

Nivel “Secundaria”
Problemas

13 de Octubre de 2017

Problema 1: Examen General

El examen nacional de matemática se realiza cada año bisiesto en Nlogonia. Cada ciudadano es evaluado, por lo que es posible estudiar el desarrollo de la lógica y las matemáticas en el país a lo largo de los años.

Después de que los exámenes son corregidos, se ordena a los ciudadanos según su nota (cuanto mayor es la nota, mejor es el ciudadano). Los ciudadanos obtienen descuentos en sus impuestos de acuerdo a su posición en este ranking.

La Oficina Central de Estadísticas (OCE) está a cargo de procesar las notas obtenidas en el examen. Este año, sin embargo, Vasya, una de las personas a cargo, está en el hospital, y tú fuiste contratado para terminar su trabajo.

Escribe un programa que, dado el número de ciudadanos y sus notas, responda consultas informando la nota obtenida por el ciudadano en la posición dada.

Entrada

La entrada contiene varios casos de prueba. La primera línea de cada caso contiene dos enteros **N** ($1 \leq \mathbf{N} \leq 100$) y **Q** ($1 \leq \mathbf{Q} \leq 100$), el número de ciudadanos y el número de consultas respectivamente.

Cada una de las siguientes **N** líneas contiene la nota $\mathbf{n_i}$ obtenida por el i-ésimo ciudadano ($0 \leq \mathbf{n_i} \leq 30000$).

Cada una de las siguientes **Q** líneas contiene una posición $\mathbf{p_i}$.

La entrada finaliza con el final de archivo (EOF).

Salida

Para cada caso de prueba, imprime, para cada consulta, una línea que contenga la nota del ciudadano en la posición $\mathbf{p_i}$.

Ejemplos

Entrada	Salida
6 5 30 30 40 250 100 15 1 5 3 2 4	250 30 40 100 30

Problema 2: Kaprekar

El número 6174 es conocido como constante de Kaprekar en honor al matemático hindú Dattathreya Ramachandra Kaprekar. Este número es interesante por lo siguiente: sea X cualquier número de 4 dígitos (ceros a la izquierda están permitidos) en el que no todas sus cifras son iguales, luego la rutina de Kaprekar que empieza en este número siempre converge a 6174. Esto es, la rutina de Kaprekar converge a 6174 si, y sólo si, X es un número de 4 dígitos y al menos dos de ellos son distintos. La rutina de Kaprekar se define de la siguiente manera:

```
int kaprekar(int X) {
   int cnt = 0;
   while (X != 6174) {
       int may = mayor_numero_con_digitos_de(X);
       int men = menor_numero_con_digitos_de(X);
       X = may - men;
       cnt = cnt + 1;
   }
   return cnt;
}
```

`mayor_numero_con_digitos_de(X)` es el mayor número que puede ser formado usando los dígitos de X.
`menor_numero_con_digitos_de(X)` es el menor número que puede ser formado usando los dígitos de X.

Por ejemplo:

```
mayor_numero_con_digitos_de(3524) = 5432
menor_numero_con_digitos_de(3524) = 2345
mayor_numero_con_digitos_de(10) = 1000 //porque 10 = 0010 con 4 dígitos
menor_numero_con_digitos_de(10) = 1
```

Entrada

La primer línea contiene un entero **T** ($1 \leq T \leq 10^4$), la cantidad de casos de prueba. Cada caso de prueba consiste de una línea con un entero **X** ($0 \leq X \leq 9999$).

Salida

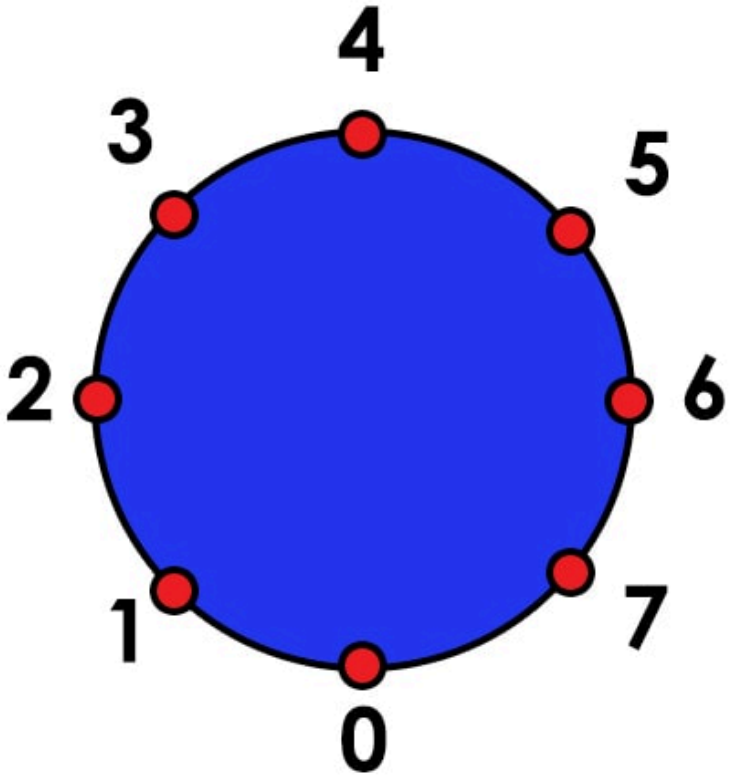
Para cada caso de prueba imprima “Caso **#X**: **Y**”, donde **X** es el número de caso correspondiente, empezando en 1, y **Y** es el retorno de la rutina de Kaprekar ó -1 en caso que entre en un bucle infinito.

Ejemplo

Entrada	Salida
3 3524 0 10	Caso #1: 3 Caso #2: -1 Caso #3: 5

Problema 3: Laguna Azul

La Laguna Azul es un estanque redondo donde varios pájaros viven tranquilamente (ó no tanto...). Bino,un chico, quiere capturar todos los pájaros que viven en la Laguna Azul. Hay **P** sitios en el estanque donde un pájaro se puede quedar, como se puede ver en la figura debajo para **P** = 8.



Cada pájaro puede realizar un número de vuelos, luego de estos, el pájaro está demasiado cansado para volar de nuevo. Bino empezará en la posición 0, y luego camina en una dirección en sentido horario, hasta que captura todos los pájaros.

Cada pájaro tiene una lista ordenada de lugares a donde prefiere ir si Bino llega a donde está. Por ejemplo, para el primer caso de prueba, hay un solo pájaro que empieza en el sitio 1, cuando Bino llega, el pájaro escapa al sitio 2, cuando Bino llega, el pájaro escapa al sitio 3, y cuando Bino llega al 3, el pájaro está cansado, y el mismo es capturado.

Tu tarea es encontrar cual es mínimo de vueltas al estanque que Bino debe dar para capturar a todos los pájaros. En un vuelta completa, Bino visita todos los sitios, y retorna a la posición 0 (revisitando la posición 0).

Está garantizado que ningún pájaro empieza en la posición 0, y también que ningún pájaro trata de escapar al mismo lugar donde está.

Entrada

La primer línea contiene dos enteros **A** ($1 < \mathbf{A} \leq 10^3$) y **P** ($1 < \mathbf{P} \leq 10^9$), representando respectivamente el número de pájaros y el número de sitios. Luego lo siguen **A** líneas. Cada línea empezará con un entero $\mathbf{N_i}$ ($1 \leq \mathbf{N_i} \leq 10^3$) representado la cantidad de sitios al que el pájaro intentará escapar, luego lo siguen $\mathbf{N_i}$, representando la lista de sitios donde el pájaro escapará.

Salida

Imprimir una sola línea conteniendo la cantidad mínima de vueltas al estanque que Bino debe tomar para capturar todos los pájaros.

Ejemplo

Entrada	Salida
1 8 3 1 2 3	1
2 8 3 1 2 3 3 3 6 1	2

Problema 4: Viaje a Marte en velocidad prima

Un grupo de científicos está realizando nuevas experiencias para crear una nave espacial que permita viajes a Marte mucho más rápidos que en la actualidad. Esta nave espacial utilizará dos cohetes con un nuevo combustible, mucho más eficiente que los utilizados actualmente. Pero la velocidad que los nuevos cohetes pueden proveer a la nave espacial está relacionada directamente al peso del combustible almacenado en estos cohetes (en kg) y a una curiosa relación, entre números primos del peso de combustible. Por ejemplo, si el peso total del combustible del cohete es 1010 kg, la máxima velocidad (en km/h) será la suma del décimo número primo empezando por 1010 (inclusive): 1013 -> 1019 -> 1021 -> 1031 -> 1033 -> 1039 -> 1049 -> 1051 -> 1061 -> 1063, ó 10380 km/h.

Los científicos están muy intrigados por esta relación matemática y quieren que Ud. realice un programa que calcule el tiempo, que le tomará a la nave espacial, llegar a Marte desde la Tierra con este nuevo combustible, dado el peso de los cohetes (por supuesto, se intentan crear cohetes del mayor tamaño posible) y asumiendo que la distancia desde la Tierra a Marte, el día del lanzamiento, será de 60 millones de kilómetros.

Entrada

La entrada contiene un solo entero **peso** (1000 < **peso** ≤ 60000) indicando el máximo peso del combustible (en kg) que puede ser almacenado por los cohetes.

Salida

Tu programa debe mostrar dos líneas. La primer línea contiene la velocidad alcanzada por la nave espacial, seguida del texto "km/h". La segunda línea contiene el tiempo estimado de viaje a Marte en horas y días (truncado) con sus mensajes de texto correspondientes, como se muestra abajo.

Ejemplo

Entrada	**Salida**
`1010`	`10380 km/h` `5780 h / 240 d`
`60000`	`600578 km/h` `99 h / 4 d`

Problema 5: Virus

El departamento de Salud Pública de Longoria acaba de publicar un alerta. Un virus está infectando toda la población.

Después de muchos estudios, los investigadores de Longoria encontraron que después de infiltrarse en un cuerpo huésped, los virus se unen en parejas para llegar a ser letales. El nivel de letalidad de una infección es determinado por la suma de la diferencia de edad en días de los pares de virus. Los virus no unidos no influencian el nivel.

Por lo tanto, si hay 4 virus en el cuerpo huésped con edades (en días), que son iguales a:
4, 10, 9, 43
Y están unidos de la siguiente manera:
4 con 9 y 43 con 10

Entonces el nivel de letalidad será de (9-4)+(43-10)=38.

El departamento de Salud Pública de Longoria te solicitó a ti que escribas un programa que, dado la cuenta de virus en un huésped y la edad de cada uno de ellos, calcule el nivel máximo de letalidad que puede alcanzar la infección.

Entrada

La entrada contiene varios casos de pruebas. La primera línea de cada caso de prueba consiste de un entero **N** ($1 \leq \mathbf{N} \leq 1000$), el número de virus en el cuerpo huésped. La siguiente línea contiene **N** enteros $\mathbf{a_i}$ ($0 \leq \mathbf{a_i} \leq 1000$) separados por un espacio en blanco, las edades (en días) de todos los virus en el cuerpo huésped.

La entrada termina con un final de línea (EOF).

Salida

Para cada caso de prueba, imprimir una sola línea con el nivel máximo de letalidad que puede alcanzar la infección.

Ejemplos:

Entrada	Salida
4 4 9 43 10 3 0 100 50	40 100

Problema 6: Máquina de Café

El nuevo edificio de los Trescientos Matemáticos Tecnológicos (TMT) tiene 3 pisos. En ciertas épocas del año, los funcionarios del TMT beben mucho café. Por ello, la presidencia del TMT decidió regalar a los funcionarios una nueva máquina de expreso. Esta máquina debe ser instalada en uno de los 3 pisos, pero la instalación debe ser hecha de forma que las personas no pierdan mucho tiempo subiendo y bajando escaleras.

Cada empleado del TMT bebe 1 café expreso por día. Cada uno necesita ir del piso donde trabaja hasta el piso donde está la máquina y volver a su puesto de trabajo. Todo funcionario tarda 1 minuto para subir o bajar un piso. Como el TMT se preocupa mucho por la eficiencia, quiere posicionar la máquina para minimizar el tiempo total gastado subiendo y bajando escaleras.

Tu tarea es ayudar a la dirección a posicionar la máquina para minimizar el tiempo total gastado por los empleados subiendo y bajando escaleras.

Entrada

La entrada consiste de 3 números $\mathbf{A_1}$, $\mathbf{A_2}$, $\mathbf{A_3}$ ($0 \leq \mathbf{A_1}$, $\mathbf{A_2}$, $\mathbf{A_3} \leq 1000$), cada uno en una línea diferente. El número $\mathbf{A_k}$ representa el número de personas que trabajan en el **k**-ésimo piso.

Salida

Su programa debe imprimir una única línea que contenga la cantidad de minutos que serán gastados si la máquina se coloca en el mejor lugar posible.

Ejemplos

Entrada	Salida
10 20 30	80
10 30 20	60
30 10 20	100

Problema 7: Construyendo Paredes

Después de que el colosal titán destruyera la pared de Maria, la Tropa de Exploración ha decidido construir una nueva, esta pared será tan dura que ningún titán podrá destruirla.

Pero si el titán es demasiado alto este simplemente puede saltar por encima de la pared, a causa de esto el Ejército de Exploración te contrató para que escribas un programa, que dada la altura de la pared y el tamaño de los titanes conocidos, responda cuales titanes pueden pasar por encima de la pared.

Un titán puede saltar por encima de la pared sólo si es más alto que ella.

Entrada

La primer línea de entrada contiene dos enteros **N** ($1 \leq N \leq 100$) y **W** ($1 \leq W \leq 1000$) representando cuantos titanes la Tropa de Exploración conoce y el tamaño de la pared que intenta construir.

Cada una de las siguientes **N** líneas contiene un string **S** ($1 \leq |S| \leq 100$) representando el nombre del titán, seguido por un entero **H** ($1 \leq H \leq 1000$) representando la altura del titán. El string está compuesto por minúsculas, mayúsculas y espacios.

El nombre del titán nunca comienza o termina con espacio.

Salida

Tu programa debe mostrar cuáles titanes podrían pasar por encima de la pared, los titanes deben ser mostrados en el orden que aparecieron en la entrada.

Ejemplo

Entrada	Salida
3 50 Titan Colossal 60 Titan Encoracado 15 Titan Femea 14	Titan Colossal

Problema 8: Imagen

Rafael encontró un nuevo hobby: escribir usando caracteres del teclado. Aunque este nuevo tipo de arte es simple y limitado, la creatividad es suficiente para hacer muy buenos dibujos.

Después de hacer un par de dibujos, Rafael se preguntó cómo podrían ser si eran redimensionados, pero tener que rehacer todos los dibujos parecía un poco cansador y te preguntó si lo ayudabas.
Cuando quieres redimensionar algo, una imagen de **N** líneas y **M** columnas tendrá **A** líneas y **B** columnas, y, dado que las nuevas dimensiones son más grandes que las dimensiones originales, algunos caracteres tendrán que repetirse.

Digamos que **A** es 3 veces más grande que **N**. En ese caso, cada línea tendrá que repetirse 3 veces, así la imagen es redimensionada correctamente.

Dado un dibujo hecho por Rafael, muestra como sería el dibujo si fuera redimensionado a una nueva dimensión dada

Entrada

Habrá varios casos de prueba.
Cada caso de prueba comienza con dos enteros **N** y **M** ($1 \leq N, M \leq 50$), representando respectivamente, la altura y el ancho del dibujo de Rafael.
A continuación vienen N líneas, cada una conteniendo M caracteres, representando el dibujo hecho por Rafael. Después, vienen dos enteros **A** y **B** ($N < A \leq 100$, $M < B \leq 100$, **A** es múltiplo de **N**, y **B** es múltiplo de **M**), representando, respectivamente, la nueva altura y el nuevo ancho que Rafael quiere que el dibujo tenga.
El último caso de prueba es indicado cuando **N**=**M**=0, que no deberá ser procesado.

Salida

Para cada caso de prueba, imprimir **A** líneas, conteniendo **B** caracteres cada una, representando el dibujo de Rafael redimensionado

Ejemplo

Entrada	Salida
3 3 ### #__ ### 6 9 0 0	######### ######### ###______ ###______ ######### #########

Problema 9: Super Primos: Ataquen!

La Asociación de Primos Indivisibles eligio una categoría de números primos llamada Super Primos. Un número es considerado super primo si en adición a ser primo, todos sus dígitos son primos. La Asociación te solicitó a ti que hagas un programa que caracterice los números.

Entrada

La entrada contiene varios casos de prueba, cada caso de prueba comienza con un entero **N** ($0 < \mathbf{N} < 10^5$) en una sola línea. La entrada finaliza en el último caso de prueba.

Salida

Para cada caso de prueba, la clasificación del número de entrada se espera en una sola línea, que puede ser: **“Super”**, si el número es un Super Primo; **“Primo”** si el número sólo es un primo; ó **“Nada”** si el número tiene divisores más allá del 1 y el mismo.

Ejemplo

Entrada	Salida
23 33 43	Super Nada Primo

Problema 10: Quién es el más rápido?

Otavio, Bruno e Ian son amigos de la infancia, apasionados por los desafíos y los deportes acuáticos. En esta época de Olimpiadas se desafían entre ellos, simulando algunas competiciones como natación. El problema es que en la natación, por ejemplo, se entrenan juntos y a veces la diferencia de tiempo entre ellos es muy pequeña, por eso, en la mayoría de los casos, son horas y horas de discusión sobre quién ganó. Ahora decidieron invertir en el desarrollo de equipos electrónicos para ser utilizados específicamente en la natación, que identifican el momento en que terminaron de nadar y muestran quién fue el más rápido. Eres parte del equipo de desarrollo y tu tarea en el proyecto es codificar un programa que reciba la hora de cada uno de los 3 amigos y diga quién fue el ganador.

Entrada

La primer línea de entrada contiene un entero QT (1 ≤ QT ≤ 105), indicando el número de casos de pruebas. Cada uno de los casos de pruebas consta de tres números, separados por un espacio en blanco, O (0 ≤ O ≤ 100), B (0 ≤ B ≤ 100) y I (0 ≤ I ≤ 100), que representan, respectivamente, el tiempo en segundos de Otavio, Bruno e Ian. Los tiempos tendrán un máximo de tres cifras decimales.

Salida

Para cada caso de prueba, el programa deberá imprimir una línea con el nombre del ganador, el más rápido. Si hay un empate y no se puede determinar un único ganador, deberá imprimir la palabra "Empate" sin las comillas.

Entrada	Salida
7 1 2 3 2 0.459 1 8 9 6.7 3.000 3.000 3.000 8.05 11.10 8.001 2.09 3.40 3.008 2.03 2.04 2.02	Otavio Bruno Ian Empate Ian Otavio Ian

Anexo: cómo manejar entradas cuando no se conoce la cantidad de datos que ingresan?

¿Qué es una entrada que termina con EOF?

En este tipo de entrada no se especifica la cantidad de casos de prueba. Pueden ser uno, dos o más de un millón.
Por ejemplo, un archivo de entrada con tres casos de prueba podría ser así:

7123
32
125

En C++, esta lectura se puede resolver de la siguiente manera:

```
int N;
while(cin >> N){
...
}
```

Esto significa que mientras haya valores enteros en el archivo de entrada, se lee la variable N.

En un entorno Windows, cuando este programa se ejecuta, el fin de datos viene indicado por la pulsación de Ctrl-Z (mantener pulsada la tecla Ctrl, y simultáneamente pulsar Z).
Ctrl-Z indica EOF (End Of File) y la condición del while será falsa.