

Universidad Mayor de San Andrés
Carrera de Informática
International Collegiate Programming Contest
2015



Competencia Interna de Primera División

22 de agosto de 2015

Las páginas están numeradas desde la 1 a la 18.

Problem Setters: Jhonatan Castro (A, C, E, G), Sergio Guillen (C, B, F)
y Edwin Guzman (D).

Chief Judge: Samuel Loza.

Esta página fue intencionalmente dejada en blanco.

Índice

Último Recurso	1
MITM	3
R00tHouse Mesh	5
Panal	9
Chantaje OVNI	11
Chanchullo Reticulado	13
Dead Drop	17

Esta página fue intencionalmente dejada en blanco.

Problema A

Último Recurso

Tiempo Límite: 3 Segundos

Una vez mas es la “Competencia Interna de Primera División del ICPC”, y nuestros héroes de este problemset a quienes irán conociendo mientras lo vayan leyendo, no están preparados para este!. Lo que pasó es que se dedicaron a aprender más sobre seguridad informática y redes, y dejaron de lado los algoritmos, por esa razón es sabido que no tienen ninguna oportunidad el día de hoy en la Competencia de Primera División del Instituto Científico Patito Corcho (ICPC).

Como ellos saben esto, como último recurso decidieron utilizar sus conocimientos en hacking para introducirse en las máquinas de los otros competidores, y apagarlas!. Ellos saben que esto hará que los jueces se molesten y que los expulsen de la competencia, pero como es su último año compitiendo y saben que no tienen chances de clasificar, quieren que su último concurso sea épico y recordado por siempre.. :D

Ahora, la topología de red utilizada en los laboratorios de la carrera de Ciencias de la Computación es bastante extraña ya que les gusta jugar e investigar mucho. Cada computadora está conectada con las cuatro que tiene a su alrededor, una delante, una detras, una al lado izquierdo y otra al lado derecho y cada que un trozo de información pasa entre una de las máquinas, el tiempo que le toma será tomado como “uno”. Existen algunas máquinas lentium bastante antiguas en el laboratorio, a estas el tiempo que les toma pasar la información será tomada como “dos”.

Lo que nuestros amigos juackers intentarán hacer, será enviar un script alrededor del laboratorio, para que pase por las máquinas de sus compañeros que están en la competencia y las apague!. Ellos quieren que este viaje sea el más optimo posible, así que como saben poco de algoritmia, te piden que les ayudes a encontrar el camino más corto que debe recorrer su script si empezará en la computadora de uno de ellos y pasará de máquina en máquina apagando las de sus víctimas para luego volver a la máquina del atacante.

Entrada

La entrada comienza con dos enteros n y m ($1 \leq n, m \leq 10$) que indican la cantidad de filas y columnas de computadoras que tiene el laboratorio. Después de eso debes leer n filas y m columnas de caracteres, de los cuales cada uno representa lo siguiente:

- “X” Indica las computadoras objetivo a las que se debe atacar.
- “x” Es también una computadora objetivo, pero esta es antigua (lentium).
- “#” Es la computadora del atacante, de donde sale el script.
- “O” Indica las computadoras que están siendo utilizadas por otros participantes.

- “o” Es una computadora utilizada por otro participante, pero esta es antigua (lentium).
- “.” Indica que esa computadora está apagada.

Se garantiza que no habrá mas de 16 computadoras objetivo y que siempre existe la posibilidad de atacar a todas. El atacante siempre está en una computadora rápida. Las computadoras ya atacadas se apagan, pero justo antes de apagarse envían el script a travez de la red.

La entrada termina cuando n y m son iguales a cero.

Salida

Por cada caso de entrada, imprimir el tiempo óptimo que le tomará al script viajar desde la máquina del atacante a todas las máquinas víctimas y luego volver.

Ejemplos de entrada	Ejemplos de salida
3 3 #00 oX0 0X. 8 8 0.oX0.o .oX0oX0 o..0o0x ..X0o.0 oX00ox0 00x0o00 0.0Xo0X 0..0o#. 0 0	7 33

Problema B

MITM

Tiempo Límite: 3 Segundos

El sistema de inscripciones del Instituto Científico Patito Corcho (ICPC) ha sufrido una serie de ataques que lograron vulnerar este sistema. Los encargados de seguridad pensando que se robaron las contraseñas de administrador, investigaron un poco y decidieron implementar un nuevo mecanismo de seguridad que prevenga ataques MITM (Man In The Middle).

El ataque MITM funciona de la siguiente forma: Un atacante haciendo uso de ciertas técnicas adquiere la capacidad de leer, insertar y modificar a voluntad, los mensajes entre dos partes sin que ninguna de ellas conozca que el enlace entre ellos ha sido violado. El atacante debe ser capaz de observar e interceptar mensajes entre las dos víctimas (sitios web, mensajes de chat, contraseñas, etc.)

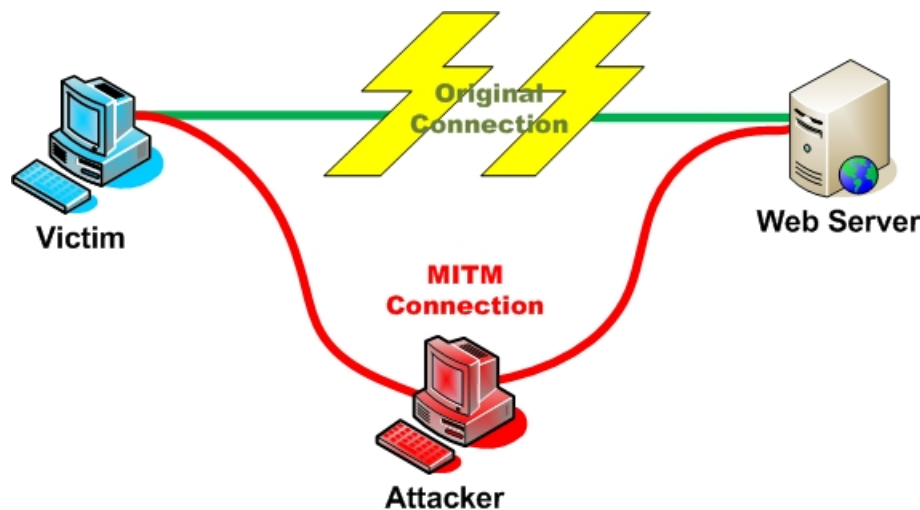


Figura 1. Ilustración del ataque “man-in-the-middle”.
https://www.owasp.org/index.php/Man-in-the-middle_attack

El mecanismo de seguridad que deciden implementar consta de adicionar muchas letras aleatorias al inicio y fin de la contraseña que viajará por la red, los encargados de seguridad lo denominan *Mecanismo Cáscara*. Este nuevo mecanismo enfurece a nuestro buen amigo hacker Alan que decide hacer uso de ingeniería social (su mejor habilidad) para averiguar el funcionamiento de este mecanismo, tras días de intentar obtener esta información Alan se sale con la suya!.

Para poder tener la contraseña Alan deja un script en una computadora dentro del instituto que se ejecutará cada mañana e interceptará el tráfico de red entre la computadora del administrador del sistema y el servidor obteniendo una contraseña (a la que se le aplicó el *Mecanismo Cáscara*) por día.

Después de K días Alan regresa a la computadora donde dejó el script y copia las K contraseñas interceptadas en una memoria flash, al llegar a su casa revisa las contraseñas a las que se les aplicó el *Mecanismo Cáscara* y se sorprende ya que las cadenas son demasiado largas como para analizarlas visualmente. Pero Alan está totalmente seguro que existe solo una cadena que se repite entre las K cadenas.

Lamentablemente las habilidades de programación de Alan no son muy buenas, para ello Alan te busca y te pide que le ayudes a encontrar la contraseña.

Entrada

La entrada consta de T casos de prueba ($1 \leq T \leq 10$). Cada caso de prueba inicia con un número K ($1 \leq K \leq 10$) que indica el número de contraseñas que Alan interceptó. Luego siguen K líneas, cada línea consta de una cadena S_i de caracteres (a-z) donde $1 \leq \sum_{i=1}^K |S_i| \leq 10^5$.

Salida

La salida consiste en una línea por caso que tiene una cadena que representa la contraseña.

Ejemplos de entrada	Ejemplos de salida
2 5 mddqeixxdoklkgramaovxvqvz dghwgrajwjhgramaiokmpp mnvgramak kkkjgramanvnvnbv tmxiopgramakppo 3 fjjcnsjxcrackposka mmcrabxcrackzackddsrufr jjkxdcrackmmkn	grama crack

Problema C

R00tHouse Mesh

Tiempo Límite: 10 Segundos

Una red mesh es un conjunto de routers inalámbricos (wifi) interconectados entre sí inalámbricamente haciendo uso de diferentes protocolos. Esta red puede cubrir grandes áreas como ciudades completas o barrios (haciendo uso de antenas y routers en el techo de edificios) de tal forma que todas las personas conectadas en esta área puedan comunicarse. Una persona puede conectar su celular o laptop a estos routers y podrá comunicarse con todas las personas conectadas dentro del área de la red mesh.

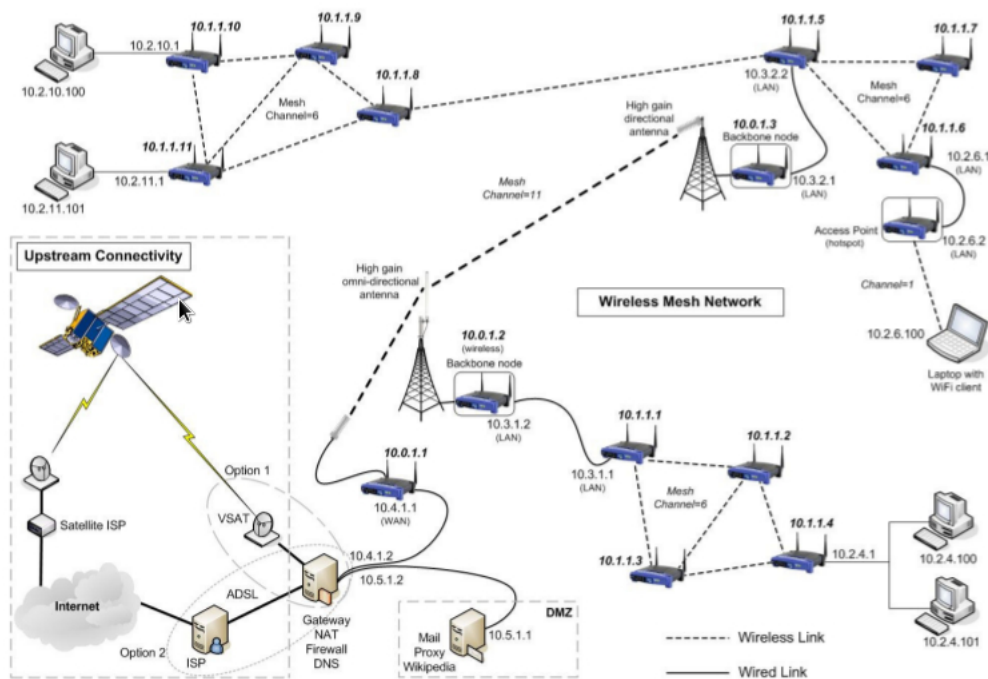


Figura 1. Diagrama de una posible configuración de una red mesh.

Una de las ventajas de este tipo de redes es la capacidad de ser descentralizada. Si existe una gran cantidad de routers conectados y la conexión entre dos de estos cae, esto no debería afectar a la red, es decir, una máquina o un router conectado a la mesh aún debería tener la capacidad comunicarse con cualquier otra máquina o router conectado a la mesh. Por otro lado, si la comunicación entre dos routers cae y esto sí afecta a que a una máquina o router no pueda comunicarse con cualquier otra máquina o router conectado a la mesh, entonces esta red no es descentralizada.

Un grupo de hackers en la ciudad de Río Fugitivo decidió implementar una red mesh. Tras días de trabajo, la red por fin está lista. Con el objetivo de tener una red descentralizada en la

que cualquier persona pueda: conectarse, montar servicios, compartir información, jugar, etc. Para asegurar la descentralización de la red y su calidad, es necesario hacer pruebas cada cierto tiempo. Las pruebas que los hackers hacen son de la siguiente forma.

- Se hace ping entre dos nodos seleccionados aleatoriamente para ver si existe conexión.
- A veces ocurre que algunos nodos de la mesh caen.
- También, otras veces pero lamentablemente con menos frecuencia, nuevos nodos son creados.

Inicialmente los hackers realizaban estas pruebas manualmente, pero con el pasar del tiempo la red mesh creció demasiado, ahora existen demasiados routers conectados! Es por ello que este grupo de hackers necesita automatizar estas pruebas y generar un reporte para ver si la red mesh es descentralizada. Es por ello que debes aplicar tus conocimientos de programación para automatizar este proceso.

Entrada

La entrada consta de T casos de prueba ($1 \leq T \leq 10$). Cada caso de prueba inicia con dos números R y C ($2 \leq R, C \leq 10^5$) que representan la cantidad de routers y la cantidad de enlaces iniciales entre estos dispositivos. Luego debes leer C pares de enteros a, b ($0 \leq a, b < R$) que indican las conexiones iniciales entre los nodos.

A continuación serán introducidos una serie de comandos cuya cantidad no excederá 10^5 , los cuales podrían ser:

- “PING a b ” Este comando indica que se está probando la conexión entre el nodo a y el nodo b .
- “RM c ” Cuando este comando es introducido, significa que el nodo c ya no existe en la mesh.
- “ADD x_1 x_2 x_n ” El comando “ADD” añade un nodo nuevo a la mesh. Al nuevo nodo le corresponde el valor que le continúa a la cantidad de nodos que existen. Por ejemplo, si existen ya n nodos etiquetados del 0 al $n-1$, la etiqueta del nuevo nodo será n . Los valores que preceden al comando son los nodos con los que tiene conexión el nuevo nodo.
- “EXIT” Indica el fin de los comandos.

Lamentablemente la cantidad de nuevos nodos en la mesh es mucho menor a la cantidad de los nodos que desaparecen. Se sabe que la cantidad de comandos “ADD” no superan los 10 en cada caso de prueba.

Salida

Por cada caso de entrada debes imprimir dos líneas con las siguientes oraciones en el idioma inglés.

“X successful tests”

“Y failed tests”

Donde debes reemplazar en “X” la cantidad de pruebas que fueron exitosas, y en “Y” la cantidad de pruebas que fallaron. Es indistinta la frase en caso de que la cantidad de tests exitosos o fallidos sea solo uno.

Ejemplos de entrada	Ejemplos de salida
1 6 7 0 1 1 2 2 3 3 4 2 4 4 5 5 0 PING 0 3 PING 1 4 RM 2 RM 5 PING 0 3 PING 4 3 ADD 0 1 4 PING 1 4 ADD ADD 7 PING 7 1 EXIT	4 successful tests 2 failed tests

Esta página fue intencionalmente dejada en blanco.

Problema D

Panal

Tiempo Límite: 8 Segundos

Las abejas son animales muy interesantes. Estas construyen lindos panales con el fin de tener rápido acceso a todas sus celdas hexagonales. Nos intriga saber el tiempo que le toma visitar a una abeja una lista de celdas.

Una abeja puede caminar por las celdas y recorrer a una celda contigua. Se dice que dada la celda $[X, Y]$ la abeja puede visitar las celdas $[X, Y - 1]$, $[X - 1, Y - 1]$, $[X - 1, Y]$, $[X, Y + 1]$, $[X + 1, Y]$, $[X + 1, Y + 1]$ como se muestra en la Figura 1.

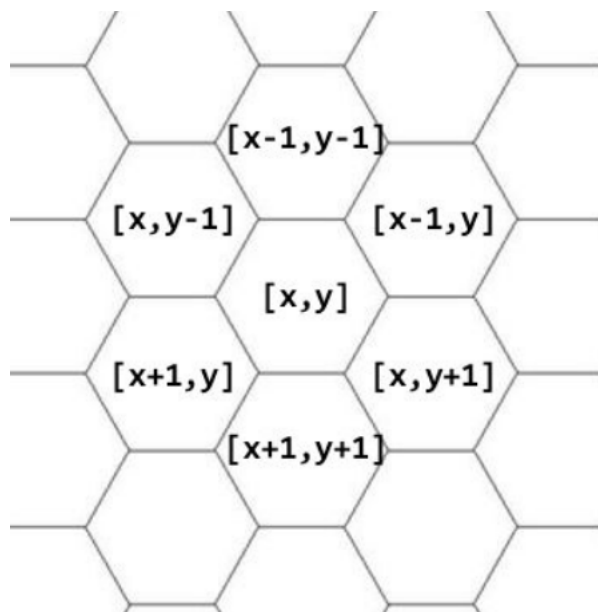


Figura 1. Imagen que muestra un panal y sus coordenadas.

El panal puede ser de tamaño $N * M$ siendo N la cantidad de filas y M la cantidad de columnas. La abeja comienza su recorrido en la coordenada $[0, 0]$.

Entrada

La entrada comienza con un número T que indica la cantidad de casos de prueba.

Cada caso de prueba comienza con tres números enteros N, M y P ($1 \leq N, M, P \leq 50$) donde N y M son las dimensiones del panal, P es la cantidad de celdas que se tiene que visitar.

A continuación está una lista de P pares que representan las coordenadas $[X, Y]$ de todas las celdas que las abejas tienen que visitar, todas las celdas se encuentran en las coordenadas " $0 \leq X < N$ " y " $0 \leq Y < M$ ".

Salida

Por cada caso de prueba, imprima un número entero que representa el tiempo que le toma recorrer la lista de celdas a la abeja.

Ejemplos de entrada	Ejemplos de salida
2	6
4 4 2	118
3 3 0 3	
45 50 4	
14 14 0 12 41 49 0 0	

Problema E

Chantaje OVNI

Tiempo Límite: 3 Segundos

Como todos sabemos, el gobierno nos oculta muchas cosas. Por ejemplo la existencia de vida extraterrestre. La Asociación de Conexiones Marcianas (ACM) ha contactado a nuestros jóvenes Hackers Alan y Botas para ayudarlos en una serie de tareas, he aquí una de ellas.

Los extraterrestres del planeta Marte, resultaron ser chantajistas, ya que amenazan al gobierno con revelar toda la verdad sobre su existencia si ellos no cumplen todas sus demandas.

Una de sus demandas es comprarles los números para las puertas de sus casas, para que estas sean identificadas. Los extraterrestres mandarán al gobierno distintas secuencias de números y el gobierno debe enviar hasta marte todos los dígitos marcianos requeridos para sus puertas.

Por ejemplo, si los marcianos quieren enumerar todas sus casas del 1 al 6, el gobierno debe mandar un dígito uno, un dígito dos, un dígito tres, un dígito cuatro, un dígito cinco y un dígito seis.

Ayuda a Alan y Botas a cumplir esta misión!..

Una observación más, la base numérica que los marcianos utilizan, es la base siete.

Entrada

Cada entrada contiene dos dígitos A y B ($1 \leq A \leq B \leq 10^9$). Estos representan el inicio de la secuencia y el final de la misma en base decimal.

La entrada termina cuando $a = b = 0$.

Salida

Se debe imprimir por cada caso de prueba, 7 números. Los cuales indican la cantidad de dígitos cero, uno, dos, tres, cuatro, cinco y seis que el gobierno necesita para cada petición de los marcianos.

Ejemplos de entrada	Ejemplos de salida
1 7	1 2 1 1 1 1 1
12 321	86 137 142 142 135 136 115
1337 7331	3408 5680 5669 3400 3575 3572 3607
12345678 12345679	2 2 3 3 2 0 6
0 0	

Esta página fue intencionalmente dejada en blanco.

Problema F

Chanchullo Reticulado

Tiempo Límite: 1 Segundo

Durante la historia el hombre siempre ha buscado formas de ocultar mensajes y tener una forma de comunicarse secretamente. El hombre con el tiempo inventa diferentes técnicas desde algunas utilizadas con la tinta en el papel hasta complejos sistemas matemáticos los cuales utilizamos hoy en día.

Una de las técnicas utilizadas hace muchos años se llamaba Cifrado Reticulado. Llamemos clave de cifrado/descifrado a un papel cuadrado con cortes de ventanilla en ciertos cuadrados de la cuadrícula. Este papel siempre será cuadrado, es decir misma cantidad de filas y columnas además de tener un número par de filas y columnas. Las personas que deseaban mantener un mensaje secreto usando esta técnica deben tener esta clave de cifrado/descifrado.

Las ventanillas cortadas de nuestra clave de cifrado/descifrado no serán cortadas arbitrariamente. Las reglas para generar una clave de cifrado/descifrado son las siguientes:

- El papel cuadrado lo dividimos en cuatro regiones. La cantidad de cuadrículas en cada región es la cantidad de cuadrículas que serán cortadas. Ejemplo. Si nuestro papel es de 8×8 cuadrículas en total entonces tendremos cuatro regiones cada una con 16 cuadrículas lo que nos indica que deben ser 16 cuadrículas las que deben ser cortadas.
- Cada región la enumeramos dando vuelta el papel en dirección opuesta a las manillas del reloj. (Ver Figura 1)
- Se deben elegir 16 cuadrículas a cortar en los que no se repita el número elegido. Ejemplo: si se elige en la primera región el número 1 para ser cortado, entonces el número 1 no puede ser cortado en ninguna de las tres regiones restantes. La Figura 2 indica que las casillas pintadas de negro se encuentran recortadas.

1	2	3	4	13	9	5	1
5	6	7	8	14	10	6	2
9	10	11	12	15	11	7	3
13	14	15	16	16	12	8	4
4	8	12	16	16	15	14	13
3	7	11	15	12	11	10	9
2	6	10	14	8	7	6	5
1	5	9	13	4	3	2	1

Figura 1.

	■		■	1		■	
				■			
■		■				■	
2			■				4
	■				■		
■				■			
		■				■	
			■	3			■

Figura 2.

Una vez se tenga la clave de cifrado/descifrado para poder ocultar un mensaje debemos tener otro papel con las mismas longitudes que nuestra clave de cifrado/descifrado. Para ocultar un texto debemos poner la letra de izquierda a derecha del texto en las casillas del nuevo papel donde la clave de cifrado/descifrado estén recortadas (o marcadas con negro), luego dar vueltas la clave en el orden que están marcadas las caras en la Figura 2 y seguir escribiendo las letras hasta completarlas. Por ejemplo se desea ocultar el siguiente mensaje: *suspenda la reunion de los delegados de la region. alguien ha avisado a la poli.* La hoja del texto cifrado (sin incluir espacios, puntos, etc.) se armaría de la siguiente forma: (Ver Figura 3)

	s	u		s	
			p		
e	n			d	
		a			
	l			a	
r			e		
	u			n	
		i			o

	s	n	u		d	s	
			e	p			l
e	o	n			s	d	
d			a	e			l
	l	e			a	g	
r			a	e			
	d	u			o	n	s
d			i				o

e	s	n	u	l	d	s	
	a		e	p	r		l
e	o	n	e		s	d	g
d		i	a	e		o	l
	l	e		n	a	g	
r	a		a	e	l		g
	d	u	u		o	n	s
d	i		i	e		n	o

e	s	n	u	l	d	s	h
a	a	a	e	p	r	v	l
e	o	n	e	i	s	d	g
d	s	i	a	e	a	o	l
d	l	e	o	n	a	g	a
r	a	l	a	e	l	a	g
p	d	u	u	o	o	n	s
d	i	l	i	e	i	n	o

Figura 3.

Pero bueno, he aquí la historia de nuestros dos amigos hackers Alan y Botas que encontraron esta técnica navegando por la red y la utilizaron para hacer travesuras en el Instituto Científico Patito Corcho (ICPC). Nuestros amigos hackers tenían la misma clave de cifrado/descifrado y la utilizaban para pasarse las respuestas en los exámenes de la materia de *Historia de las Computadoras*. Por mala suerte el profesor de esta materia descubrió que Alan y Botas se pasaban un papelito durante el examen. El contenido de este era irreconocible para el profesor lo que lo enfureció más y anuló el examen de nuestros amigos hackers.

El profesor obstinado con saber cual era el contenido de este papelito se pasó noches tratando de descifrarlo. Rendido tras no haber conseguido descifrar el contenido habló con nuestros amigos hackers Alan y Botas ofreciéndoles una segunda oportunidad para reemplazar el examen anulado a cambio de que nuestros amigos le digan que técnica utilizaron para cifrar sus trampas. Pues bien, nuestros dos amigos hackers fueron generosos le dijeron que era la técnica de *Cifrado Reticulado* además de darle la clave de cifrado/descifrado.

Con mucho entusiasmo el profesor fue a su casa a investigar esta técnica y se le ocurrió la idea de tener un programa que dada la clave de cifrado/descifrado y un texto cifrado se pueda descifrar y ver el contenido real del texto. Lamentablemente el profesor había olvidado los pocos conocimientos de programación que tenía, pero recordó que tú eras uno de sus estudiantes y te pide ayuda con este inconveniente que tiene.

Ayuda al profesor a descifrar un texto dada la clave de cifrado/descifrado.

Entrada

La entrada consta de T casos de prueba ($1 \leq T \leq 100$). Cada caso de prueba inicia con un número N ($1 \leq N \leq 50$, N par) que representa el tamaño de la matriz que representa la clave de cifrado/descifrado. Luego siguen N cadenas cada una con N caracteres (0 o 1) donde 1 representa una casilla marcada y el 0 una casilla sin marcar (la matriz de cifrado/descifrado que

cumplen las reglas descritas al inicio del problema). Las siguiente N líneas forman el mensaje a descifrar, las N cadenas S_i ($1 \leq |S| \leq N$) solo cuentan con los caracteres de la a hasta la z

Salida

La salida consiste en una línea por caso que representa el texto descifrado. No es necesario incluir símbolos de separación como saltos de línea, espacios, tabulación, etc. (El ejemplo de salida de abajo debe ser impreso en una sola línea)

Ejemplos de entrada	Ejemplos de salida
1 8 01010010 00001000 10100010 00010000 01000100 10001000 00100010 00010001 esnuldsh aaaeprvl eoneisdg dsiaeaol dleonaga ralaelag pduuoons dilieino	suspendalareuniondelosdelegadosde-> laregionalguienhaavisadoalapoli

Esta página fue intencionalmente dejada en blanco.

Problema G

Dead Drop

Tiempo Límite: 1 Segundo

Un USB Dead Drop es un dispositivo USB instalado en un espacio público. Por ejemplo una memoria flash USB podría estar montada en un muro y fijada con concreto. El nombre viene de la técnica “Dead Drop” de espionaje de comunicaciones. Estos dispositivos fueron colocados por todo el mundo por los hacktivistas.



Figura 1. Deaddrop colocado en un árbol.

Los hackers Alan y Botas están a punto de ejecutar una operación muy importante en su universidad “Instituto Científico Patito Corcho” (ICPC) que se encuentra en la ciudad de Río Fugitivo. El edificio principal del ICPC (también conocido como Monoblock) tiene jardines a sus costados, en los cuales los árboles están colocados de una forma muy característica gracias a la gente de la Carrera de Ciencias de la Computación. En cada uno de estos árboles se encuentran Dead Drops y Alan y Botas deben recuperar toda la información de estos para que su operación secreta tenga éxito.

La forma en que los árboles están plantados en el jardín es la de un árbol binario perfecto (ver la Figura 2), en las cuales los árboles son los nodos y los caminos del jardín los arcos.

Se te da el entero N que indica la altura de el árbol (La altura de un árbol binario perfecto es el número de arcos en el camino entre el nodo raíz y alguno de los nodos hoja). Por lo tanto hay $2^{N+1} - 1$ árboles con Dead Drops y $2^{N+1} - 2$ caminos en el jardín en total.

La imagen de abajo muestra cómo luce el jardín cuando $N = 2$.

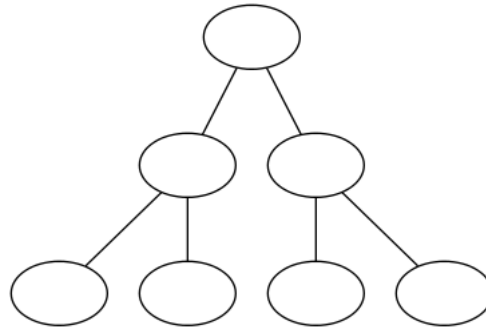


Figura 2. Ejemplo de un árbol binario perfecto de altura 2.

Alan y Botas son unos buenos nerds, y por esta razón quieren minimizar la cantidad de personas que irán a recoger la información de los Dead Drops. Cada persona que recogerá la información de un Dead Drop, visitará un árbol de inicio y un árbol fin sin visitar un árbol dos veces. (Note que la ruta de cada persona está únicamente determinada por el árbol de inicio y final). Es probable que el árbol de inicio sea el mismo que el árbol final, en este caso la persona solo visitará un árbol.

Ayuda a nuestros hackers a calcular la mínima cantidad de personas que necesitan si cada árbol solo puede ser visitado por una persona.

Entrada

Primero debes leer un entero t ($1 \leq t \leq 10^4$) que indica la cantidad de casos de entrada. Luego por cada caso de entrada debes leer un N ($0 \leq N \leq 60$).

Salida

Por cada entrada imprimir la cantidad de personas que se necesitarán para recorrer por todos los árboles del jardín sin que dos personas pasen por el mismo árbol.

Ejemplos de entrada	Ejemplos de salida
3	1
1	5
3	683
10	