

## CONCURSO ESTATAL DE PROGRAMACIÓN



3 Y 4 DE MAYO

INSTITUTO TECNOLÓGICO SUPERIOR  
DEL SUR DE GUANAJUATO**Problemas**

Este documento contiene 11 problemas.

**Autores de los problemas**

Kuko,UG/CIMAT.

Hugo Humberto Morales Peña, UTP Colombia &amp; RPC.

Luis Germán Gutiérrez Torres, ITSUR

David Morales Orozco, ITSUR

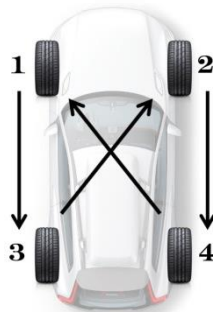
**Uriangato, Gto. 4 de mayo de 2017.**

## Problema A - Cuidando las llantas

Autor: Luis Germán Gutiérrez Torres, ITSUR.



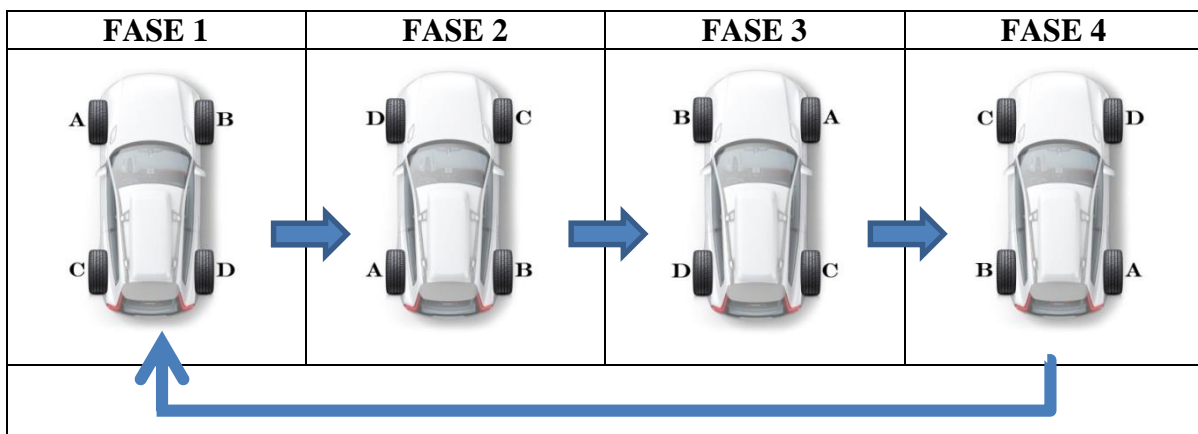
La rotación de llantas es una parte importante del mantenimiento del vehículo. Si no se realiza esta rotación las llantas se desgastan mucho más rápido. La idea de este proceso es tener cada llanta pasando parte de su vida en cada posición en el carro. Para llevar a cabo esto, la llanta se mueve a otra ubicación de acuerdo a un patrón. Primero, asumimos que las posiciones de la llanta están numeradas de izquierda a derecha, del frente hacia atrás. Entonces establecemos un patrón de rotación:



En el diagrama podemos ver que para cada fase del ciclo de rotación una llanta es movida desde una posición a otra de acuerdo a la siguiente tabla:

Posición inicial		Posición final
1	→	3
2	→	4
3	→	2
4	→	1

Por lo tanto, si nuestras cuatro llantas están representadas por A, B, C y D, hay cuatro fases válidas del ciclo de rotación:



Escribir un programa que tome una cadena inicial y una cadena final, ambas representan las llantas de un carro. Cada carácter será una letra mayúscula ('A' – 'Z') del alfabeto inglés (es decir, no hay letra Ñ) y representará un número que identifica una llanta. La cadena inicial representa la ubicación de las llantas antes de la rotación y la cadena final representa la ubicación de las llantas después de una o varias rotaciones. La posición de un carácter representa la ubicación donde se encuentra la llanta. El carácter representa las llantas en el orden 1, 2, 3, 4 (correspondientes al diagrama).

Utilizando el patrón de rotación mostrado, tu programa debe regresar 1, 2, 3 o 4 si las llantas están en la primera, segunda, tercera o cuarta fase del ciclo de rotación. Si las llantas han sido rotadas de forma inapropiada (esto es, no están en ninguna fase), tu programa debe imprimir -1.

## Entrada

En la primera línea se leerá una cadena de cuatro caracteres (A – Z) del alfabeto inglés que representa la posición inicial de las cuatro llantas en el orden 1, 2, 3, 4 correspondiente al diagrama. Los cuatro caracteres de serán distintos. En la segunda línea se leerá también cadena de cuatro caracteres (A – Z) del alfabeto inglés que representa la posición final de las cuatro llantas.

## Salida

Se debe imprimir un solo número entero (1, 2, 3 o 4) que representa la fase de rotación en la que se encuentran las llantas de acuerdo a las posiciones inicial y final.

El programa debe imprimir -si la cadena final no corresponde a ninguna de las fases del ciclo de rotación.


























Ejemplo de Entrada 1	Ejemplo de Salida 1
WXYZ WXYZ	1
Ejemplo de Entrada 2	Ejemplo de Salida 2
LOCA OLAC	3
Ejemplo de Entrada 3	Ejemplo de Salida 3
ABCD ABDC	-1

## Problema B - Globos de Feria


























Autor: David Morales Orozco, ITSUR.



Se tiene un tablero electrónico el cual tiene representado globos de color rojo (oscuro) y globos de color azul (claro), los cuales están establecidos en filas que se indican por una letra del alfabeto inglés, y columnas que están indicadas por un número que comienza en el número 1, como se muestra en la siguiente figura.

	1	2	3	4	5
A					
B					
C					
D					
E					

Dicho tablero se puede configurar para presentar diferentes combinaciones de globos de color rojo y azul, para que una vez este configurado dicho tablero, se le puede disparar con una pistola del tipo Famicom Light Gun (Pistola del NES) y el objetivo del juego es atinar a un número determinado de globos rojos para poderse hacer acreedor de un premio, cuando aciertas en un blanco el globo se le pone una marca que indica que ya has acertado, esto se muestra en la siguiente figura:

	1	2	3	4	5
A					
B					
C					
D					
E					

La cantidad de disparos que tiene un jugador se configura al iniciar cada juego, para los cuales si le atinas a las coordenadas de un globo este se marca con una X, lo cual indica que ya has atinado a este globo, para finalizar de acuerdo al número exacto de globos rojos punchados, se te dará o no un premio, para cuestiones de ejemplo en la siguiente figura se puede ver que si se atina exactamente a un globo rojo, gana un premio chico, si le atina a exactamente 3 globos rojos gana un premio mediano, y si atina a exactamente a 5 globos rojos gana un premio grande.

```
LISTA DE PREMIOS
1 Chico
3 Mediano
5 Grande
```

**Notas:** Puede haber disparos que den en el mismo blanco, pero esto significa que solo acertó a un único globo.

## Entrada

Un número  $1 \leq P \leq 10$ , que indica el número de premios que hay por entregar, seguido de  $P$  líneas, que indican por cada  $P_i$  líneas un número  $1 \leq X \leq 15$  que indica la cantidad exacta de globos rojos que debes atinar, para conseguir dicho premio, seguido de un espacio para continuar con una cadena  $NP$  que indica nombre del premio al cual se haría acreedor esta cadena no incluye espacios y solo son caracteres alfanuméricos del alfabeto ingles.

Seguido de 2 números  $2 \leq F \leq 20$  y  $2 \leq C \leq 20$  cada uno en su propia línea, los cuales indican el tamaño de filas ( $F$ ), y el tamaño de columnas ( $C$ ) que tiene el tablero de globos, para continuar se presentarán  $F$  líneas los cuales contendrán  $C$  caracteres separados por un espacio en blanco que indican si en esa posición hay un globo rojo o un globo azul (caracteres 'R' y 'C'), correspondientes a esa fila del tablero.

Un número  $1 \leq D \leq 20$  que indica el número de disparos que se hicieron con la Famicom Light Gun, seguido de  $D$  líneas que indican la fila y columna en la cual el jugador hizo el disparo.

## Salida

La siguiente frase:

Fueron **#N** globo(s) rojos y **#premio**.

Donde **#N** es el número de globos rojos a los cuales le atino el usuario, y **#premio** es el premio al que se hizo acreedor de acuerdo a lo especificado en la entrada, si no gano premio se indica con la frase "no gano premio" sin comillas, y por ejemplo si gano el premio mediano se indicaría "gano premio mediano", es decir la frase gano premio seguido del premio que fue acreedor el concursante.

Ejemplo de Entrada 1	Ejemplo de Salida 1	Descripción
3 1 Chico 3 Mediano 5 Grande 5 5 R R R R R R R R R R R R A R A R R R R R R R R R R 5 A 1 A 2 C 5 D 2 E 3	Fueron 4 globo(s) rojos y no gano premio.	El tipo de premio es una cadena de caracteres sin espacios.

Ejemplo de Entrada 2	Ejemplo de Salida 2	Descripción
2 1 Mapache 3 Oso 5 5 R R R R R R R R R R R R A R A R R R R R R R R R R 1 A 1	Fueron 1 globo(s) rojos y gano premio Mapache.	El tipo de premio es una cadena de caracteres sin espacios.

## Problema C – Virus duplicador.

Autor: Kuko, UG/CIMAT.



### Descripción

La detección de virus en los equipos de cómputo es vital para su eliminación. Recientemente en ATSA Technologies han descubierto un nuevo virus que tiene la capacidad de duplicarse antes la amenaza de ser eliminado. Han descubierto que se aloja en el registro de memoria de la RAM y ante la amenaza se pasa al registro de la memoria ROM.

Los datos del registro de memoria de la RAM, están numerados y cada dato tiene asociado una cantidad de memoria. De igual manera están definidos los datos de la memoria ROM. Particularmente en ambos registros, los datos están acomodados de manera **no creciente** de acuerdo a su cantidad de memoria.

Para evitar ser eliminado, el virus se mueve de manera inteligente entre los registros; para evitar ser detectado, si el virus se encuentra en el dato  $i$  con cantidad de memoria  $m_i$ , entonces el virus se **pasa** a un dato  $j$  ( $i < j$ ) en la ROM con cantidad de memoria  $r_i$ , tal que  $r_i \leq m_i$ . Descubrieron además, que el virus siempre trata de pasarse a la ROM, desde un dato donde tenga la mayor cantidad de opciones.

ATSA Technologies ha contratado a tu equipo de programadores para ayudarlos a encontrar la máxima cantidad de registros de la ROM, en los que podría estar el virus.

### Entrada

En la primera línea habrá un número entero  $N$  que representa la cantidad de datos que tiene cada registro, tanto de la RAM como de la ROM.

En la siguiente línea habrá  $N$  números enteros que representan las cantidades de memoria  $\{m_i\}$  de los datos en la RAM. El primer número es la memoria del dato 1, el segundo indica la cantidad de memoria del dato 2, etc.

En la última línea, al igual que en la anterior, se encuentran las cantidades de memoria  $\{r_i\}$  de los datos de la ROM.

## Salida

Un único número que represente la máxima cantidad de datos en los que podría estar el virus.

## Ejemplo

Entrada	Salida	Descripción
<pre> 6 10 9 6 6 5 1 9 9 8 8 7 6 </pre>	3	<p>Si el virus estuviera en el primer dato, no podría pasarse a ningún de los datos de la ROM. Si estuviera en el dato 2 (con memoria 9), únicamente podría pasarse al dato 2 de la ROM.</p> <p>Sin embargo, si estuviera en el dato 3 (con memoria 6) podría pasarse al dato 4, 5 y 6.</p>
<pre> 5 5 4 3 2 1 9 8 7 6 5 </pre>	4	

## Límites

- $0 < N < 10^6$
- $0 < m_i, r_i \leq 10^3$ , donde  $m_i \geq m_{i+1}$  y  $r_i \geq r_{i+1}$



## Problema D – Simulador de ofertas.

Autor: Kuko, UG/CIMAT.



### Descripción

Entre las muchas aplicaciones que desarrolla ATSA Technologies, existe una aplicación que ayuda a las empresas, simulando el lanzamiento de nuevas ofertas entre dos productos.

Para utilizar la aplicación, la empresa ingresa al sistema dos productos, digamos  $A$  y  $B$ , donde cada producto regala un cupón. La aplicación simula dos tipos de ofertas que se pueden hacer en la tienda principal de la empresa:

- Presentas  $R_A$  cupones del producto  $A$  más  $R_B$  cupones del producto  $B$  y obtiene un producto  $A$ .
- O bien, presentas  $S_A$  cupones del producto  $A$  más  $S_B$  cupones del producto  $B$  y obtienes un producto  $B$ .
- En ambos casos, el producto que recibes, también tiene un cupón.

En la última versión, la aplicación simula ambas ofertas para determinar cual oferta es mejor; sin embargo en una nueva actualización que se desea implementar, quieren determinar cual sería el máximo costo de producción de todos los productos  $A$  y  $B$  mientras sea posible cambiar cupones por productor.

Para ello, ATSA Technologies ha lanzado el concurso CODING CUP para que puedan implementar esta nueva actualización.

### Entrada

En la primera línea habrá dos números  $P_A$  y  $P_B$  que representan la cantidad de productos  $A$  y  $B$  que se lanzaron con cupones.

En la siguiente línea habrá cuatro números enteros separados por espacio,  $R_A$ ,  $R_B$ ,  $S_A$  y  $S_B$ .

En la última línea habrá dos números enteros  $C_A$  y  $C_B$  que representan los costos de producir un producto  $A$  y  $B$  respectivamente.

### Salida

Un único número que represente el máximo costo de producción que se puede tener mientras sea posible aprovechar la oferta.

## Ejemplo

Entrada	Salida	Descripción
4 3 2 1 1 2 2 6	40	<p>Primero definamos "XCY" como "X cupones del producto Y".</p> <p>Lo peor que le puede pasar a la empresa es que se utilicen los cupones de la siguiente manera.</p> <p>.1. Con los productos iniciales se tienen 4CA y 3CB.</p> <p>.2. Después se utiliza la oferta de 1CA + 2CB, para obtener UN producto <i>B</i>; entonces quedarán 3CA y 1CB.</p> <p>.3. Con el nuevo producto B, se obtiene un nuevo cupón, así al final se tendrán 3CA y 2CB.</p> <p>.4. Ahora se utiliza la oferta para adquirir un producto A, y utilizando el cupón de ese mismo producto, al final se tendrán 2CA y 2CB.</p> <p>.5. Finalmente se adquiere otro producto B, para terminar con 1CA y 1CB, con los que ya no se pueden adquirir más productos.</p> <p>En total el costo de producción fue:</p> <p>(4 productos A iniciales + 1 producto A por cupones) * 2 + (3 productos iniciales B + 2 productos B por cupones) * 6 = 40</p>

## Límites

- $0 < P_A, P_B \leq 10^6$
- $0 < R_A, R_B, S_A, S_B \leq 10^6$
- $0 < C_A, C_B \leq 100$

## Problema E - Frecuencia cardíaca en paracaidismo

Autor: David Morales Orozco, ITSUR.



En el mundo del paracaidismo después de algunos trágicos accidentes de algunos de sus pioneros, se comenzó a utilizar como espectáculo en ferias sobre todo usando a animales pequeños que eran lanzados desde plataformas para deleite del público y quizás sufrimiento de los animales pero siempre se podían ver que caían sin hacerse ningún daño, después de estos espectáculos podemos ver hasta dónde ha llegado esta disciplina con la hazaña del 14 de octubre de 2012 donde Felix Baumgartner alcanzo la estratosfera para después saltar en caída libre para después ser apoyado por un paracaídas para alcanzar un aterrizaje perfecto con el cual estableció el record de altura en 39 kilómetros.

Dentro de este contexto el Dr. Blade un ferviente estudioso de las caídas en paracaídas está realizando un estudio para probar una droga experimental la cual espera reduzca la frecuencia cardíaca de las personas que saltan en paracaídas, por lo cual el Dr. te ha pedido que realices un programa informático en el que se puedan registrar 8 mediciones de frecuencia cardíaca y tú programa indique en qué etapa del salto se registró la frecuencia cardíaca menor las etapas del salto se describen en la siguiente tabla (para facilidad del programa informático los nombres de las etapas van sin acentos):

# Medición	Nombre de la etapa	Frecuencia Cardíaca
1	Inicial	68
2	Colocar paracaidas	70
3	Subida al avion	82
4	Despegue	71
5	Salto	74
6	Apertura de paracaidas	84
7	Aterrizaje	88
8	Recuperacion	75

Para el ejemplo mostrado en la tabla la etapa en la que se registro la menor frecuencia cardíaca es: Colocar paracaidas es importante destacar que no hay dos frecuencias cardíacas iguales registradas, es decir siempre habrá una que es menor a las demás.

## Entrada

8 números enteros  $50 \leq F_i < 220$  que representan la frecuencia cardíaca en cada una de las etapas del salto en paracaídas.

## Salida

Mensaje con la etapa que corresponde con la frecuencia cardíaca menor, sin acentos.

Ejemplo de Entrada 1	Ejemplo de Salida 1
73 70 82 71 74 84 88 75	Colocar paracaídas

Ejemplo de Entrada 2	Ejemplo de Salida 2
70 75 80 74 110 90 89 92	Inicial

## Problema F – Red de eliminación.

Autor: Kuko, UG/CIMAT.

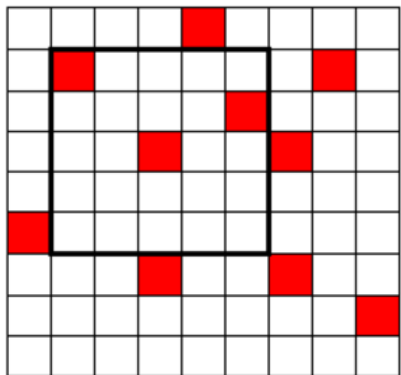


### Descripción

Una de las muchas formas de eliminar virus, es mediante la revisión de los registros de la memoria, pero cuando la cantidad de virus es demasiado grande, se requieren técnicas heurísticas para encontrarlos.

En ATSA Technologies están estudiando una nueva técnica de eliminación que consiste en representar la memoria de la equipos como una matriz de  $N \times N$  celdas, donde en cada celda  $(f, c) = (\text{fila}, \text{columna})$  puede o no haber un virus. A pesar de que se tienen localizados los virus, no es posible eliminarlos uno a la vez porque provocaría una alerta y los virus cambiarían de lugar.

Para eliminar a la mayor cantidad de virus posibles, se ha implementado un módulo llamado, red de eliminación. La red se puede modelar como una matriz de  $M \times M$ , la cual se coloca virtualmente sobre la memoria en algún lugar, de tal manera que las celdas de la matriz de memoria y las celdas de la red, concuerdan perfectamente.



En la imagen de arriba se puede observar la memoria ( $9 \times 9$ ) y los virus (en color rojo); la red de  $5 \times 5$  que si se coloca en el cuadrado negro, eliminaría 3 virus.

ATSA Technologies esta utilizando la CODING CUP para implementar una actualización a la red, la cual consiste en que de manera autónoma, ésta determine cuál es el mejor lugar para colocarse y así eliminar la mayor cantidad de virus. Así que tú y tu equipo podrían ser los nuevos desarrolladores.

### Entrada

Una única línea con tres números  $N$ ,  $M$  y  $V$  que representan el tamaño de la memoria, el tamaño de la red y la cantidad de virus que hay en la memoria.

En cada una de las siguientes líneas, habrá dos números enteros,  $f_i$  y  $c_i$  que representan la fila y columna de la celda donde se encuentra el virus  $i$ . Nunca hay más de un virus en una celda.

### Salida

Un único número que represente la máxima cantidad de virus que se pueden eliminar con la red.

## Ejemplo

Entrada	Salida	Descripción
9 5 10 1 5 2 2 2 8 3 6 4 4 4 7 6 1 7 4 7 7 8 9	5	Es el descrito en la imagen.

## Límites

- $0 < N, M \leq 10^7$
- $0 < K \leq 2 \cdot 10^3$

## Problema G - PSP.

Autor: Luis Germán Gutiérrez Torres, ITSUR.



### Descripción

Personal Software Process es una metodología para Ingeniería de Software que se utiliza en el ITSUR. En cierta parte de esta metodología es necesario calcular los parámetros de regresión y los coeficientes de correlación de una regresión lineal.

Escriba un programa para:

- Calcular los parámetros de regresión  $B_0$  y  $B_1$ .
- Los coeficientes de correlación  $r_{x,y}$  y  $r^2$  para un conjunto de  $n$  pares de datos.
- Dado un estimado  $X_k$ , calcula una predicción mejorada  $Y_k$  donde  $Y_k = B_0 + B_1 * X_k$ .

Las fórmulas para los cálculos se muestran a continuación:

The formulas for calculating the regression parameters  $\beta_0$  and  $\beta_1$  are

$$\beta_1 = \frac{\left( \sum_{i=1}^n x_i y_i \right) - (n x_{avg} y_{avg})}{\left( \sum_{i=1}^n x_i^2 \right) - (n x_{avg}^2)}$$

$$\beta_0 = y_{avg} - \beta_1 x_{avg}$$

The formulas for calculating the correlation coefficient  $r_{x,y}$  and  $r^2$  are

$$r_{x,y} = \frac{n \left( \sum_{i=1}^n x_i y_i \right) - \left( \sum_{i=1}^n x_i \right) \left( \sum_{i=1}^n y_i \right)}{\sqrt{\left[ n \left( \sum_{i=1}^n x_i^2 \right) - \left( \sum_{i=1}^n x_i \right)^2 \right] \left[ n \left( \sum_{i=1}^n y_i^2 \right) - \left( \sum_{i=1}^n y_i \right)^2 \right]}}$$

$$r^2 = r * r$$

## Entrada

Primero se leerá el valor de  $X_k$ , a continuación se debe leer un número  $n$  que representa el número de pares de datos a recibir donde  $2 \leq n \leq 100$ .

A continuación se leerán  $n$  pares de datos  $x, y$  donde  $x$  e  $y$  son valores de tipo real.

## Salida

Imprimir los valores  $B_0$ ,  $B_1$ ,  $r_{xy}$ ,  $r^2$  y por último  $Y_k$ . Cada uno de ellos en un renglón y redondeados a dos decimales.

## Ejemplo

Entrada	Salida
386	-22.55
10	1.73
130 186	0.95
650 699	0.91
99 132	644.43
150 272	
128 291	
302 331	
95 199	
945 1890	
368 788	
961 1601	



## Problema H - Palíndromos especiales.

Autor: Luis Germán Gutiérrez Torres, ITSUR.



Para el Coding Cup se han sacado de la manga dos tipos los palíndromos, los completos y los parciales. Los COMPLETOS son aquellas palabras que se leen igual de adelante hacia atrás y viceversa y además coinciden sus mayúsculas y minúsculas, por ejemplo “OSO” es un palíndromo completo y “oSo” también lo es.

El otro tipo de palíndromos son los PARCIALES. Se definen como aquellos palíndromos que pueden tener variaciones en las mayúsculas y minúsculas, por ejemplo “IsaacNoRoncaAsi”.

Desarrolla un programa que lea N palabras e imprima primero los palíndromos completos y después los palíndromos parciales.

### Entrada

En la primera línea se leerá un entero N donde  $1 \leq N \leq 100$ , seguido de N cadenas de caracteres. Estas cadenas son palabras sin espacios que contienen solamente letras de la “a” a la “z” del alfabeto inglés y pueden ser mayúsculas o minúsculas.

### Salida

Se debe imprimir el texto “\*PALINDROMOS COMPLETOS\*” seguido de todos los palíndromos completos que se encuentren en la lista con su respectivo salto de línea. Al finalizar esta lista se debe imprimir un salto de línea para separar las dos listas.

A continuación se debe imprimir el texto “\*PALINDROMOS PARCIALES\*” seguido de todos los palíndromos parciales que se encuentran en la lista de entrada con su respectivo salto de línea.

Si no se encuentran palíndromos completos se debe imprimir “-1” en lugar de la lista de palíndromos completos.

Si no se encuentran palíndromos parciales se debe imprimir “-1” en lugar de la lista de palíndromos parciales.

Importante:

La lista de palíndromos completos y la lista de palíndromos parciales se debe imprimir en el mismo orden en el que se fueron leyendo. Revise los ejemplos de entrada y salida que se presentan a continuación.

<p>Ejemplo de Entrada 1</p> <p>10 SamYoSeEsoyMas SAMYOSEESOYMAS AMA Oso oso CompuHp AnitaLavaLaTina Parciales ITSUR Otro PALINDROMOS COMPLETOS: SAMYOSEESOYMAS AMA oso</p>	<p>Ejemplo de Salida 1</p> <p>*PALINDROMOS COMPLETOS* SAMYOSEESOYMAS AMA oso  *PALINDROMOS PARCIALES* SamYoSeEsoyMas Oso AnitaLavaLaTina</p>
<p>Ejemplo de Entrada 2</p> <p>10 AnaLavaLana EvaYaHayAve AtarALaRata LoSeDamaDeSol isaacnoRoncaasi IsaacNoRoncaAsi MasArrozAlaZorraSam SometamosOMatemos YoSoy ATiNoBonita</p>	<p>Ejemplo de Salida 2</p> <p>*PALINDROMOS COMPLETOS* isaacnoRoncaasi  *PALINDROMOS PARCIALES* AnaLavaLana EvaYaHayAve AtarALaRata LoSeDamaDeSol IsaacNoRoncaAsi MasArrozAlaZorraSam SometamosOMatemos YoSoy ATiNoBonita</p>
<p>Ejemplo de Entrada 3</p> <p>3 Suerte enTu Examen</p>	<p>Ejemplo de Salida 3</p> <p>*PALINDROMOS COMPLETOS* -1  *PALINDROMOS PARCIALES* -1</p>

## Problema I - Minutos por kilometro

Autor: David Morales Orozco, ITSUR.



Blade es un corredor principiante y ha ido mejorando poco a poco su nivel, pero él no tiene un celular ni otro dispositivo que le diga que ritmo de carrera hizo durante un entrenamiento, pero como él tiene una computadora, te ha pedido como estudiante de una carrera informática, que le hagas un programa donde Blade escriba cuantos kilómetros corrió y en qué tiempo lo hizo (el tiempo siempre será en minutos) el programa informático debe indicar a qué ritmo de carrera corrió indicado en minutos por kilómetro, es decir si el corrió 10 kilómetros en 60 minutos, tuvo un ritmo de carrera de 6 minutos por kilómetro.

### Entrada

Un número entero  $1 \leq K \leq 42$  con los kilómetros que recorrió Blade, seguido de un número entero con los minutos que tardo en recorrer dichos kilómetros.

### Salida

Un número entero que indica los minutos que tarda en recorrer cada kilómetro en promedio, este resultado debe estar redondeado a dos cifras.

Ejemplo de Entrada 1	Ejemplo de Salida 1
20 120	6.00

Ejemplo de Entrada 2	Ejemplo de Salida 2
5 8	1.60

## Problema J - Ulam' Sequence.

Autor: Hugo Humberto Morales Peña - UTP Colombia & RPC.



### Descripción

An Ulam number is a member of an integer sequence devised by and named after Stanislaw Ulam, who introduced it in 1964. The standard Ulam sequence (the (1, 2)-Ulam sequence) starts with  $U_1 = 1$  and  $U_2 = 2$ . Then for  $n > 2$ ,  $U_n$  is defined to be the smallest integer that is the sum of two distinct earlier terms in exactly one way and larger than all earlier terms.

As a consequence of the definition, 3 is an Ulam number ( $1 + 2$ ), and 4 is an Ulam number ( $1 + 3$ ), here  $2 + 2$  is not a second representation of 4, because the previous terms must be distinct. The integer 5 is not an Ulam number, because  $5 = 1 + 4 = 2 + 3$ .

For this problem, you have to write a program that calculates the  $n$ -th term in Ulam's sequence. That is, determine  $U_n$ .

### Entrada

Input begins with an integer  $t$  ( $1 \leq t \leq 4 \times 10^3$ ), the number of test cases, followed by  $t$  lines, each line contains an integer  $n$  ( $1 \leq n \leq 10^4$ ).

### Salida

For each test case, you should print a single line containing the value of  $U_n$ .

### Ejemplo

Entrada	Salida	Descripción
15	1	Use fast I/O methods
1	2	
2	3	
3	4	
4	6	
5	8	
6	11	
7	13	
8	16	
9	18	
10	26	
11	28	
12	36	
13	38	
14	47	
15		

## Problema K – Imágenes Arbolarias.

Autor: Kuko, UG/CIMAT.



### Descripcion

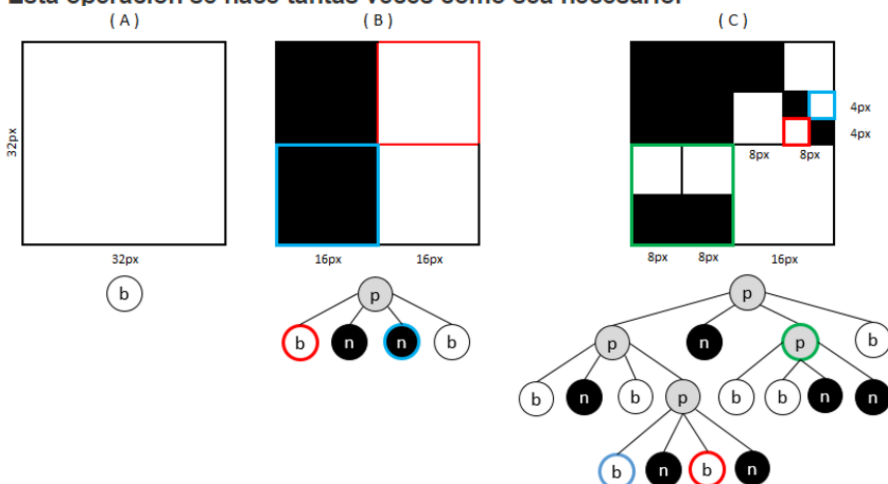
Atsa, el gran tecnologo de la ciudad y CEO de ATSA Technologies, ha escuchado de las habilidades de tu equipo de programadores, así que los ha contratado para trabajar con imágenes binarias, es decir, que tienen únicamente pixeles negros y blancos. Estas imágenes son de tamaño  $32 \times 32$  ( $32 \times 32 = 1024$ ) y son guardadas mediante una representación llamada representación de árbol.

En la representación de árbol, una imagen se puede dividir en 4 partes iguales llamadas cuadrantes. Cada uno de estos se puede o no dividir nuevamente en 4 cuadrantes, y estos a su vez también pueden ser divididos, etc. **Se considera que la imagen original es también un cuadrante.** Para determinar si se divide o no un cuadrante, éste debe contener tener los 2 colores, es decir, algunos de sus pixeles deben ser blancos y otros negros; al dividir este cuadrante, el cuadrante también tendrá una representación de árbol.



*Orden de los cuadrantes al dividir un cuadrante más grande*

Para la representación de árbol de un cuadrante que fue dividido, se utiliza un nodo llamado nodo padre, el cual tiene 4 nodos hijos (un nodo hijo por cada cuadrante). Si alguno de estos 4 cuadrantes se tiene que dividir, el nodo hijo que estaba asociado a este cuadrante se convierte en nodo padre con 4 nodos hijos (como se describió antes). **Esta operación se hace tantas veces como sea necesario.**



Las figuras muestran la representación de las imágenes utilizando nodos padres e hijos. Los nodos se escriben en orden de izquierda a derecha, de acuerdo a la división de los cuadrantes. Los círculos de colores muestran el cuadrante con el que están asociados.

Podemos escribir la representación de árbol, utilizando 3 letras; para un nodo hijo, si el cuadrante al que esta asociado es totalmente blanco entonces usamos la letra  $b$ , o si es totalmente negro, usamos la letra  $n$ . Si un nodo hijo se convirtió en padre, entonces escribimos la letra  $p$  seguida de su representación. Para comenzar la representación se inicia con el cuadrante #1, luego el #2, luego el #3 y finalmente el #4. Por ejemplo la imagen ( $A$ ) se escribiría como  $b$ , la imagen ( $B$ ) como  $pbnnb$ , y la imagen ( $C$ ) como  $ppbnnbpbnnbnpbnnb$ .

Finalmente, la imagen ( $A$ ) no tiene pixeles negros, la imagen ( $B$ ) tiene 512 pixeles negros y la imagen ( $C$ ) tiene 480 pixeles negros.

Su objetivo como equipo, será obtener el resultado de sumar dos imágenes, dadas en su representación de árbol, y obtener el número de pixeles negros que tiene la imagen resultante. Par sumar dos imágenes, se toma el mismo pixel en ambas imágenes y si alguno de ellos es negro, entonces ese mismo pixel en la imagen resultante será de color negro.

## Entrada

En la primera línea encontrarás la representación de la primera imagen ( $I_1$ ).

En la segunda línea encontrarás la representación de la segunda imagen ( $I_2$ ).

Los caracteres de ambas imágenes, siempre serán  $p$ ,  $b$  y  $n$ .

## Salida

Un único número que represente la cantidad de pixeles negros que tendrá la imagen resultante de sumar las dos imágenes  $I_1$  y  $I_2$ .

## Ejemplo

Entrada	Salida	Descripción
b pbnnb	512	Es la suma de las imagenes (A) y (B), cuyo resultado sería la imagen (B).
n pbbbn	1024	La primera imagen es totalmente negra.
pnbnb pbpnbnbbpnbnb	768	
b pbpbpnbnbpbnbnbb	64	

## Límites

- Las cadenas que representan a  $I_1$  y a  $I_2$  siempre serán representaciones válidas de imágenes que miden 32px por 32px.