**Menú de pastes**

Tiempo límite por caso: 1 seg

Memoria límite por caso: 256MB

En los días previos a la competencia, en el COMI quisimos probar los tradicionales pastes. Sin dudarlo entramos en una sucursal de pastes Kikos y pedimos un menú. El menú era una cuadrícula de ***m*** filas por ***n*** columnas con un sabor de paste en cada cuadro.

Siempre buscando sistematizar y optimizar, numeramos los sabores de la siguiente forma:

* Los sabores de la fila superior se numeraron del *1* al ***n*** comenzando por la izquierda
* Los de la siguiente fila hacia abajo se numeraron del ***n****+1* al *2****n*** comenzando, de nuevo, por la izquierda
* Y así con cada fila hasta que los sabores de la fila inferior quedaron con los números del ***n(m-1)+1*** a ***nm***

Los miembros del COMI pidieron entonces ***p*** pastes diciendo el número de sabor que querían.

Desgraciadamente, al tomar la orden giraron el menú 90 grados en el sentido de las manecillas del reloj cambiando así los sabores que recibió cada miembro del COMI ***(para que te quede más claro, revisa la explicación del ejemplo)***.

**Problema**

Escribe un programa que dadas las dimensiones del menú ***m, n,*** el número ***p*** de pastes que se pidieron y la lista de números de sabores pedidos entregue como resultado los números de sabor que realmente se recibieron en el mismo orden en que fueron pedidos.

**Entrada**

Tu programa debe leer del teclado la siguiente información:

* En la primera línea los números ***m***, ***n***, las dimensiones del menú.
* En la segunda línea el número ***p*** de pastes a pedir.
* En la tercera línea ***p*** números enteros separados por un espacio que representan los pastes que se ordenaron.

**Salida**

Tu programa debe escribir en la pantalla ***p*** números enteros separados por un espacio que representan los números de paste que se recibieron después de que se giró el menú.

**Restricciones**

**1 <= *m, n* <= 2,000,000** Dimensiones del menú

**1 <= *p* <= 1,000** Número de pastes que se pidieron.

**Ejemplo**

|  |  |  |
| --- | --- | --- |
| **Entrada**  | **Salida**  | **Explicación** |
| 2 351 3 5 6 1 | 4 5 6 3 4 | El menú tiene una dimensión de ***m=2*** filas por ***n=3*** columnas. Según el método de numeración los sabores se numeraron así:

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |

Y los miembros del COMI pidieron los sabores ***1, 3, 5, 6, 1***Al girar el menú 90 grados quedó de la siguiente forma

|  |  |
| --- | --- |
| 4 | 1 |
| 5 | 2 |
| 6 | 3 |

Por lo que ahora al servir un paste 1 en realidad se está sirviendo un paste 4 del menú original.  |

**Evaluación**

Para un conjunto de casos de prueba con valor de **47** puntos, ***n, m*** <= 1,000.

Para otro conjunto ***agrupado*** de casos de prueba con valor de **53** puntos, ***n, m*** <= 2,000,000.

¿Y mis 50 mil puntos qué?

*Recuerda que los compiladores no te permiten definir una matriz de 2 millones x 2 millones de enteros. Si tu solución para el problema menú de pastes es utilizar una matriz, sólo funcionará para los primeros 47 puntos en dónde m y n son menores o iguales a 1,000, por lo que de ser así, te recomendamos que envíes tu código definiendo matrices de tamaño 1,000 x 1,000.*

**Granjas**

Tiempo límite por caso: 1seg

 Memoria límite por caso: 16MB

La industria de los pastes en Pachuca demanda una gran cantidad de ingredientes. En total se necesitan ***N*** ingredientes distintos de origen agrícola. En los alrededores de Pachuca hay ***N*** granjas, cada una de estas granjas planta uno de los ingredientes.

Las granjas cambian cada año de ingrediente para optimizar el uso del suelo. Para asegurarse de que ningún ingrediente quedará desabastecido las granjas se organizaron de la siguiente forma:

* Se numeraron las granjas del ***1*** al ***N***.
* Llenaron un arreglo ***A*** con los números del ***1*** al ***N*** acomodados de manera aleatoria.
* Cada cambio de año la granja ***i*** plantará el ingrediente que plantó en el año anterior la granja ***A[i].***

**Problema**

Escribe un programa que recibiendo el número ***N*** de granjas/ingredientes y el arreglo ***A*** determine ¿cuántos años pasarán antes de que todas las granjas vuelvan a plantar exactamente los mismos ingredientes que plantaron este año?

**Entrada**

Tu programa debe leer del teclado los siguientes datos:

* La primera línea contiene el entero ***N***, la cantidad de granjas/ingredientes.
* En la segunda línea ***N*** números enteros separados por un espacio que representan los valores del arreglo ***A.***

**Salida**

Tu programa debe escribir en la pantalla un único número que indica la cantidad de años que pasarán antes de que todas las granjas vuelvan a plantar exactamente los mismos ingredientes que plantaron este año.

**Restricciones**

**1 <= *N* <= 50,000** Cantidad de granjas/ingredientes

**Ejemplo**

|  |  |  |
| --- | --- | --- |
| **Entrada**  | **Salida**  | **Explicación** |
| 53 4 5 1 2 | 5 | Si este año la granja 1 plantó el ingrediente ***1***, la granja 2 el ingrediente ***2***, la 3 el ingrediente ***3,*** la 4 el ***4***, y la 5 el ***5***.Para el segundo año, de acuerdo con el arreglo la granja 1 plantará lo que el año anterior había plantado la granja 3, ya que ***A[1]=3***, la granja 2 plantará lo que había plantado la granja 4, etc.El segundo año entonces las granjas plantarán: 3, 4, 5, 1, 2El tercer año las granjas plantarán: 5, 1, 2, 3, 4El cuarto año plantarán: 2, 3, 4, 5, 1El quinto: 4, 5, 1, 2, 3El sexto: 1, 2, 3, 4, 5 **(Exactamente lo mismo que el primer año)****Por lo tanto al pasar 5 años, las granjas volverán a plantar exactamente los mismos ingredientes que el primer año.** |

**Evaluación**

Para un conjunto de casos ***sin agrupar*** con valor de **33** puntos el número máximo de años que pasarán es de 1,000.

Para otro conjunto de casos ***agrupados*** con valor de **31** puntos el número máximo de años que pasarán es de 100,000,000.

Para un último conjunto de casos ***agrupados*** con valor de **36** puntos el número máximo de años que pasarán cabe en un entero signado de 64 bits.

Envié 100 problemas de a 500… ¿'on 'tan?

**Nieves Pizzero**

Tiempo límite por caso: 1seg

Memoria límite por caso: 64MB

Nieves, un famoso integrante del COMI se ha aficionado a cocinar pizzas. Lo que diferencia a Nieves de cualquier otro pizzero es que Nieves puede cocinar pizzas de cualquier diámetro. Por si fuera poco, Nieves está haciendo su maestría en visión computacional.

Nieves cocina pizzas una por una, cada que termina una pizza, Nieves la coloca encima de las pizzas anteriores de modo que su centro quede alineado con los centros de todas las pizzas anteriores. Cuando alguien le pide una de las pizzas de la pila, Nieves toma esa pizza y tira a la basura todas las pizzas que estaban arriba de ella.

Como proyecto de maestría Nieves ha desarrollado un sistema de visión computacional para supervisar sus pizzas, dicho sistema funciona de la siguiente manera: hay una cámara colocada en el techo, justo encima del centro de la pila de pizzas. En cualquier momento, Nieves puede solicitarle a su sistema que le diga ¿cuántas pizzas ve? Dado que la cámara está arriba las pizzas de mayor diámetro *tapan* a las pizzas pequeñas que están debajo de ellas en la pila. Si hay dos pizzas del mismo diámetro juntas, la cámara sólo ve la que esté más arriba.

Nieves desea que le ayudes a comprobar si su sistema funciona correctamente.

**Problema**

Escribe un programa que sea capaz de procesar 3 tipos de comandos:

* Comando ***0***: Al recibir este comando tu programa deberá contestar cuántas pizzas ve la cámara en ese momento
* Comando ***1 d***: Este comando le indica a tu programa que Nieves apiló una nueva pizza de diámetro ***d***
* Comando ***2 q***: Este comando le indica a tu programa que Nieves quitó ***q*** pizzas de la pila. *Nieves nunca quita más pizzas de las que hay en la pila en un momento dado.*

**Entrada**

Tu programa debe leer del teclado la siguiente información:

* En la primer línea el número ***N*** la cantidad de comandos que recibirá tu programa.
* En las siguientes ***N*** líneas puede haber 1 ó 2 números. El primer número será un 0, 1 ó 2 e indicará el tipo del comando. En el caso de los comandos 1 y 2 habrá un segundo número que indicael valor de ***d*** o ***q*** respectivamente.

**Salida**

Por cada comando del tipo 0 que haya en la entrada tu programa deberá escribir una línea con un número que indique la cantidad de pizzas que veía la cámara en ese momento.

**Restricciones**

**1 <= *N* <= 500,000** La cantidad de comandos a ejecutar.

**1 <= *q* <= *N*** El número de pizzas que se quitan de una pila en un comando.

**1 <= *d*** **<= 1,000,000,000** Diámetro de una pizza.

**Ejemplo**

|  |  |  |
| --- | --- | --- |
| **Entrada**  | **Salida**  | **Explicación** |
| 901 31 101 401 22 20 | 0212 | En la entrada hay 4 comandos del tipo 0.El primer comando tipo 0 llega cuando no hay pizzas en la pila por lo que la respuesta es ***0***Después se apilan dos pizzas una de diámetro 3 y encima una de diámetro 1, la de diámetro 1 no tapa a la de diámetro 3.En el segundo comando tipo 0 la cámara ve ***2*** pizzas.Posteriormente se apila una pizza de diámetro 4 que al ser mayor tapa todas las anteriores.Con el tercer comando tipo 0 la cámara ve sólo ***1*** pizza.Finalmente se quitan dos pizzas, las dos pizzas de arriba son las de diámetro 2 y 4 respectivamente, al quitarlas quedan únicamente la pizzas de diámetro 3 y 1. Cuando llega el último comando tipo 0, la cámara ve sólo 2 pizzas. |

**Evaluación**

Para un grupo de casos de prueba con valor de **11** puntos los diámetros de las pizzas serán incrementales.

Para un grupo de casos de prueba con valor de **17** puntos, ***N*** <= 5,000.

Para un grupo de casos de prueba con valor de **29** puntos no existe el comando quitar.

Para un grupo de casos de prueba con valor de **43** puntos, ***N*** <= 500,000 y ***d*** < 1,000,000,000.

Ahí véale… reevalúele bien.

**Freddy, el amante de los hot-cakes**

Tiempo límite por caso: 1seg

 Memoria límite por caso: 128MB

A diferencia de Nieves, Freddy es un experto preparando hot-cakes. Es tan hábil que puede preparar ***N*** hot-cakes de manera simultánea. Para hacer esto Freddy inicia con una pila de ***N*** hot-cakes y los numera del ***1*** al ***N***, los coloca sobre el sartén, el número ***1*** hasta abajo y el ***N*** hasta arriba y a partir de ahí realiza ***V*** *volteos.* Un *volteo* consiste en lo siguiente:

* Freddy elige un número ***a*** entre ***1*** y ***N***.
* Entonces toma la pila de hot-cakes desde el hot-cake que está en la posición ***a*** hasta el hot-cake ***N*** y los voltea, de modo que el hot-cake que estaba hasta arriba queda en la posición ***a*** y el hot-cake que estaba en la posición ***a*** queda hasta arriba.
* Después de ***V*** *volteos*, los hot-cakes están listos.

**Problema**

Escribe un programa dado ***N, V*** y los números ***ai*** que eligió Freddy, determine en qué posición quedó cada uno de los ***N*** hot-cakes.

**Entrada**

Tu programa debe leer del teclado los siguientes datos:

* La primera línea contiene los enteros ***N, V*** la cantidad de hot-cakes y el número de volteos que se harán.
* La siguiente línea contiene ***V*** números separados por un entero que indican los índices ***ai*** que elige Freddy.

**Salida**

Tu programa deberá escribir una línea con ***N*** números enteros separados por un espacio cada uno que indiquen el número de hot-cake que quedó en cada posición de la pila después de los ***V*** volteos.

**Restricciones**

**1 < *N* <= 1,000,000** La cantidad de hot-cakes.

**1 < *V* <= 30,000** El número de volteos que se harán.

**Ejemplo**

|  |  |  |
| --- | --- | --- |
| **Entrada**  | **Salida**  | **Explicación** |
| 10 3462 | 1 8 7 6 5 4 9 10 3 2 | Inicialmente los hot-cakes inician ordenados, es decir 1,2,3,4,5,6,7,8,9,10. Para el primer volteo, Freddy elige el índice 4. Al hacer el primer volteo tenemos que los hot-cakes quedan en el siguiente orden: 1,2,3,10,9,8,7,6,5,4.Para el segundo volteo se elige el índice 6, al hacerlo los hot-cakes quedan: 1,2,3,10,9,4,5,6,7,8.Para el último volteo se elige el índice 2, al hacerlo los hot-cakes quedan: 1,8,7,6,5,4,9,10,3,2. |

**Evaluación**

Para un grupo de casos de prueba con valor de **30** puntos, ***N*** <= 1,000 y ***V*** <= 1,000.

Para un grupo de casos de prueba con valor de **33** puntos, ***N*** <= 1,000,000 y ***V*** <= 1,000.

Para un grupo de casos de prueba con valor de **37** puntos, ***N*** <= 1,000,000 y ***V*** <= 30,000.

¡Me evaluaron como PUERCO!