

1. Se cuenta con un arreglo de tamaño “n” ($n \leq 10$) cuyos elementos son números enteros positivos de 1 o 2 cifras, todos diferentes. Escriba un programa que reciba el arreglo y mediante una función que recibe el tamaño n y el arreglo, muestre las sumas que pueden ser obtenidas con los subconjuntos formados por dichos números. Ejemplo: si $n=4$ y los valores son 1, 2, 5 y 10

Entonces las sumas son 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17 y 18

Sugerencia: si el arreglo contiene n valores entonces puede utilizar números binarios de n cifras para encontrar todos los subconjuntos

0001	10
0010	5
0100	2
1000	1
0011	5 + 10
0110	2 + 5
...	
1001	1 + 10
1011	1 + 5 + 10
1111	1 + 2 + 5 + 10

2. Escriba un algoritmo que dados 2 números enteros x y n donde $0 < n < 25$ determine el valor de la siguiente serie:

$$S = \frac{x^{(2n+1)}}{3!} - \frac{x^{(2n-1)}}{5!} + \frac{x^{(2n-3)}}{7!} - \dots,$$

Considera que no existen operadores ni funciones de potencia ni de factorial.

3. Un tablero “**n-diagonal**” (o también conocido como “Tablero Diagonal Cíclico”) es un tablero con $n \times (n + 1)$ casillas. Nos referiremos a las casillas de este tablero mediante las coordenadas (i, j), en donde i es la fila (contadas desde 1 y de arriba a abajo) y j es la columna (contadas desde 1 y de izquierda a derecha). Como ejemplo:

	1	2	...	n	n+1
1			...		
2			...		
⋮	⋮	⋮	⋱	⋮	⋮
n			...		

Una propiedad interesante es que se pueden visitar todas sus casillas siguiendo un patrón diagonal que abarca todo el tablero de manera cíclica. El recorrido comienza en la casilla (1,1), avanzando hacia la derecha y hacia abajo a lo largo de la diagonal principal. Cuando se sobrepasa el límite del tablero, ya sea por la derecha o por abajo, el recorrido continúa desde la primera fila o la primera columna, respectivamente. Este proceso se repite hasta que todas las casillas han sido visitadas. El recorrido asegura que, tras alcanzar una casilla límite, se salta de vuelta a la fila o columna inicial.

Como ejemplo este tablero de 4 x 5 casillas, rellenas con un número que indica en qué momento se visitó:

1	17	13	9	5
6	2	18	14	10
11	7	3	19	15
16	12	8	4	20

Haz un programa que dado un número “n” ($n < 10$) muestre en pantalla:

- Un tablero “n-diagonal” con sus casillas rellenas con el número correspondiente al momento en que se visitan.
- La matriz cuadrada resultante de eliminar el menor elemento de cada fila del tablero “n-diagonal”.

- Diseña un algoritmo que permita resolver el siguiente problema:
El alcalde de la ciudad contrató a Bob el constructor para que tape los agujeros que hay en una carretera. La carretera tiene un largo **N** y un ancho **M**. Bob dividió esta carretera en cuadrados tal que el resultado fue una rejilla de **N x M**.

Ayuda a Bob a saber cuántos huecos tendrá que tapar, un hueco es un conjunto de cuadrados imperfectos conectados, se dice que dos cuadros o casillas están conectadas si comparten un lado en común. (Ver el ejemplo).

Entrada

N, **M**, el largo y ancho de la carretera.

Seguidamente de **N** filas cada una con **M** caracteres. Si el carácter es 'X' significa que la sección está bien. Si el carácter es 'O' significa que hay un problema en la sección.

Salida

La cantidad de huecos que tendrá que tapar Bob el constructor.

Restricciones

$1 \leq N, M \leq 100$

Ejemplo

Entrada

6 10

X	X	X	X	X	X	X	X	X	X
X	O	X	O	O	O	X	X	X	X
X	O	X	O	O	O	X	X	X	O
X	O	O	X	X	X	X	O	X	O
X	X	X	X	X	O	O	O	X	X
O	O	O	O	O	O	X	X	X	X

Salida

4 huecos

Huecos en el ejemplo

X	X	X	X	X	X	X	X	X	X
X	0	X	0	0	0	X	X	X	X
X	0	X	0	0	0	X	X	X	0
X	0	0	X	X	X	X	0	X	0
X	X	X	X	X	0	0	0	X	X
0	0	0	0	0	0	X	X	X	X

5. Dados dos conjuntos A y B de m y n elementos enteros, se define la distancia entre los conjuntos A y B como el número de elementos de la diferencia simétrica de dichos conjuntos con sus elementos indexados.

Ejemplo:

Conjunto A : $\{1; 3; 5; 7\}$

Conjunto A con sus elementos indexado: $A_{Ind} = \{1_1; 3_2; 5_3; 7_4\}$

Conjunto B : $\{1; 2; 4; 5\}$

Conjunto B con sus elementos indexado $B_{Ind} = \{1_1; 2_2; 4_3; 5_4\}$

Diferencia simétrica de A_{Ind} y B_{Ind} : $(A_{Ind} - B_{Ind}) \cup (B_{Ind} - A_{Ind}) = \{2_2; 3_2; 4_3; 5_3; 5_4; 7_4\}$

La distancia entre A y B es: 6

Nota: Validar que los elementos de un conjunto no se repitan.

6. Desarrolle un programa que reciba por consola un texto que representa a un matriz de 9×9 que representa a un SUDOKU, donde los ceros representan a los casilleros vacíos. Su programa debe determinar para cada casillero vacío la lista de valores posibles. Ejemplo: si se ingresa la siguiente matriz:

Entrada

530070000
600195000
098000060
800060003
400803001
700020006
060000280
000419005
000080079

Representa:

	1	2	3	4	5	6	7	8	9
1	5	3			7				
2	6			1	9	5			
3		9	8					6	
4	8				6				3
5	4			8		3			1
6	7				2				6
7		6					2	8	
8				4	1	9			5
9					8			7	9

Salida (solo se muestra algunos ejemplos de la salida):

Casillero (2,2) valores posibles: 2, 4, 7

Casillero (4,7) valores posibles: 4, 5, 7, 8, 9

Nota: Recordar que no puede haber valores repetidos en las filas, ni en las columnas, ni en los recuadros de 3 x 3

7. Diseñe una función que reciba una oración, de no más de 200 caracteres, que contiene tanto palabras como números enteros de hasta 3 cifras. Asuma que las palabras no contienen dígitos numéricos, ni letras tildadas ni caracteres especiales, ni signos de puntuación. Luego ejecute lo siguiente:

Calcule la suma de los números enteros en dicha oración (s).

Obtenga el resto (r) de la división entera entre dicha suma (s) y el número 5. Luego a dicho resto (r) le sume 1 y obtenga un nuevo número (n).

Luego determine cuantas palabras en la oración tienen "n" vocales diferentes. Debe imprimir en pantalla un mensaje indicando cuantas palabras tienen "n" vocales diferentes y la función deber retornar la cantidad de palabras calculadas.

Ejemplo:

Se recibe la siguiente oración:

Mis primos tienen casi 40 años yo tengo 34 y tenemos que jugar pelota con gente de 18 esta bien difícil ganar

Debe retornar el número 2 (que indica que hay 2 palabras con 3 vocales diferentes)

El programa principal debe imprimir **Hay 2 palabras con 3 vocales diferentes**

Nota: $s = 40 + 34 + 18 = 92$; $r = 92 \bmod 5 = 2$; $n = r + 1 = 3$;

8. Sergei ha diseñado una nueva línea de juguetes anidados: “babushkas generalizadas” (en adelante BG), inspirados en las tradicionales babushkas rusas (una babushka es una muñeca que puede abrirse en dos mitades, de modo que se encuentra otra muñeca dentro, la misma que puede tener otra muñeca dentro y así seguir varias veces, hasta llegar a una muñeca final que no puede abrirse). En estas BG, cada juguete puede contener varios juguetes (no sólo 1) más pequeños dentro de él.

Vladimir ha desarrollado una notación para describir cómo deben construirse los juguetes anidados. Un juguete se representa con un número entero positivo, según su tamaño. Más concretamente: si al abrir el juguete representado por m encontramos los juguetes representados por n_1, n_2, \dots, n_r , debe ser cierto que $n_1, n_2, \dots, n_r < m$. Y si esto es así, decimos que el juguete m contiene directamente a los juguetes n_1, n_2, \dots, n_r (ojo: cualquier juguete que pueda estar contenidos en cualquiera de los juguetes n_1, n_2, \dots, n_r no se consideran directamente contenido en el juguete m).

Una babushka generalizada (BG) se denota con una secuencia no vacía de enteros no nulos de la forma:

$$a_1, a_2, \dots, a_N$$

tal que el juguete k se representa en la secuencia con dos enteros $-k$ y k , ubicándose el negativo antes que el positivo.

Por ejemplo, la secuencia:

$$-9 \ -7 \ -2 \ 2 \ -3 \ -2 \ -1 \ 1 \ 2 \ 3 \ 7 \ 9$$

representa una BG formada por seis juguetes, a saber, 1, 2 (dos veces), 3, 7 y 9.

Obsérvese que el juguete 7 contiene directamente los juguetes 2 y 3. Obsérvese que la primera copia del juguete 2 se produce a la izquierda de la segunda y que la segunda copia contiene directamente al juguete 1. Sería erróneo entender que el primer -2 y el último 2 deberían estar emparejados.

Por otro lado, las siguientes secuencias no describen babushkas generalizadas:

$$-9 \ -7 \ -2 \ 2 \ -3 \ -1 \ -2 \ 2 \ 1 \ 3 \ 7 \ 9$$

porque el juguete 2 es mayor que el juguete 1 y no puede asignarse dentro de él.

$$-9 \ -7 \ -2 \ 2 \ -3 \ -2 \ -1 \ 1 \ 2 \ 3 \ 7 \ -2 \ 2 \ 9$$

porque 7 y 2 no pueden asignarse juntos dentro de 9.

Tu tarea es escribir un programa que determine si una secuencia leída representa una babushka generalizada válida.

Entrada

La primera línea constará de un entero “l” que indicará la cantidad de líneas a leer ($l < 20$)

Cada caso de prueba es una línea de enteros no nulos de 3 cifras o menos.

Salida

Por cada línea, imprime “: -) babushkas!” si es válida o “:-(intenta de nuevo.” si no lo es.

Ejemplo de Entrada

2

-9 -7 -2 2 -3 -2 -1 1 2 3 7 9

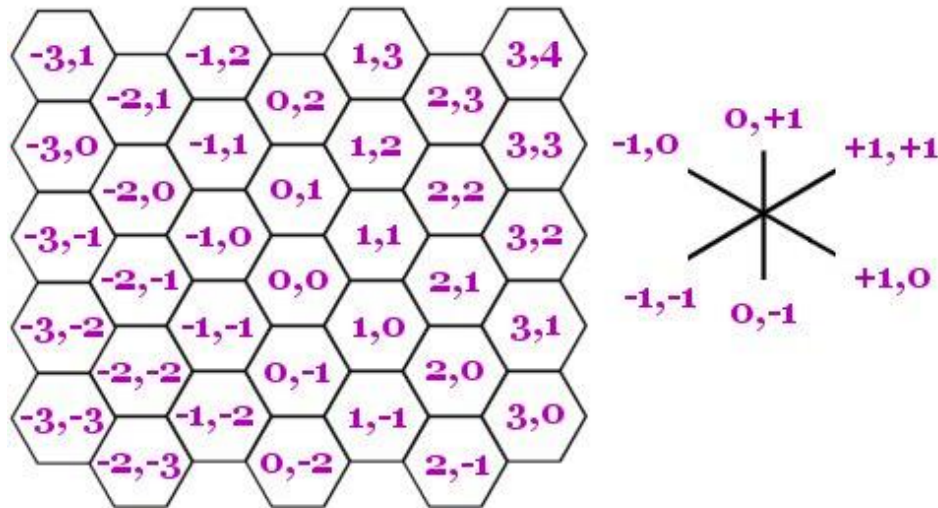
-9 -7 -2 2 -3 -1 -2 2 1 3 7 9

Ejemplo de Salida

: -) babushkas!

:-(intenta de nuevo.

9. Una abeja muy organizada ha numerado las celdas hexagonales de su – potencialmente infinito – panal de abeja en la forma indicada en la figura 2. Luego, la abeja se pregunta: ¿cuántas celdas deben ser recorridas para llegar desde la celda con numeración (x, y) hasta la celda con numeración (u, v) ?



Sistema de coordenadas hexagonales

Desarrolle un programa que lea 4 enteros x, y, u, v , en el rango $[-1000, 1000]$ e imprima la cantidad de celdas que deben ser recorridas para llegar desde la celda (x, y) hasta la celda (u, v) del panal de abeja (incluyendo las celdas inicial y final).

Entrada	Salida
0 0 1 1	2
1 0 3 3	4
2 1 2 1	1

10. Nuestro mundo ha sido invadido por extraterrestres que cambian de forma y secuestran personas y roban sus identidades. Eres un inspector de un grupo de trabajo dedicado a detectarlos y capturarlos. Como tal, se te proporcionaron herramientas especiales para detectar extraterrestres y diferenciarlos de los humanos reales. Tu misión actual es visitar una ciudad que se sospecha que ha sido invadida, inspeccionar en secreto a todas las personas allí para saber quiénes son extraterrestres y quienes no, e informarlo todo al Cuartel General. Luego podrían enviar fuerzas a la ciudad por sorpresa y capturar a todos los alienígenas a la vez.

Los extraterrestres conocen el trabajo de inspectores como usted y están monitoreando todos los canales de radio para detectar la transmisión de dichos informes, con el fin de anticipar cualquier represalia. Por lo tanto, se han realizado varios esfuerzos para cifrar los informes y el método más reciente utiliza polinomios. La ciudad que debes visitar tiene N ciudadanos, cada uno identificado por un número entero par distinto de 2 a $2N$. Desea encontrar un polinomio P tal que, para cada ciudadano i , $P(i) > 0$ si el ciudadano i es un ser humano, y $P(i) < 0$ en caso contrario. Este polinomio será transmitido a la sede. Con el objetivo de minimizar el ancho de banda, el polinomio tiene algunos requisitos adicionales: cada raíz y coeficiente debe ser un número entero, el coeficiente de su término de mayor grado debe ser 1 o -1, y su grado debe ser el más bajo posible.

Para cada ciudadano, sabes si es humano o no. Dada esta información, debes encontrar un polinomio que satisfaga las restricciones descritas.

Entrada

La entrada consta de una sola línea que contiene una cadena S de longitud N ($1 \leq N \leq 104$), donde N es la población de la ciudad. Para $i = 1, 2, \dots, N$, el i -ésimo carácter de S es la letra mayúscula "H" o la letra mayúscula "A", lo que indica respectivamente que el ciudadano $2i$ es un humano o un extraterrestre.

caso 1

HHH

caso 2

AHHA

caso 3

AHHHAH

Salida

La primera línea debe contener un número entero D que indique el grado de un polinomio que satisface las restricciones descritas. La segunda línea debe contener $D + 1$ números enteros que representen los coeficientes del polinomio, en orden decreciente de los términos correspondientes. Se garantiza que existe al menos una solución tal que el valor absoluto de cada coeficiente sea menor que 263.

caso 1

0 1

caso 2

2-

1 10 -21

caso 3

3

1 -23 159 -297

Consideraciones:

- La tarea es grupal (grupo de 4 o 5 personas)
- Cada pregunta vale 2 puntos.
- La fecha límite de entrega es el **sábado 23/06/2024 a las 19:00 horas**.
- Se debe subir un archivo zip que contenga:
 - Un jupyter notebook con las soluciones a las preguntas
 - La exportación a pdf de dicho Jupyter notebook
 - Informe con las ideas y/o puntos más importantes de sus soluciones