

2022

Nivel “AEDD”

Problemas

11 de Noviembre de 2022

Problema “Sembradora”

Autor: Alejandro Olivera Torres

Enunciado:
Se ha desarrollado un prototipo de sembradora ecológica -a energía solar- de semillas, la cual se puede mover sólo en línea recta dejando una semilla por celda.
En los campos de la región, la superficie de sembrado es de forma rectangular, y se encuentra dividida en celdas cuadradas. En cada celda sólo se puede sembrar y hacer crecer una planta si hay una semilla (y no más de una).
La sembradora puede pasar mas de una vez por una celda, lo cual provocará que en esa celda la semilla no se convierta en planta.

Entrada:
La primer línea de la entrada contiene dos valores enteros N y M que representan la cantidad de filas ($1 \leq N \leq 100$) y columnas ($1 \leq M \leq 100$), respectivamente, de la granja.
La segunda línea contiene un entero positivo K que representa la cantidad de pasadas de la sembradora.
El resto de la entrada, contiene líneas que representan las pasadas de la sembradora (en cada pasada se recorre una fila ó una columna completa). Cada una de estas líneas contendrá una letra 'F' seguida de un espacio y luego un entero que representa el número de Fila, o una letra 'C' seguida de un espacio y luego un entero que representa el número de la columna. Las filas están numeradas de arriba a abajo del 1 al N. Las columnas están numeradas de izquierda a derecha del 1 al M.

Salida:
Un entero no-negativo que es igual al número de semillas que se desperdiciaron.

Entrada	Salida
3 6 3 F 2 C 2 C 2	7

Problema “Juntando repetidas”

Autor: Nahuel Meyer

Enunciado: Luis es un niño amante del futbol, tanto así que decidió gastar sus ahorros para completar el album de figuritas del mundial, después de haber abierto muchísimos paquetes, realizó una lista con las figuritas que le fueron tocando, como el no sabe escribir muy bien, mezcló mayúsculas y minúsculas.
En vez de escribir “Messi”, pudo haber escrito “MesSI”, “meSSi”, “mEsSi”, etc.

Luis tiene muchos amigos y quiere cambiar sus figuritas con ellos, como sabe que eres un gran programador, te pidió que desarolles un programa, que dada la lista de figuritas, muestre una lista *Bonita* de las figuritas que se encuentran repetidas ordenadas por número de repeticiones, de haber empates en número de repeticiones, deberás ordenarlas por orden alfabético.
Una figurita se considera repetida, si aparece mas de una vez en la lista, y se deberá mostrar la cantidad de apariciones.
Una palabra es considerada *Bonita*, si tiene la primera letra en mayúsculas, y las demás en minúsculas, la lista es *Bonita*, si todas sus palabras son *Bonitas*.

Sin considerar mayúsculas y minúsculas, puedes quedarte tranquilo ya que Luis escribió la misma cadena para representar a la misma figurita; es decir, si escribió “Messi”, luego no escribio “Messi Leo” ó “Leonel Messi”, siempre respetó escribir el nombre del jugador como la primera vez (“Messi” en este caso).

Entrada:
Recibirás un número entero *N*, que corresponde al número de figuritas que Luis escribió en su lista. Luego recibirás *N* cadenas de caracteres *s*, que corresponden al nombre de cada figurita.

Salida:
Deberás mostrar una lista *Bonita,* ordenada descendentemente por número de repeticiones, de ser igual el número de repeticiones, deberás ordenarla por orden alfabético.

Entrada	Salida
4 mEssI RoNaldo meSSi Mbappe	Messi 2
4 LionelMessi cristianoRonaldo LiONeLMessi cristianoRonaldo	Cristianoronaldo 2 Lionelmessi 2

Problema “Ordenamiento por MMIs”

Autor: Daniel Ambort

Enunciado:
Ezequiel es un adolescente con una habilidad muy particular: es capaz de sostener un tubo de vidrio en su mano, que contiene N bolas numeradas de 1 a N, y con un *MMI* (Movimiento Mágico Imposible) del tubo, logra el siguiente efecto: Si el tubo no tiene todas las bolas ordenadas ascendentemente (con la mayor de ellas en la parte inferior del tubo), consigue que **la bola con el número más alto y que todavía no se encuentra en su lugar definitivo, se intercambie mágicamente con la bola que está inmediatamente por debajo de ella.**
Ezequiel desea poder ejecutar el truco con los ojos vendados, y por lo tanto quiere que tu le digas cuantos *MMIs* necesita hacer para asegurarse que el tubo está completamente ordenado.

Entrada:
La entrada comienza con una línea de texto con un número N (1 <= N <= 1000), y luego una línea de texto que contiene N números enteros positivos (todos en [1..N] sin ninguno repetido). El primer número de esta serie corresponde al número de la bola de la parte superior del tubo, y el último al número de la bola en la parte inferior del tubo.

Salida:
Muestra por pantalla una sola línea con el número de *MMIs* necesarios para ordenar completamente las bolas del tubo de forma ascendente.

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

Obs.: en el segundo ejemplo, Ezequiel realiza tres *MMIs* (en cada una de las columnas, el *MMI* se realiza sobre los números en negrita).

MMI1	MMI2	MMi3	
3	2	**2**	1
2	**3**	**1**	2
1	**1**	3	3

Problema “Primos Concéntricos”

Autor: Daniel Ambort

Enunciado: A Joaquín le encantan los números primos y le pareció interesante este problema de una guía de práctica de AEDD de hace 20 años, cuando su tío estudiaba en la UTN, el problema decía: Se recibe un número N (2 <= N <= 100 y N%2=0), y se debe construir una matriz con los primeros números primos que puedan caber hasta llenar completamente dicha matriz. Los números primos se deben almacenar en sentido horario ó antihorario, comenzando por el extremo superior izquierdo del borde externo de la matriz (ver los ejemplos!!!)
Como salida se debe visualizar los 4 números que queden en el centro de la matriz, respetando sus posiciones y el sentido (horario ú antihorario) de rellenado.

Entrada: un número entero N (1 < N < 100 y N % 2 == 0), y una cadena que indicará el sentido de rellenado de la matriz de NxN elementos: “Horario” u “Antihorario”.

Salida: los cuatro números primos que ocupan el centro de la matriz, en el orden en que fueron cargados en la matriz, según el proceso explicado.

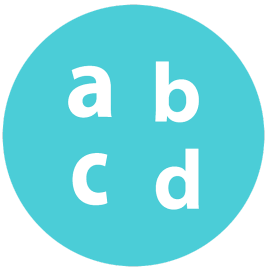
Entrada	**Salida**
2 Horario	2 3 7 5
6 Antihorario	137 151 139 149

Obs.: En el último ejemplo, la matriz queda:

```
  2   71   67   61   59   53
  3   73  131  127  113  47
  5   79  137  151  109  43
  7   83  139  149  107  41
 11   89   97  101  103  37
 13   17   19   23   29  31
```

Problema “Encontrando las subpalabras”

Autor: Tomás Assenza



Enunciado: María es tu amiga que estudia la carrera de Licenciatura en Letras. Como siempre fue muy apasionada en estos temas, le gusta escribir palabras (cadenas) y buscar subpalabras. Definimos a una subpalabra P como a cualquiera que se pueda formar solo usando letras existentes en una palabra M. No importa si esta nueva palabra existe o no en el diccionario.

Un día te reuniste con María y hablaron de todo lo que aprendieron en sus carreras hasta ahora, por lo que, mientras le contabas tus experiencias con las distintas materias de primer año, pasaste a contarle de Algoritmos y Estructuras de Datos y a ella le resultó interesante saber si podrías generar un programa que calcule la cantidad máxima de subpalabras que pueden ser generadas desde una palabra dada.

Debido a tu alto interés en AEDD, decidiste aceptar el reto.

Entrada:
La entrada contiene un solo string **S (1 <= S <= 7)** con una palabra. Se asegura que no se ingresarán espacios en el string.

Salida:
La salida consta de un único número con la cantidad máxima de subpalabras posibles en base a la palabra inicial.

Entrada	**Salida**
PERRO	170

Entrada	**Salida**
USB	15

Entrada	**Salida**
FIFO	34

Problema “Lo que la tarjeta SD se llevó”

Autor: Tomás Assenza



Enunciado: Esteban siempre fue muy apasionado por la fotografía. Desde muy chico disfrutaba sacándole fotos al paisaje del cielo santafesino, y habitualmente buscaba ideas para fotos nuevas. Lastimosamente, la memoria de la cámara que le regalaron cuando cumplió 14 años se corrompió, perdiendo muchas de sus fotos.

Desesperado, solicitó tu ayuda para recuperar sus fotos. Cuando abriste los archivos, te encontraste con que las fotos estaban en un formato extraño: en lugar de verse como fotos, sólo eran matrices de 0s y 1s. Por suerte, lograste crear un programa que las convierta a fotos pero notaste que cada vez que lo probabas terminaban dadas vueltas, e invertidas horizontalmente.

Para automatizar la recuperación de fotos, decidiste generar un programa que **de vuelta e invierta** estas matrices horizontalmente.

Para dar vuelta una matriz, se deben dar vuelta todas sus filas. Por ejemplo, dar vuelta una fila con contenido [1, 1, 0] resulta en [0, 1, 1].

Para invertir una matriz, se deben invertir todas sus filas. Esto significa cambiar todos los 0s por 1s, y todos los 1s por 0s. Por ejemplo, invertir [0, 1, 1] resulta en [1, 0, 0].

Entrada:
La entrada contiene un número n, seguido de una matriz de **nxn (1 <= n <= 20)** conteniendo números enteros **imagen[i][j] (0 <= imagen[i][j] <= 1).**

Salida:
La salida consta de la matriz dada vuelta e invertida.

Entrada	**Salida**
3 1 1 0 1 0 1 0 0 0	1 0 0 0 1 0 1 1 1

Entrada	**Salida**
4 1 1 0 0 1 0 0 1 0 1 1 1 1 0 1 0	1 1 0 0 0 1 1 0 0 0 0 1 1 0 1 0

Problema “Cuánto te falta para terminar???”

Autor: Daniel Ambort

Enunciado: María tiene una gran biblioteca con libros de ciencia ficción, y cada vez que su amigo Lucas la visita, le pide prestado un libro para leer (y Lucas tiene la habilidad de elegir siempre un libro que María todavía no ha leído, y luego de eso, María inmediatamente tiene muchas ganas de leer ese libro -ni ella sabe aún si es porque reconoce que Lucas elige buenos libros, ó tiene miedo que Lucas no le devuelva el libro nunca más…-)
Entonces, a los pocos días, María, cada vez que se encuentra con Lucas, le preguntará:

Cuánto te falta leer del libro que te presté ???

A lo que Lucas, siempre demasiado vueltero para expresarse (pero exacto), le responderá, algo como:

- *Hasta el día de hoy leí **1/4** del libro*
- *Si leo **20** páginas más, sólo me faltará leer **1/3** del mismo.*

De esa forma María podría deducir que el libro que le prestó tiene 48 páginas.
María necesita que construyas un programa para poder ingresar los datos númericos de las dos afirmaciones de Lucas, y que le informe la cantidad de páginas que tiene el libro que le prestó, y cuántas le faltan leer.

Nota: Lucas siempre contesta con dos afirmaciones semejantes a las que vimos arriba. María quiere que tú le diseñes un programa, para que ella pueda ingresar 3 valores (el denominador y el numerador de la fracción de la primer afirmación, la cantidad de páginas al comienzo de la segunda afirmación, y el denominador y el numerador de la última fracción). Como salida deberás visualizar el total de páginas del libro, y en la línea siguiente la cantidad de páginas que le quedan por leer del mismo.

Entrada: un par de valores enteros (D1 y N1) que corresponden a la fracción de la primer afirmación de Lucas, un valor entero B en [0..1000], la cantidad de páginas de la segunda afirmación (valor en 1..1000), y otro par de valores enteros (D2 y N2) que corresponden a la fracción de la segunda afirmación de Lucas. (D1, N1, D2 y N2 son valores en [0..10] y corresponden a valores de fracciones siempre “correctas”). En ningún caso puede suceder que Lucas diga que ya leyó el libro completo…

Salida: un valor entero que indica la cantidad total de páginas del libro en una línea, y en la siguiente un valor entero con la cantidad de páginas que debe leer Lucas para terminar el libro.

Entrada	Salida
1 4 20 1 3 // son los valores que configuran las dos afirmaciones que siempre hace Lucas	48 36
3 4 20 1 5	400 100

Nota: en la tabla anterior, el texto que comienza con "//" son comentarios para facilitar la comprensión de entradas y salidas, y no forman parte de las mismas.

Problema “Dímelo con una frase”

Autor: Daniel Ambort

Enunciado: Guillermo estudia ingeniería en sistemas de información, y como está muy acostumbrado a trabajar con números, cada vez que su mamá Clementina le pregunta “Guille, cuánto te queda por estudiar de la materia que rendís el viernes ?”, él acostumbra responder con una frase cómo “el 52.28 %”. Guillermo estudia cada materia con un plan riguroso de 14 días -su madre lo sabe, pero le dá pereza convertir porcentajes a días-, para determinar que 52.28% son algo más de 7 días.
La mamá, más acostumbrada a frases que a números, le pidió que este tipo de respuestas las exprese en frases (que ella -de alguna extraña manera- comprende con precisión absoluta, porque las madres tienen un sexto sentido…), según la siguiente tabla:

Porcentaje	**Expresión Usual**
[0]	“Nada”
]0..20]	“Poco”
]20..30]	“Un cuarto maso”
]30..45]	“Un tercio maso”
]45..55]	“La mitad maso”
]55..75]	“Mas de la mitad”
]75..100[	“Casi todo”
[100]	“14 dias”

Entrada: un valor flotante P (0 <= P <= 100) que indica el porcentaje de días que le faltan a Guillermo para terminar de estudiar la materia que debe rendir.

Salida: una cadena de caracteres con la frase que Clementina comprende perfectamente.

Entrada	**Salida**
52.28	La mitad maso
100	14 dias

Problema “Choque De Potencias”

Autor: Juan Pablo Cabaña

Enunciado: Juan y Yanina juegan todos los días a choque de potencias, el juego consiste en adivinar el valor de una suma de potencia de dos (desde 0 a N). En este juego, Yanina le pregunta un exponente (N) a Juan y él tiene que responder con la suma de potencias de 2 desde 0 a N. Si Juan acierta, entonces Juan gana, sino gana Yanina. Sin embargo, hace varios días Yanina no para de ganarle a Juan al preguntarle exponentes muy grandes. ¿Podrías ayudar a Juan a derrotar a Yanina?

Entrada: La primera línea contiene un entero T ($1 <= N <= 10^5$), representando el número de casos de prueba. Las siguientes T líneas contienen un entero N ($0 <= N <= 10^9$).

Salida: Por cada entero N, imprimir la suma de potencias módulo (10^9) + 7, ya que la suma de potencia puede alcanzar valores grandes.

Entrada	Salida
5 0 1 2 3 4	1 3 7 15 31

Problema “Puerta de Escape”

Autor: Juan Pablo Cabaña

Enunciado: J.G. está en el inicio de un laberinto, sin embargo, no es un laberinto normal. En este caso no hay una salida definida, sino que cada vez que intenta escapar, la puerta de escape cambia.

Formalmente, J.G. siempre arranca en la habitación 1, y en cada intento de escape la puerta de escape se encuentra en la habitación X y se abre únicamente en el segundo K. En el segundo 0, J.G. se encuentra en la habitación 1, en un segundo puede moverse a cualquier habitación a la que se encuentre conectada su habitación actual. J.G. no puede permanecer quieto en la habitación actual, es decir que luego de cada segundo debe estar en una nueva habitación.

J.G. necesita saber si puede escapar del laberinto dado una habitación X y un segundo K, y de cuántas formas puede hacerlo (si es que las hay).

Entrada: La primera línea contiene un entero N (1 <= N <= 100), el número de habitaciones en el laberinto.

La segunda línea contiene un entero C (0 <= C <= N^2), el número de conexiones entre habitaciones.

Las siguientes C líneas contienen 2 enteros A (1 <= A <= N) y B (1 <= B <= N), que indican que hay una conexión bidireccional entre las habitaciones A y B.

La siguiente línea contiene un entero Q (1 <= Q <= 10^5), la cantidad de veces que J.G. va a intentar resolver el laberinto.

Las siguientes Q lineas contienen un entero X (1 <= X <= N) y K (0 <= K <= 64), la habitación de escape y el segundo en el que se abre la puerta de escape.

Salida: Imprimir para cada par de enteros X, K, la cantidad de formas en las que puede ser resuelto el laberinto módulo (10^9)+7, si no existen caminos posibles, imprimir 0.

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

Problema “Probando Fármacos”

Autor: Juan Pablo Cabaña

Enunciado: Alejandro trabaja en un instituto que busca encontrar fármacos eficaces contra el Alzheimer y el Parkinson. Antes de pasar a la fase de prueba real, donde se crean los fármacos y se prueban en moléculas reales, se simula el efecto que tendrán mediante una computadora.

Formalmente, un fármaco está compuesto por una cadena de letras donde cada letra representa un tipo de molécula. La eficacia de un fármaco se mide por la cantidad de veces que aparece el mismo en la célula donde va a actuar. Cada célula también se encuentra representada por una cadena de letras.

Entrada: La primera línea contiene la cadena de letras (letras mayúsculas del alfabeto inglés) correspondiente a la célula donde actuará el fármaco. Cada célula tiene a lo sumo 10000 letras.

La segunda línea contiene un entero N (1 <= N <= 1000), la cantidad de fármacos candidatos a probar.

Las siguientes N líneas contienen una cadena de letras (letras mayúsculas del alfabeto inglés) correspondiente al fármaco a probar. Cada fármaco tiene a lo sumo 7 letras.

Salida: Imprimir el o los fármacos que tengan la mayor eficacia, en el orden en el que fueron dados en la entrada.

Entrada	Salida
AABAAB 4 ABA AAB B BAA	AAB B

Problema “Función Incógnita Recursiva”

Autor: Daniel Ambort

Enunciado: El papá de Nahuel (alias “Joyanes”) era de joven un programador muy celoso de la propiedad intelectual de su código, y por lo tanto elegía nombres para sus funciones que no seguían normas básicas de buen nombrado de las mismas. En aquella época las funciones recursivas eran una novedad…, y Nahuel no está muy convencido que realmente Joyanes las supiera usar eficientemente.

Nahuel encontró una hoja de código impresa con la siguiente función (que por el tiempo transcurrido se dañó y varias letras ó palabras desaparecieron -indicadas con guiones bajos-):

```
_____ Incognita(int x, int y){
__ (y==1) return true;
else
    if (x%y == 0)
      return _____
    else
      return Inc___nita(x, y-1);
}
```

además Nahuel encontró el siguiente código:

```
cin >> N;
int generados=0, numero=2;
while (generados<N){
      if (Incognita(numero, numero-1))
          generados++;

      numero++;
}
cout << numero-1 << endl;
```

y tiene una tabla, escrita a mano muy prolijamente por Joyanes, donde registraba las salidas correctas para diferentes valores de N, cuando se ejecutaba ese bloque de código:

N	Salida
2	3
4	7
8	19
28	107
1500	12553

Tu tarea es lograr reconstruir la función incógnita, para entender que valores está retornando. Nahuel sabe que alguien cargó este problema en la competencia TecnoMate 2022, y para lograr obtener el aceptado, tiene que trabajar sobre el código para que genere los valores correctos para cualquier N.

Entrada: Un valor entero positivo N (1<=N<=5000)

Salida: El valor entero que imprime en pantalla el bloque de código que contiene el while que invoca a la función Incógnita.

Entrada	Salida
1500	12553