



UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO  
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y  
MECÁNICA  
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS

ACM CHAPTER CUSCO

CONCURSO DE PROGRAMACIÓN  
**CUSCONTEST XIX**

*PROBLEMSET*

Cusco, 4 de Agosto de 2023

Este problemset contiene 14 problemas etiquetados de la ‘A’ a la ‘N’.

## Información General

A menos que se indique lo contrario, las siguientes condiciones son válidas para todos los problemas.

### Nombre del programa

1. La solución debe ser enviada en formatos del lenguaje seleccionado. Ejm: `codigo.c`, `codigo.cpp`, `codigo.java`, `codigo.py`, `codigo.cs`.

### Entrada

1. La entrada debe ser leída desde la entrada estándar (consola).
2. La entrada consiste en un único caso de prueba, que es descrito en el formato de cada problema. No existen datos extras en la entrada.
3. Cuando una línea de datos contiene muchos valores, estos son separados por exactamente un espacio entre ellos. No existen otros espacios en las entradas.
4. Se utiliza el alfabeto Inglés. No hay letras con tildes, diéresis, eñes, u otros símbolos.

### Salida

1. La salida debe ser escrita como salida estándar (consola).
2. El resultado debe ser escrito en la cantidad de líneas especificada para cada problema. No debe imprimirse otros datos. Ejm: no incluir: “ingrese el número”.
3. Cuando una línea de datos de salida contiene muchos valores, estos deben ser separados por exactamente un espacio entre ellos. No deben imprimirse otros espacios en las salidas.
4. Debe ser utilizado el alfabeto Inglés. No letras con tildes, diéresis, eñes, u otros símbolos.

### Límite de Tiempo

1. El límite de tiempo informado para cada problema corresponde con el tiempo total permitido para la ejecución completa de los casos de prueba.

### Consejos

1. Para leer múltiples números en una línea en Python usa: `A = [int(x) for x in input().split(' ')]`
2. Para soluciones en java, enviar el archivo `.java` sin el “package name”.

Problemas coordinados por Jared León, y planteados por:

Autor	Afiliación
Dennis Huilca	Senior Software Engineer, InnovarITP, PE
Edú Sanchez	Senior Software Engineer, Google, BR
Grover Castro	PhD in Computer Sc., Universität Leipzig, DE
Isaac Campos	Tema lead, Finbe USA, USA
Jared León	PhD Stud. in Maths, University of Warwick, UK
John Vargas	Senior Data Scientist, Topaz, USA
Josué Nina	Data integration and ETL developer, Provista, USA
Rodolfo Quispe	Applied Data Scientist, Microsoft, USA

## Problema A. El profundo impacto de Emma

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1 segundo
Límite de memoria:	64 megabytes
Autor:	John Vargas

La profesora Emma llevó a sus  $n$  alumnos de jardín a visitar el hoyo natural más profundo de la tierra. La profesora hablaba vigorosamente por el celular mientras los niños observaban el pozo muy de cerca hasta que... Oh no! ahora solo hay  $n - 1$  niños, y no hay rastro del alguno más! La profesora entra en pánico, pero rápidamente se tranquiliza al pensar que seguramente el niño fue corriendo de vuelta a la ciudad por haber olvidado su billetera o algo parecido (eso explicaría que los otros niños se encuentren en shock).

La profesora inicialmente tenía a todos los alumnos numerados del 1 al  $n$ , y ahora decide averiguar cuál es el niño que le falta. Ayúdala con esta tarea.

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 100$ ), indicando el número de casos de prueba.

Cada caso de prueba comienza con un entero  $n$  ( $2 \leq n \leq 10^4$ ). La siguiente línea contiene  $n - 1$  enteros distintos  $a_i$  ( $1 \leq a_i \leq n$ ) separados por un espacio y en ningún orden en particular, indicando la lista de niños que todavía están presentes.

### Salida

Para cada caso de prueba, imprime el número del niño desaparecido.

### Ejemplo

Entrada estándar	Salida estándar
2	4
8	2
1 2 3 5 6 7 8	
2	
1	

## Problema B. El juego de la vida de Conway

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1 segundo
Límite de memoria:	64 megabytes
Autor:	Dennis Huilca

En esta ocasión deberás escribir un simulador del Juego de la Vida de Conway en un toro. Es decir, un tablero de  $n \times n$  cuya parte superior está conectada a su parte inferior y cuyas laterales también están conectadas entre sí. Una posición del tablero tiene exactamente 4 vecinos: los que comparten un lado en común.

Se da una regla de actualización para celdas muertas y otra para celdas vivas. Estas reglas son dadas en forma de una cadena con 5 caracteres que contienen los símbolos “\*” y “.”. Para una celda muerta, si el  $i$ -ésimo carácter de su regla correspondiente está vivo y exactamente  $i - 1$  de sus vecinos están vivos, entonces esta celda comienza a vivir en la siguiente generación. Para una celda viva, si el  $i$ -ésimo carácter de su regla correspondiente está vivo y exactamente  $i - 1$  de sus vecinos están vivos, entonces esta celda continúa viva en la siguiente generación.

Por ejemplo, si la regla de actualización para celdas muertas es “..\*..” y para celdas vivas es “\*\*...”, entonces una celda muerta comienza a vivir si y solo si tiene dos vecinos vivos, y una celda viva se mantiene viva si y solo si tiene cero o un vecino vivo.

Calcula la evolución del juego por  $m$  generaciones.

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 10$ ), indicando el número de casos de prueba.

Cada caso de prueba comienza con dos líneas: la regla de actualización para celdas muertas y la regla de actualización para celdas vivas respectivamente. Ambas líneas contienen 5 símbolos que son “\*” o “.”. La tercera línea contiene dos enteros  $n$  ( $2 \leq n \leq 25$ ) y  $m$  ( $1 \leq m \leq 10^3$ ). Cada una de las siguientes  $n$  líneas contiene  $n$  caracteres indicando el estado inicial del tablero. Una celda viva es denotada por “\*” y una muerta por “.”.

### Salida

Para cada caso de prueba, imprime  $n$  líneas indicando el estado final del tablero, cada una conteniendo los símbolos “\*” o “.”.

### Ejemplo

Entrada estándar	Salida estándar
2	...
..*..	.*.
**...	...
3 100	*...
...	**.*
.*.	**.*
...	*...
***..	
.***.	
4 1	
....	
*...	
*...	
....	

## Problema C. Los recortes de Mordor

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1 segundo
Límite de memoria:	64 megabytes
Autor:	Jared León

El mayor enemigo de Sauron, Gándalf se encuentra alterando las tierras de Mordor para tener mayor ventaja estratégica en la batalla de Morannon. Mordor puede ser visto como un rectángulo de lados  $x$  e  $y$  con el mismo número de dígitos. Gándalf decidió que la mayor ventaja real la tendrá cuando el área de Mordor sea la mínima posible. Usando la ayuda de los Ents, Gándalf puede modificar  $x$  e  $y$  de la siguiente manera peculiar. Él puede seleccionar la misma posición en la escritura de ambos dígitos e intercambiarlos. Por ejemplo, si  $x = 3251$  e  $y = 8373$ , Gándalf puede seleccionar la tercera posición, modificando los números a 3271 y 8353 respectivamente. Él puede realizar este cambio las veces que sean convenientes (incluso 0 veces). Ayúdalo a encontrar el área mínima de Mordor que puede alcanzarse realizando estos cambios.

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 100$ ), indicando el número de casos de prueba. Cada caso de prueba contiene dos enteros  $x$  ( $1 \leq x \leq 10^{10^4}$ ) e  $y$  ( $1 \leq y \leq 10^{10^4}$ ), ambos con el mismo número de dígitos.

### Salida

Para cada caso de prueba, calcula el área mínima requerida. Como este valor puede ser muy grande, imprímelo módulo  $10^9 + 7$ .

### Ejemplo

Entrada estándar	Salida estándar
1 25 41	945

## Problema D. El camarada Bourdukovsky

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1 segundo
Límite de memoria:	64 megabytes
Autor:	Edu Sánchez

Estalló la guerra! El camarada Z. Bourdukovsky está a la búsqueda de soldados jóvenes pero no hay suficientes reclutas voluntarios. Por este motivo, el camarada recurre al Gulag más poblado de Kasrilevka para reclutar a los reclusos más jóvenes.

Él tiene una lista de los nombres de los  $n$  prisioneros  $s_1, s_2, \dots, s_n$  junto con el año de su nacimiento  $a_1, a_2, \dots, a_n$ . Lo que decide hacer primero es ordenar su lista del más viejo al más joven, y en caso de colisión de años de nacimiento, alfabéticamente de menor a mayor. Desafortunadamente, la única computadora funcional del Gulag no tiene Microsoft Excel instalado, por lo que el camarada pide tu ayuda para ordenar su lista según sus requerimientos. Escribe un programa para él.

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 100$ ), indicando el número de casos de prueba.

Cada caso de prueba comienza con un entero  $n$  ( $1 \leq n \leq 5 \times 10^3$ ). Luego, cada una de las siguientes  $n$  líneas contiene un número  $a_i$  ( $10^3 \leq a_i \leq 10^5$ ) y una cadena  $s_i$  ( $1 \leq |s_i| \leq 50$ ) sin espacios ni símbolos especiales y cuya primera letra es mayúscula, siendo todas las demás minúsculas. Ambos están separados por un espacio simple.

### Salida

Para cada caso de prueba, imprime los nombres dados en el orden requerido.

### Ejemplo

Entrada estándar	Salida estándar
2	Butkovsky
6	Govorov
1985 Karataev	Karataev
1965 Butkovsky	Glina
1977 Govorov	Ruth
2005 Ruth	Yanet
2005 Glina	Krashnoshtanov
2005 Yanet	Pecinovsky
2	
1929 Krashnoshtanov	
1944 Pecinovsky	

## Problema E. El popular Lucho

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	0.5 segundos
Límite de memoria:	64 megabytes
Autor:	Grover Castro

Lucho es el estudiante más popular del Cusco, es por eso que Movistar lo tiene muy vigilado. Movistar mantiene una lista de los amigos de Lucho y las antenas telefónicas que están muy cerca de sus casas (cada amigo está asociado a exactamente una antena). Algunas de estas antenas están conectadas alámbricamente por  $m$  cables. Para que Lucho se comunique con uno de sus amigos, debe existir una secuencia de antenas conectadas por cables que comience con la antena de Lucho y termine con la antena del amigo en cuestión. Dada la configuración de las instalaciones, Lucho puede comunicarse con todos sus  $n - 1$  amigos. Sin embargo, ya que Lucho también reúne a las personas, la oficina de Movistar dedicada a Lucho está interesada en saber cuántos pares de amigos de Lucho pueden comunicarse entre sí. Acabas de entrar a hacer tus prácticas en Movistar, así que estás encargado de hacer un programa que calcule esta información.

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 100$ ), indicando el número de casos de prueba.

Cada caso de prueba comienza con dos enteros  $n$  ( $2 \leq n \leq 10^4$ ) y  $m$  ( $1 \leq m \leq 4 \times 10^6$ ). Cada una de las siguientes líneas contiene dos enteros  $a_i$  y  $b_i$  indicando que hay una conexión bidireccional entre las antenas  $a_i$  y  $b_i$ . Lucho siempre tiene la antena 1, y siempre hay una secuencia de antenas conectadas desde la antena 1 a cualquier otra antena.

### Salida

Para cada caso de prueba, imprime el número de pares de amigos de Lucho que pueden comunicarse mutuamente.

### Ejemplo

Entrada estándar	Salida estándar
2	6
5 6	0
2 1	
1 3	
1 4	
2 3	
3 4	
5 4	
2 1	
1 2	

Ejemplo 1: Estos son todos los pares de amigos (antenas) de Lucho que pueden comunicarse entre sí:

- 2 - 4: A través de la secuencia de antenas (2, 1, 4).
- 2 - 5: A través de la secuencia de antenas (2, 3, 4, 5).
- 3 - 4: A través de la secuencia de antenas (3, 4).
- 3 - 5: A través de la secuencia de antenas (3, 2, 1, 4, 5).
- 2 - 3: A través de la secuencia de antenas (2, 3).
- 4 - 5: A través de la secuencia de antenas (4, 5).

## Problema F. Cifrado I

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1 segundo
Límite de memoria:	64 megabytes
Autor:	Isaac Campos

Olvidaste la contraseña de tu cuenta bancaria, pero recuerdas que cada vez que cambias de contraseña, guardas la nueva clave cifrada en un papel dentro de un cajón.

La forma de cifrado que usas es bastante particular. Una contraseña cifrada consta de tres pilas de números. Para reconstruirla tienes que extraer un número del tope de alguna de las tres pilas y añadirlo al final de la contraseña que estás reconstruyendo. Luego solo tienes que repetir este proceso hasta que no te queden números en las pilas.

Siendo un gran amante del orden y la lexicografía, sabes que todas las contraseñas que creaste formaran el número más pequeño posible después de descifrarlo.

Para poder acceder a tu cuenta, tendrás que descifrar todas tus contraseñas pasadas, por lo que prefieres crear un programa que las descifre por ti.

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 100$ ), indicando el número de casos de prueba.

Para cada una de las tres pilas, se da un número  $p_i$  ( $1 \leq p_i \leq 10$ ) en una línea, indicando el número de caracteres en la línea siguiente, los cuales son dados separados por un espacio simple, estando el último número en el tope de la pila. Se garantiza que cada contraseña contiene dígitos distintos entre sí.

### Salida

Para cada caso de prueba, imprime la contraseña descifrada.

### Ejemplo

Entrada estándar	Salida estándar
2	129876543
7	271485693
3 4 5 6 7 8 9	
1	
1	
1	
2	
3	
3 9 2	
3	
4 1 7	
3	
6 5 8	

Ejemplo 1: Seleccionas el tope de la segunda pila “1” y lo concatenas al final de la contraseña (que por el momento está vacía). Luego seleccionas el tope de la tercera pila, “2”, y lo concatenas al final de la contraseña. Ahora tu contraseña es “12”, ahora que la segunda y tercera pila están vacías no te queda más que extraer uno a uno los elementos de la primera pila para terminar de formar la contraseña “129876543”.



## Problema G. Cifrado II

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1 segundo
Límite de memoria:	64 megabytes
Autor:	Isaac Campos

Desde la última vez que perdiste tu contraseña aprendiste un par de cosas sobre seguridad y resulta que cifrar tus contraseñas en 3 pilas no es la forma más segura del mundo. Por este motivo, ahora decidiste cifrar tu contraseña en  $n$  pilas utilizando un proceso parecido. Esta vez únicamente tomas el número menor de entre todas las pilas.

Una vez más perdiste la contraseña de tu cuenta para el CusContest, *-no estamos seguros de cómo estás participando-*, pero igualmente guardaste todas las contraseñas que tuviste cifradas. Dada la lista de contraseñas cifradas, descífralas para poder entrar al concurso.

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 100$ ), indicando el número de casos de prueba.

Cada caso de prueba comienza con un entero  $n$  ( $1 \leq n \leq 10^3$ ), indicando el número de pilas. Para cada una de las pilas, el número de elementos de tal pila  $p_i$  se da en una línea, y en la siguiente se dan los elementos de la pila separados por un espacio. Se garantiza que el total de los números de las  $n$  pilas son distintos entre sí y que  $\sum_{i=1}^n p_i \leq 10^3$ .

### Salida

Para cada caso de prueba, imprime la contraseña descifrada.

### Ejemplo

Entrada estándar	Salida estándar
2	42531
2	246378101211951
3	
1 3 5	
2	
2 4	
4	
3	
3 6 2	
3	
10 8 7	
1	
4	
5	
1 5 9 11 12	

## Problema H. Ahora soy formal!

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1.5 segundos
Límite de memoria:	64 megabytes
Autor:	Jared León

Ahora trabajas para SUNAT! Llegó la hora de alegrar algunos negocios pequeños. Te encuentras en una calle con  $n$  tiendas cuyo capital es de  $t_1, t_2, \dots, t_n$ . Lo que harás será seleccionar un rango contíguo de tiendas (sabes que las tiendas más pequeñas suelen estar agrupadas) y pedirles a todas las tiendas en el rango una “contribución” de  $p$  soles para no inspeccionarlas más detalladamente. Debes seleccionar  $p$  de tal forma que cada tienda tenga la capacidad de pagar dicha cantidad. Y obviamente, quieres maximizar la cantidad recolectada. Por fortuna también eres un programador y puedes automatizar esta tarea.

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 10$ ), indicando el número de casos de prueba.

La primera línea de cada caso de prueba contiene un entero  $n$  ( $1 \leq n \leq 3 \times 10^3$ ). La siguiente línea contiene  $n$  enteros  $t_i$  ( $1 \leq p_i \leq 10^5$ ) separados por un espacio.

### Salida

Para cada caso de prueba, imprime la máxima cantidad que puedes recolectar de las tiendas.

### Ejemplo

Entrada estándar	Salida estándar
2	15
6	154
2 4 3 9 4 9	
6	
154 4 4 9 4 9	

## Problema I. Pillao Matao?

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1 segundo
Límite de memoria:	64 megabytes
Autor:	Josué Nina

En el pueblo de Pillao Matao, el alcalde decidió regalar cuchillos conmemorativos a los visitantes en puestos instalados a lo largo de la carretera principal. Estos puestos, se colocaron en los kilómetros que corresponden a múltiplos de  $x$  y también en kilómetros correspondientes a múltiplos de  $y$ . Los números  $x$  e  $y$  son primos. En cada puesto se regala exactamente un cuchillo por persona. Tú acabas de llegar al pueblo y tu objetivo es coleccionar la máxima cantidad de cuchillos antes de que llegues al kilómetro  $n$ , donde tendrás que ponerlos a prueba...

¿Cuál es la máxima cantidad de cuchillos que puedes coleccionar?

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 10^5$ ), indicando el número de casos de prueba.

Cada caso de prueba contiene los enteros  $n$  ( $1 \leq n \leq 10^7$ ),  $x$  ( $1 \leq x \leq 10^2$ ), e  $y$  ( $1 \leq y \leq 10^2$ ) separados por un espacio.

### Salida

Para cada caso de prueba, imprime el número de cuchillos que podrás coleccionar.

### Ejemplo

Entrada estándar	Salida estándar
3	4
10 3 5	5
10 2 7	0
5 7 11	

## Problema J. The Beatles

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1 segundo
Límite de memoria:	64 megabytes
Autor:	Jared León

Tres escarabajos se encuentran en las coordenadas  $(0,0)$ ,  $(0,1)$ , y  $(1,0)$  del plano  $\mathbb{Z}^2$ . Los escarabajos consideran una configuración estable si cada uno de ellos está a distancia exactamente 1 de otro escarabajo y sus tres posiciones no son colineales (en particular, su posición inicial es estable).

Solamente un escarabajo puede moverse al mismo tiempo y al final de su movimiento, la configuración de estos debe ser estable. ¿Cuál es el mínimo número de movimientos que necesitan para llegar a la configuración de coordenadas  $(x_1, y_1)$ ,  $(x_2, y_2)$ , y  $(x_3, y_3)$ ?

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 10$ ), indicando el número de casos de prueba.

Cada caso de prueba contiene seis enteros  $x_1, y_1, x_2, y_2, x_3, y_3$  ( $|x_i|, |y_i| \leq 25$ ). Se garantiza que la configuración final es estable. Es fácil probar que siempre es posible que los escarabajos lleguen a esta posición.

### Salida

Para cada caso de prueba, calcula el mínimo número de movimientos que necesitan para llegar a su posición final.

### Ejemplo

Entrada estándar	Salida estándar
1 2 2 3 2 2 1	4

## Problema K. Chusky el viajante I

Archivo de entrada:      **Entrada estándar**  
Archivo de salida:        **Salida estándar**  
Límite de tiempo:        1 segundo  
Límite de memoria:      64 megabytes  
Autor:                      Rodolfo Quispe

Chusky quiere viajar por todos los países de Sudamérica. Él no sabe mucho sobre esta parte del mundo y quiere que lo ayudes a planear su viaje. La primera parte es determinar, cuánto tiempo necesitará para visitar todos los países de este continente. Él planea estar 2 meses en cada país. Imprime el número de meses que necesita para viajar por toda Sudamérica.

### Entrada

Este problema no tiene entrada.

### Salida

Imprime la respuesta pedida.

### Ejemplo

Entrada estándar	Salida estándar
	24

## Problema L. Chusky el viajante II

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1 segundo
Límite de memoria:	64 megabytes
Autor:	Rodolfo Quispe

Ahora que Chusky sabe cuántos meses estará en Sudamérica, él necesita calcular cuánto dinero necesitará para su viaje. Él pasará por  $n$  países, en los cuales el cambio de moneda del  $i$ -ésimo país es  $c_i$  por 1 dogcoin (la moneda del país de Chusky), y la cantidad que necesitará en tal país es  $d_i$  en moneda local. Ayuda a chusky a determinar cuánto dinero necesita en dogcoins.

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 100$ ), indicando el número de casos de prueba.

Cada caso de prueba comienza con un entero  $n$  ( $1 \leq n \leq 10^4$ ). Luego, cada una de las siguientes  $n$  líneas contiene los números  $c_i$  ( $1 \leq c_i \leq 10^5$ ) y  $d_i$  ( $0 \leq d_i \leq 10^5$ ) separados por un espacio. Se garantiza que  $d_i$  es un múltiplo de  $c_i$ .

### Salida

Para cada caso de prueba, imprime la cantidad de dinero requerida.

### Ejemplo

Entrada estándar	Salida estándar
2	28
5	101
5 20	
4 20	
10 0	
1 15	
3 12	
2	
100 100	
1 100	

## Problema M. Chusky el viajante III

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1 segundo
Límite de memoria:	64 megabytes
Autor:	Rodolfo Quispe

Chusky comenzó su viaje y está encantado con Sudamérica. Ahora está en Cusco y mientras caminaba por la Plaza de Armas conoció a Jarencio. Intercambiaron números de teléfono y ahora están conversando por WhatsApp. Chusky percibe algo peculiar en Jarencio: cuando él le escribe algún mensaje que contiene alguna vocal, Jarencio no responde e ignora su mensaje. Jarencio es bastante excéntrico y piensa que las vocales no son caracteres cool. Chusky quiere preguntar algo a Jarencio urgentemente pero tiene problemas escribiendo sin vocales. Dado el mensaje original de Chusky, transfórmalo para un mensaje que sea cool para Jarencio.

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 100$ ), indicando el número de casos de prueba.

Cada caso de prueba contiene una cadena de texto  $s$  ( $1 \leq |s| \leq 10^4$ ) en una única línea. La cadena  $s$  puede únicamente contiene letras minúsculas y espacios.

### Salida

Para cada caso de prueba, imprime el texto que Jarencio aprobaría, con las palabras separadas *por un único espacio*.

### Ejemplo

Entrada estándar	Salida estándar
2 quiero que me recomiendes un lugar para ir a comer	qr q m rcmands n lgr pr r cmr

## Problema N. Palíndromo

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	1 segundo
Límite de memoria:	64 megabytes
Autor:	Jared León

Dadas  $n$  cadenas de texto  $s_1, \dots, s_n$ , se quiere formar un palíndromo (una cadena que se escribe de igual de derecha a izquierda como de izquierda a derecha) concatenando dichas cadenas las veces que sea necesario. La única condición es que usar la cadena  $s_i$  tiene un costo de  $c_i$  por uso. Es decir, que el costo de usar la cadena  $s_i$  cinco veces, es  $5c_i$ . Determina el costo mínimo de formar un palíndromo o reporta que es imposible hacerlo.

### Entrada

La primera línea contiene un número entero  $t$  ( $1 \leq t \leq 10$ ), indicando el número de casos de prueba.

La primera línea de cada caso de prueba contiene un entero  $n$  ( $1 \leq n \leq 40$ ). La siguiente línea contiene  $n$  enteros  $c_i$  ( $1 \leq c_i \leq 10^9$ ) separados por un espacio. Cada una de las siguientes  $n$  líneas contiene una cadena de texto  $s_i$  ( $1 \leq |s_i| \leq 20$ ).

### Salida

Para cada caso de prueba, en caso de ser posible construir un palíndromo con las cadenas disponibles, imprime el costo mínimo de hacerlo. De otra forma, imprime el texto "Imposible".

### Ejemplo

Entrada estándar	Salida estándar
2 3 3 4 5 ba abc cbaa 2 1 2 abcd abc	7 Imposible