



**Concurso de Programación
Clasificatorio URJC - AdaByron 2022**
<http://www.ada-byron.es>

Cuadernillo de problemas



Realizado en la **Escuela Técnica Superior de Ingeniería Informática (URJC)**
11 de febrero de 2022

In almost every computation a great variety of arrangements for the succession of the processes is possible, and various considerations must influence the selections amongst them for the purposes of a calculating engine. One essential object is to choose that arrangement which shall tend to reduce to a minimum the time necessary for completing the calculation.

Ada Byron

Índice

A La suma de un cuadrado	3
B Cadena de favores	5
C La infractuosa búsqueda de un despacho	7
D Dickie y las pizzas matemáticas	9
E Siendo rico gracias a la bolsa	11

Autores de los problemas:

- Dixon David Morán González (ThoughtWorks)
- Isaac Lozano Osorio (Universidad Rey Juan Carlos)
- Leonardo Santella (Amazon)
- Sergio Cavero Díaz (Universidad Rey Juan Carlos)
- Sergio Pérez Peló (Universidad Rey Juan Carlos)

Tiempo: 1 segundo



La suma de un cuadrado

Sergio es un ávido matemático en las tierras de *Matelandia*, sus proezas le han conllevado a tener una excelente reputación entre los suyos. Ahora, Sergio nos pide ayuda para que le proporcionemos respuesta a un problema que intenta demostrar matemáticamente.

El problema en sí es sencillo: dada una matriz de enteros $N \times N$, se tiene que conseguir colocar en la submatriz (de tamaño $\frac{N}{2} \times \frac{N}{2}$) de la esquina superior izquierda, elementos de la matriz tal que la suma de los mismos sea máxima siguiendo las siguientes reglas:

- Puedes invertir todos los elementos de una fila r cualquiera de la matriz.
- Puedes invertir todos los elementos de una columna c cualquiera de la matriz.
- Puedes aplicar las reglas anteriores todas las veces que se quiera.

Llamamos inversión a colocar los elementos en orden del revés, e.g. Si en una fila tenemos los elementos [3, 4, 5]. Una vez invertidos quedarían como [5, 4, 3]. A Sergio solo le interesa el número que saldría de hacer estas operaciones. De lo demás, ya se encargará él mismo.

Entrada

La entrada constará de un entero T denotando los casos de prueba. Cada caso de prueba contiene el valor de la matriz, por lo tanto se recibirá un número N que determina el tamaño de la matriz. Las siguientes N líneas contendrán los valores de cada fila, por lo que cada línea tendrá N números separados por un espacio.

Salida

Para cada caso de prueba, la salida del problema debe ser un solo número que represente la suma máxima de todos los elementos de la submatriz. Esta submatriz está localizada en la esquina superior izquierda y es el resultado de aplicar las operaciones explicadas anteriormente.

Entrada de ejemplo

```
2
2
4 3
2 1
4
11 4 8 11
5 12 5 4
1 7 10 4
6 9 11 10
```

Salida de ejemplo

```
4
39
```

Límites

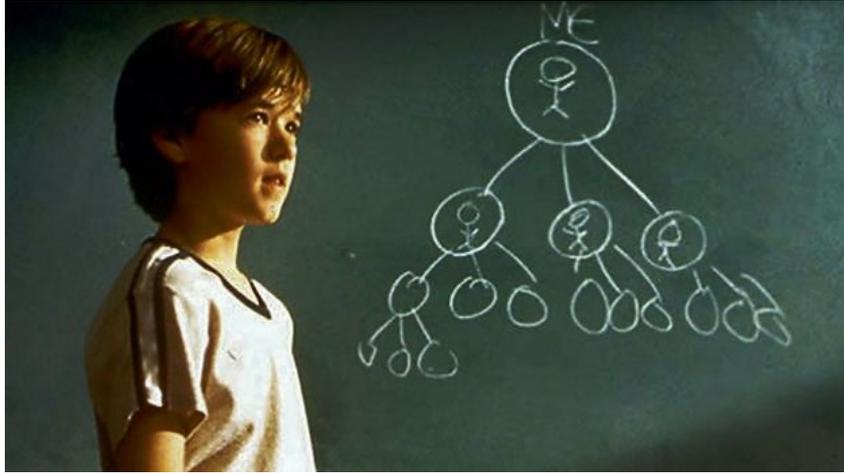
- N es par
- $2 \leq N \leq 500$

Tiempo: 1 segundo



Cadena de favores

En la película *Cadena de favores*, Trevor McKinney nos enseña cómo podemos hacer del mundo un lugar mejor haciendo un favor a tres personas conocidas. Si esas tres personas conocidas hacen un favor a otras tres personas, al final todo el mundo será ayudado.



El problema es que llevar la teoría a la práctica es, a veces, bastante complicado, especialmente cuando no todas las personas que reciben un favor cumplen con la norma de hacérselo a otras tres. Por ello, queremos construir un programa que nos ayude a saber si, dada la red de personas conectadas por los favores que se han hecho, somos capaces de conseguir que todas las personas de la red puedan recibir un favor de cualquier otra, empezando la cadena desde cualquiera persona y aunque sea necesario pasar por intermediarios. Hay que tener en cuenta que hay personas que inician su propia cadena de favores, haciendo un flaco favor al plan de llegar a todo el mundo y generando una cadena local independiente del resto, haciendo que no se cumpla la condición.

Entrada

La entrada comenzará con dos números enteros N y M que indicarán, respectivamente, el número de personas que forman parte de la cadena de favores y cuántos favores se han hecho en total en la red. A continuación, vendrán M líneas con dos números enteros O y D que indicarán que la persona O ha hecho un favor a la persona D . Es importante tener en cuenta que un favor no siempre es devuelto, por lo que la aparición de la línea $O D$ no implica que tengamos obligatoriamente la línea $D O$.

Salida

La salida mostrará por pantalla la palabra **SI**, en caso de que todas las personas de la red pueden recibir un favor de cualquier otra, aunque sea necesario pasar por intermediarios y la cadena **NO** en caso contrario.

Entrada de ejemplo

```
4 5
0 1
1 2
2 3
3 1
2 0
```

Salida de ejemplo

```
SI
```

Entrada de ejemplo

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

Salida de ejemplo

```
NO
```

Límites

- $1 \leq N \leq 400$
- $1 \leq M \leq 6000$
- $0 \leq O < N$
- $0 \leq D < N$

Tiempo: 1 segundo



La infructuosa búsqueda de un despacho

Otro día más, Alex Cremento, sigue esperando para que el director de la escuela de la universidad le ofrezca un despacho en condiciones. Y es que, aunque parezca mentira, después de 3 años de servicio, el profesor Cremento sigue sin poder atender a sus alumnos como estos se merecen.



Cierto es que al profesor Cremento le han ofrecido varios despachos y que él los ha rechazado. ¿Por qué? Porque prefiere no tener despacho a estar mal acompañado.

Puede que hoy sea el día en el que por fin, Alex Cremento, consiga un despacho, o quizás no...

Entrada

La entrada consiste en múltiples casos de prueba. La primera línea contiene un único número entero C que representa el número de casos de prueba. A continuación, se describe el contenido de cada caso de prueba.

La primera línea de cada caso de prueba contiene un único número entero N , que hace referencia al número de despachos que hay en la universidad.

Las siguientes N líneas contienen una serie de números enteros. El primero de ellos, D , hace referencia al identificador del despacho y es único: no se puede repetir en un mismo caso de prueba. El segundo de ellos, P hace referencia al número de puestos (mesas) que hay en ese despacho. Tras los dos primeros números (D y P) de la línea, hay P números enteros: $\{P_1, P_2, \dots, P_P\}$ que representan si el puesto está disponible, y en caso de que esté ocupado, representa la afinidad del profesor Cremento con esta persona.

Si el número es 0, quiere decir que ese puesto está disponible y por lo tanto, el profesor Cremento podría ocupar ese despacho. Si el número es mayor que 0, quiere decir que ese puesto está ocupado por otro profesor con el profesor Cremento se lleva bien. Por último, si el número es menor que 0, quiere decir que el profesor Cremento, se lleva mal con el profesor que ocupa ese puesto.

Salida

Para cada caso de prueba se imprime un solo entero: el identificador del despacho cuya suma de los números enteros $\{P_1, P_2, \dots, P_P\}$ sea la más alta, siempre y cuando: la suma sea mayor que 0 y haya al menos un valor $P_i \in \{P_1, P_2, \dots, P_P\}$ tal que $P_i = 0$, es decir, que al menos haya un hueco libre en el despacho. En el caso de que la suma más alta sea menor o igual a 0, o bien, no haya ningún despacho con puestos libres, se deberá imprimir el siguiente texto: "No hay despachos disponibles por ahora, quizás otro día."

En el caso de que haya dos o más despachos con huecos libres y cuya suma sea igual (mismo valor), se devolverá el identificador del primer despacho leído de la entrada.

Entrada de ejemplo

```
3
3
1 4 -3 4 3 2
2 1 -2
3 3 3 -4 0
2
1 4 2 2 1 4
2 3 -2 -1 0
4
1 2 3 2
2 3 0 -3 5
4 2 1 0
9 5 2 -2 -1 -6 -3
```

Salida de ejemplo

```
No hay despachos disponibles por ahora, quizas otro dia.
No hay despachos disponibles por ahora, quizas otro dia.
2
```

Límites

- $1 \leq C, N, P, D \leq 10000$
- $-10 \leq P_i \leq 10$

Tiempo: 1 segundo



Dickie y las pizzas matemáticas

Dickie está de vacaciones y como todos saben, Dickie es un matemático entusiasta y un amante de las pizzas. Tal día como hoy, Dickie tiene pensado asistir a su restaurante de pizzas favorito.

Dickie quiere personalizar su propia pizza y en este restaurante hay N ingredientes distintos. Además, de cada ingrediente se conoce el **sabor** (s_i) y las **calorías** (c_i).

Como Dickie acaba de empezar sus vacaciones y durante las clases, no come mucho, su estómago tiene una **capacidad** C .

Además, hoy Dickie se levantó pensando en 2 funciones que le parecen muy interesantes:

- $h(x) =$ “Número de secciones creadas por x líneas rectas de pendientes diferentes en un plano donde un punto de corte nunca es atravesado por más de dos rectas módulo 1009”.
- $w(x) =$ “Número de expresiones que contienen x pares de paréntesis correctamente colocados módulo 1009”.

Supongamos que el conjunto de ingredientes escogidos es P (donde P contiene enteros $1 \leq j \leq N$ que denotan los ingredientes). La **felicidad** de Dickie al comerse una pizza viene dada por los ingredientes que hay en ella y se representa a través de la siguiente suma:

$$\sum_{x \in P} h(s_x)$$

El **peso** de la pizza P está representado por la siguiente suma:

$$\sum_{x \in P} w(c_x)$$

El peso de la pizza deberá ser menor, o igual, que la capacidad del estómago de Dickie. Esto se traduce en que $\sum_{x \in P} w(c_x) \leq C$.

Lo anterior puede ser descrito como que la **felicidad total** está definida por la suma de la felicidad individual que le aporta cada ingrediente a Dickie. Similarmente, la **capacidad del estómago de Dickie** debe ser **mayor o igual** que la suma del peso que aporta cada ingrediente de la pizza.

Tu tarea es ayudar a Dickie a encontrar la **máxima felicidad que puede alcanzar**.

Recuerda que:

- Expresión de paréntesis correctamente colocados son de esta manera: $(())$, $(())$, $()(())$. Las siguientes expresiones **no** son expresiones de paréntesis correctamente colocados: $)())$, $)(($, $)()$.

Entrada

La primera línea de la entrada serán 2 números enteros N y C ($1 \leq N \leq 1000$ y $1 \leq C \leq 1000$) que representan el número de ingredientes disponibles y la capacidad del estómago de Dickie, respectivamente.

La segunda línea contendrá N enteros, s_i , que representan el sabor del ingrediente i -ésimo ($1 \leq s_i \leq 10^{15}$).

La tercera línea contendrá N enteros, $1 \leq c_i \leq 1000$ donde c_i , correspondientes a las calorías del ingrediente i -ésimo.

Salida

La salida debe contener un solo número entero que represente la felicidad máxima posible.

Entrada de ejemplo

```
3 6
1 2 3
1 2 3
```

Salida de ejemplo

```
9
```

Entrada de ejemplo

```
3 10
1 2 3
1 2 3
```

Salida de ejemplo

```
13
```

Límites

- $1 \leq N \leq 1000$
- $1 \leq C \leq 1000$
- $1 \leq s_i \leq 10^{15}$
- $1 \leq c_i \leq 1000$

Tiempo: 1 segundo



Siendo rico gracias a la bolsa

En internet hay demasiados gurús de las criptomonedas y la bolsa. Todos ellos te venden un método que sí o sí te va a hacer rico. Pero para llegar a ese conocimiento, tienes que comprar su curso, ¿qué necesidad de vender cursos si ese método te hace rico?

Está claro que no nos podemos fiar, pero nuestro cuñado está continuamente diciendo que hay que invertir, que se va a hacer rico, va a dejar de trabajar e incluso ya ha visto el Ferrari que quiere comprarse.

Te ha planteado su técnica infalible para invertir. La idea es que él va a invertir una cantidad de dinero D de la siguiente forma.

- La primera vez comprará independientemente del valor (al precio que sea).
- En el momento en el que la moneda valga el doble, o más, del precio al que compró, él venderá.
- Finalmente, volverá a comprar cuando la moneda valga la mitad, o menos, del valor por el cual vendió la última vez.

Nuestro cuñado es un poco vago y solo es capaz de hacer 1 operación al día. Para verificar si el método funciona, nos ha pedido que realicemos un programa que lo simule usando el histórico del máximo diario alcanzado por la moneda en la que quiere invertir. Las ganancias que va a obtener (según él) se determinan sobre el total final que tenga (ya sea invertido o sin invertir) menos el dinero inicial.

Entrada

La entrada constará de N casos de prueba. Por cada caso de prueba tendremos un número entero D que representa el dinero que el cuñado está dispuesto a invertir. Seguidamente, vendrá un número entero V con el total de datos del histórico que se tiene para la moneda. Y en la siguiente línea parecerán V números enteros, llamados M que denotan el valor entero del máximo para ese día.

Salida

La salida será las ganancias que nuestro cuñado obtendría.

Entrada de ejemplo

```
3
100
3
1000 900 500
100
3
1000 2000 500
100
4
1000 2000 500 3000
```

Salida de ejemplo

0
100
300

Explicación caso 3

1. Invierte los 100 euros iniciales cuando la moneda vale 1000.
2. Vende la inversión de 100 euros, puesto que la moneda vale el doble o más. Por lo tanto, dobla su inversión y obtiene un beneficio de 100 euros, $(200 - 100)$.
3. Después vuelve a comprar, porque el valor de la moneda es inferior a la mitad de 2000. En este caso, compra con 200 euros (dinero total que tiene).
4. Finalmente, como el valor de la moneda es el doble, o más, que el valor cuando compró, el cuñado vende la inversión de 200 euros. Él dobla su inversión (400 euros) y obtiene un beneficio final de $400 - 100$, es decir 300 euros.

Límites

- $1 \leq N \leq 100$
- $100 \leq D \leq 1000$
- $1 \leq V \leq 50$
- $1 \leq M \leq 10000$ y $(M \bmod 2) == 0$