La primer línea contendrá un solo número T ($1 \leq T \leq 100$) que representa la cantidad de casos de prueba, que se seguirá de T casos de prueba. La primer línea de cada caso contendrá dos números enteros n y m ($1 \leq n, m \leq 100$). Seguirán n líneas, una línea por cada fila del tablero, donde cada línea tendrá m caracteres c_{ij} , que representan el estado del escaque¹ de la i -ésima fila y la j -ésima columna (numerados de abajo hacia arriba y de izquierda a derecha como se muestra en el diagrama anterior). Donde c_{ij} puede ser una casilla vacía (*), un peón ♙(p), una torre ♖(t), un alfil ♗(b), un rey ♔(k), una reina ♕(q) o un único caballo ♞(c) para todo el tablero. La última línea de cada caso tendrá dos enteros x, y ($1 \leq x \leq n, 1 \leq y \leq m$) que representan la fila y columna a las que quieres llegar.

Salida

Por cada caso deberás imprimir una sola línea que contendrá un entero que es la cantidad mínima de movimientos del caballo para llegar a su destino o -1 en caso de que sea imposible llegar a esa casilla.

Entrada Ejemplo

```
2
7 8
*****
*****
*****
*****
*****
bbbqkptt
*c*****
1 6
2 2
**
c*
2 2
```

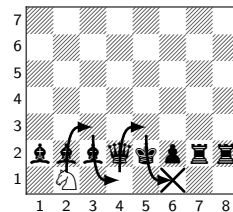
Salida Ejemplo

```
4
-1
```

Notas

En el primer caso se puede llegar de la siguiente manera:

En el segundo caso no hay manera de llegar a la casilla (2, 2).



¹Cuadrado del tablero



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema D - Doncella Jean

Límite de tiempo: 1 segundos



Problema

En un lejano reino, existe una Doncella llamada Jean, es una chica muy bella pero también muy peculiar. Con lo anterior nos referimos a que tiene costumbres poco casuales. Para resumir, ella cuenta con muchos pretendientes en los reinos vecinos, incluyendo el suyo. Algunos de ellos son príncipes, duques y algunos otros con buen estatus social. No obstante hay un plebeyo que está enamorado de ella, para su fortuna (o no) es un “friend” de nuestra querida doncella, y como se imaginarán ambos tienen una amistad por Facebook.

Nuestro desdichado plebeyo (llamado Anselmo) últimamente ha tenido charlas muy frecuentes con la Doncella, él siente que varios de los mensajes de ella son indirectas pero no está muy seguro. Para cerciorarse de ello ha seleccionado N mensajes y los ha marcado como “Le gusto” y “No le gusto”. Él piensa que si la proporción de “Le gusto” es significativamente mayor a la de “No le gusto” entonces decidirá si está en la friendzone o no. Para no complicarse define que si la proporción de “No le gusto” es $\geq 0,2$ entonces se encuentra en la friendzone.

Ayuda a Anselmo a determinar si se encuentra en la friendzone o no dados los N mensajes de las últimas conversaciones.

Entrada

La primer línea es un número C que denota la cantidad de casos de entrada. En los siguientes C bloques aparecerán un número $2 \leq N \leq 10000$ y N mensajes. Los mensajes serán “si” y “no” para denotar “Le gusto” y “No le gusto” respectivamente.

Salida

Para cada caso deberás imprimir como respuesta “friendzone” cuando $r \geq 0.2$ donde r denota la proporción de “No le gusto”, en caso contrario la carita feliz “:)”.

Entrada Ejemplo

3

4

si

si

no

si

5

si

si

si

si

si

8

si

no

si

si

si

si

si

si

Salida Ejemplo

friendzone

:)

:)

Kenny Yahir Méndez Ramírez - Grupo de Algoritmia



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema E - Explanada

Límite de tiempo: 1 segundos

Problema

Uno de tus amigos ha decidido visitarte en la escuela, sin embargo él nunca ha venido a la FES y no sabe cómo encontrarte, por lo tanto has decidido ayudarlo para que llegue a la explanada de MAC. La explanada la podemos ver como un polígono convexo² en el plano, y tú sabes muy bien las coordenadas de todos sus vértices, pues fue lo primero que te enseñaron en PIMAC. Para poder darle indicaciones a tu amigo necesitas saber si ya ha llegado a la explanada, así que le has pedido que te mande sus coordenadas. Ahora sólo necesitas hacer un programa que te diga si tu amigo está dentro de la explanada o no.

Entrada

La primera línea contendrá un entero $1 \leq t \leq 100$, el número de casos de prueba. La primera línea de cada caso de prueba será un entero $3 \leq n \leq 100$ que denota el número de vértices del polígono que representa a la explanada. En las siguientes n líneas de cada caso de prueba habrá dos enteros $0 \leq x_i, y_i \leq 100$ representando las coordenadas del i -ésimo vértice del polígono. Los puntos estarán dados en orden de tal forma que forme un polígono convexo (puede ir en el sentido de las manecillas del reloj o en sentido contrario a las manecillas del reloj). La última línea de cada caso de prueba serán dos enteros $0 \leq x_a, y_a \leq 100$, las coordenadas de tu amigo. El punto (x_a, y_a) no va a estar nunca sobre el perímetro del polígono.

Salida

Para cada caso de prueba deberás imprimir “si” (sin comillas) si tu amigo está dentro de la explanada, o “no” en caso contrario

²Un polígono convexo es una figura en la que todos sus ángulos interiores miden menos de 180 grados o π radianes.

Entrada Ejemplo

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

Salida Ejemplo

```
no
si
```

David Felipe Castillo Velázquez - Grupo de Algoritmia



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema F - Fiesta

Límite de tiempo: 1 segundos

Problema

Después del intenso concurso de programación de la XIV Semana de MAC tu equipo piensa dar una gran fiesta en tu casa para celebrar su aplastante triunfo sobre los demás equipos.

El problema es que para entonces tu cerebro ya no podrá trabajar muy bien, puesto que lo exprimiste al máximo en el concurso, y necesitas saber cuál es el valor esperado de la cantidad de personas que asistirán a tu fiesta, por lo que desde ahora harás un programa que te ayude en esta tarea.

Lamentablemente es probable que no todas las persona asistan a tu fiesta, porque algunas no resolvieron muchos problemas y no se sienten de ánimo.

Se han invitado n personas a tu fiesta y cada persona tiene una probabilidad p_i ($i = 1, 2, \dots, n$) de asistir, que está dada en función de su desempeño, es decir,

$$p_i = \sum_{j=1}^m \frac{r_{ij}}{m}$$

, con

$$r_{ij} = \begin{cases} \frac{1}{a_{ij}} & \text{si } a_{ij} \neq 0 \\ 0 & \text{si } a_{ij} = 0 \end{cases}$$

, donde a_{ij} es igual a la cantidad de intentos que le tomó a la persona i resolver el problema j , $a_{ij} = 0$ en caso de que la i -ésima persona no haya resuelto el j -ésimo problema, ($j = 1, 2, \dots, m$), y m es la cantidad de problemas en total .

Entrada

La primer línea contendrá un solo número T ($1 \leq T \leq 100$) que representa la cantidad de casos de prueba.

La primer línea de cada caso contendrá dos números enteros n y m ($1 \leq n \leq 100$, $1 \leq m \leq 11$).

Seguirán n líneas, una línea por cada persona, donde la i -ésima línea tendrá m enteros a_{ij} ($0 \leq a_{ij} \leq 100$) separados por un espacio.

Salida

Por cada caso deberás imprimir una sola línea que contendrá el valor esperado de personas en tu fiesta redondeado a 2 decimales.

Entrada Ejemplo

```
2
5 3
1 0 1
1 1 1
0 19 2
11 0 2
3 0 7
4 7
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
```

Salida Ejemplo

```
2.21
4.00
```

Notas

En el primer caso la primera persona pudo resolver el primer y tercer problema en un intento, mientras que la segunda resolvió todos, por lo que sus probabilidades de ir son 0,66 y 1 respectivamente.

En el segundo caso todos resolvieron todos los problemas en un solo intento, por lo que se espera que todos asistan.



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema G - Guíando a Edgar

Límite de tiempo: 1 segundos

Problema

Edgar es una persona a la que le gusta optimizar su tiempo, por lo tanto cada que se tiene que desplazar de un lugar a otro dentro de la FES, le gusta irse por el camino que minimice el tiempo que le toma desplazarse. Además a él le gusta siempre pasar por lugares nuevos, porque asegura que eso ayuda a crear nuevas conexiones neuronales, entonces cada que planea la trayectoria para llegar de un lugar a otro, él considera un camino en específico por el que debe pasar obligatoriamente.

En este momento Edgar está demasiado ocupado encargándose del concurso, así que te ha pedido que le ayudes a organizar sus siguientes Q rutas para no tener que perder el tiempo.

Dado que la FES es muy grande, nos vamos a concentrar sólo en los N lugares favoritos de Edgar, numerados del 1 al N , con caminos entre cada uno de los N lugares. Los caminos entre cada par de lugares i, j , son bidireccionales, es decir, Edgar puede recorrer el camino de i a j o de j a i .

Entrada

La primera línea tendrá un entero $1 \leq T \leq 100$, el número de casos de prueba. La primera línea de cada caso de prueba será un entero $2 \leq N \leq 50$, los lugares favoritos en la FES de Edgar, seguido vendrán N líneas, con N enteros cada una. El j -ésimo valor de la i -ésima línea, $1 \leq w_{ij} \leq 1000$, indicará el tiempo que le toma a Edgar recorrer el camino del lugar i al lugar j ($w_{ij} = w_{ji}$). La $N + 2$ línea de cada caso de prueba será un entero $1 \leq Q \leq 100$, el número de rutas que tienes que planear para Edgar, seguido de Q líneas, cada una con cuatro enteros $1 \leq a, b, c, d \leq N$, indicando que Edgar quiere ir del lugar a al lugar b pasando obligatoriamente por el camino entre c y d (en cualquier sentido), $a \neq b, c \neq d$.

Salida

Para cada caso de prueba deberás imprimir Q enteros, representando el tiempo que el toma a Edgar realizar cada una de las rutas en el mismo orden en que fueron solicitadas.

Entrada Ejemplo

```
1
5
0 9 5 2 1
9 0 5 2 2
5 5 0 2 6
2 2 2 0 7
1 2 6 7 0
2
5 1 5 2
5 1 2 3
```

Salida Ejemplo

```
5
11
```

Explicación de la entrada ejemplo

Para la primera ruta, Edgar puede tomar la siguiente sucesión de caminos: 5-2-5-1, con un tiempo total de $2+2+1=5$.

Para la segunda ruta, Edgar puede tomar la siguiente sucesión de caminos: 5-2-3-4-1, con un tiempo total de $2+5+2+2=11$.

David Felipe Castillo Velázquez - Grupo de Algoritmia



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema H - Hurgando en el CEDETEC

Límite de tiempo: 1 segundos

Problema

Por fortuna el CEDETEC acaba de recibir presupuesto ilimitado y ha aumentado el número de cubículos desde $-\infty$ a ∞ . El grupo de Algoritmia ha aprovechado la expansión del CEDETEC y se mudó de cubículo.

Lamentablemente Kenny no se enteró de la mudanza y ahora no sabe en qué cubículo quedó el grupo de Algoritmia, pero como es muy importante que lo encuentre decidió ir buscando de cubículo a cubículo, sin embargo resulta que Kenny quiere ir preguntando de una manera un poco peculiar: puede empezar a buscar en cualquier cubículo, una vez que haya buscado en el cubículo n sólo puede buscar en el cubículo $n \pm a$ o $n \pm b$.

Tu tarea consiste en saber si Kenny podrá algún día encontrar el cubículo de Algoritmia, sin importar a qué cubículo se mudó, conociendo los valores a y b que escogió Kenny.

Entrada

La primera línea será un entero $1 \leq T \leq 100$, indicando los números de casos de prueba. Para cada caso de prueba habrá una única línea con dos enteros $1 \leq a, b \leq 10^8$, explicados anteriormente.

Salida

Para cada caso de prueba debes imprimir “si” (sin las comillas) si Kenny podrá encontrar el cubículo de Algoritmia o “no” (sin las comillas), en caso contrario

Entrada Ejemplo

2
3 4
2 4

Salida Ejemplo

si
no