

El sumidero de Spacla (sumidero)

“Teniamo un objetivo y estamos a un pasito de conseguirlo, ¿y saben que es lo mejor de todo? Que depende de nosotros.”

Lionel Messi, *Final de Copa America*, 2021

Spacla es un niño muy travieso y curioso por las matemáticas y sus distintas estructuras. Realmente no puede dejar a las matemáticas en paz, le fascinan.

Dado cierto día D perteneciente al conjunto de días $\text{JUNIO} = \{1, 2, 3, \dots, 30\}$ de un año arbitrario; Spacla se ingenió una nueva forma de pasar sus productivas tardes, un juego al que llamaba “El sumidero de Spacla”.

Paso un rato divirtiéndose mucho resolviendo y resolviendo los distintos ejercicios que el mismo planteaba sobre la estructura matemática que descubrió el día D , hasta que pensó de una manera muy sabia “Si yo fuese informático, ¡podría resolver esto mil veces más rápido con una estructura de datos! Creo que la llamaría, planta de segmentos o Segment Plant para los anglosajones, si, así la llamaría!”

Que tristeza que nuestro amigo Spacla no es informático, ahora debe hacer sus cálculos a mano. Pero claro, pasaron unos días y Spacla se aburrió de su sumidero porque demostró una formula simple para resolver cualquier tipo de ejercicio relativo a su problema.

Como Spacla odia desperdiciar su conocimiento, decide darle los derechos de autor de su “Sumidero de Spacla” a su gran amigo Penguino, para que este pueda usarlo en algún problema de la Olimpiada de Ositos Barrigoncitos Ilógicos. Obviamente Spacla le pide a Penguino que le enseñe una solución con “Planta de segmentos” una vez terminado su problema; a lo cual Penguino solo lo mira confundido y se ríe un poco.

En fin, esto nos trae a este punto exacto en la historia, con Penguino planteando un problema a pesar de que debe estar redactando la tesis más tesis en la historia de las tesis. Pero, te preguntaras... **¿DE QUE TRATA ESTE SUMIDERO DE SPACLA?**

Déjame explicarte, el sumidero de Spacla es una estructura que es como una pirámide de números al revés. Empieza con una hilera de N números naturales, la siguiente hilera (Debajo de la primera) tiene $N - 1$ números naturales, obtenido de sumar las 2 casillas inmediatamente superiores.

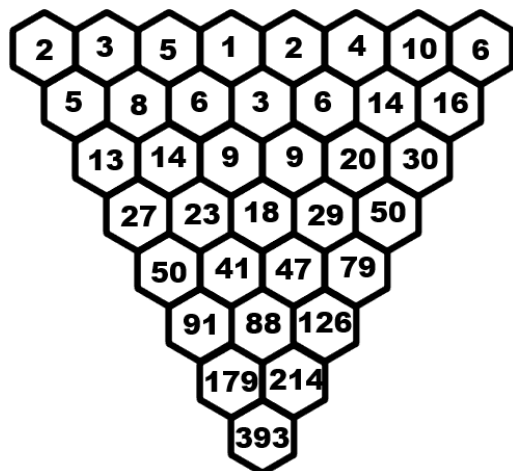
Es decir, la casilla i en la hilera j será el resultado de sumar las casillas i y $i + 1$ de la hilera $j - 1$. Así las filas de números van teniendo cada vez un numero menos.

Spacla le dio la clave de la diversión a Diego, explicándole que se deben hacer Q acciones, estas pueden ser consultas o actualizaciones.

- **CONSULTA:** Dado una fila F , y una posición P , decir que numero se encuentra en aquella casilla.
- **ACTUALIZACION:** Se cambia el valor de una posición específica de la primera fila de números. Y para asegurarse que la actualización ha sido buena, se imprime una CONSULTA con $F = N, P = 1$, es decir el valor que se encuentra en el fondo del sumidero.

Penguino sigue intentando recordar cómo se resolvía este problema cuando se piden consultas masivas, es decir una consulta donde se te pide encontrar la suma de toda una fila... ¿Tu podrías resolverlo? Por suerte este problema no pide eso...

Aqui podemos ver un ejemplo:



En este caso, si se te pregunta cuál es el valor en la fila 2 ($F = 2$), posición 3 ($P = 3$), deberías responder 6.

Como los números en el sumidero pueden ser increíblemente grandes, debes responder usando modulo $10^9 + 7$.

Entrada

Se te darán dos números enteros N, Q ; representando el número de enteros en la primera fila y el número de acciones respectivamente.

Sigue una línea con N enteros V_i , los valores de la primera línea del “sumidero”, donde V_i es el i -esimo valor de la primera línea empezando por el valor V_1 .

Siguen Q líneas, cada una con un carácter.

‘A’ para actualización, a esta le sigue dos valores P y V (La posición a actualizar en la primera fila y el valor actualizado respectivamente).

‘C’ para una consulta, a esta le sigue dos números enteros F y P (La fila y posición consultada respectivamente).

Salida

Por cada ‘C’ en la entrada debes sacar el resultado de la consulta en una línea.

Por cada ‘A’ en la entrada debes sacar el valor de la casilla más inferior del sumidero.

Límites

- $1 \leq N \leq 10000$
- $1 \leq Q \leq 200$ cuando $N \leq 6000$, $1 \leq Q \leq 100$ cuando $6001 \leq N \leq 10000$
- $0 \leq V, V_i \leq 10^9$ para todo $1 \leq i \leq N$
- $1 \leq F \leq N$
- $1 \leq P \leq N - F + 1$

Ejemplos

Entrada	Salida
5 5 2 4 5 3 9 C 2 3 A 4 4 C 3 2 C 1 1 C 2 2	8 73 18 2 9
5 5 1 1 1 1 1 C 5 1 C 3 2 C 2 1 A 3 10 C 5 1	16 4 2 70 70
16 5 100000 100000 100000 100000 100000 100000 100000 100000 100000 100000 100000 100000 100000 100000 100000 100000 C 16 1 C 10 4 C 2 15 A 1 1 C 16 1	276799979 51200000 200000 276699980 276699980

Subtareas

- (7 puntos) para todo $i : V_i = 1$, y no habrá acciones de actualización, es decir los valores en la primera fila siempre serán 1.
- (8 puntos) No habrá acciones de actualización y $N \leq 1000$.
- (5 puntos) $N \leq 100$
- (21 puntos) $N \leq 6000$ y toda consulta de tipo 'C' siempre tendrá $F = N$.
- (13 puntos) $N \leq 6000$.
- (46 puntos) Limites originales.

NOTA:

Se te aconseja usar las siguientes lineas al inicio de tu funcion main:

```
std::ios::sync_with_stdio(false);  
cin.tie(0);  
cout.tie(0);
```

Usar '\n' en lugar de endl al imprimir tus respuestas.

Y ah claro! Ten cuidado con la memoria ;) *Usa long long solo para calcular resultados, int para guardar valores(Esta recomendacion no aplica si apuntas a los 100 puntos, para 100 puntos no deberia preocuparte la memoria siempre y cuando tu solucion enserio saque 100 puntos).*