

## Construyendo Pueblos

*Contribución de Lautaro Lasorsa*

### Descripción del problema

El reino de Arbolonia está colonizando una nueva isla, y para eso busca construir nuevos pueblos en ella. El procedimiento para construir un pueblo es el siguiente :

- Primero, se llega al único puerto que hay en la isla y se desembarca. En este momento se elige un número del 0 al 9, que indica en qué dirección se seguirá. El 0 es la dirección más a la izquierda y el 9 la dirección más a la derecha.
- Después de recorrer una cierta distancia por el camino elegido, se vuelve a repetir la elección, nuevamente eligiendo una dirección entre 0 y 9 para seguir.
- Después de  $M$  elecciones, se llega al punto en el que se construirá el nuevo pueblo.

Se garantiza que, si para dos pueblos las primera  $i$  elecciones fueron iguales y en la elección  $i + 1$  el primero toma una ruta más a la derecha de la que toma el segundo, quedará más a la derecha sin importar las siguientes decisiones.

Es decir, sea  $A$  el vector que representa las decisiones tomadas en la construcción del primer pueblo (es decir,  $A_0$  contiene el entero de la primera decisión,  $A_1$  el de la segunda, etc.), y  $B$  el vector que representa las decisiones tomadas para construir el segundo, el pueblo más a la derecha será el que corresponda al vector lexicográficamente mayor.

El rey de Arbolonia quiere seguir la colonización en tiempo real, y para eso necesita las siguientes operaciones :

- Se crean 2 pueblos,  $A$  y  $B$ , con  $A$  a la derecha de  $B$ , con una población  $P_A$  y  $P_B$  respectivamente. Después de hacer esto, quiere saber cuál es la

población total de los pueblos que ya existían y quedaron comprendidos entre  $A$  y  $B$ . Es decir, están a la derecha de  $A$  y a la izquierda de  $B$ . Se garantiza que ningún pueblo es creado 2 o más veces.

- Indicar que un pueblo  $C$ , ya existente, sufrirá un cambio en su población de  $x$ , es decir, la población aumenta en  $x$ . Si es negativo indica que la población se redujo. Desea saber después de esto cuál es la población de dicho pueblo.

Tu trabajo es implementar el sistema informático necesario para responder las preguntas de su majestad.

### Detalles de implementación

Debes implementar las funciones

- `Inicializar(N, M)`, una función que será llamada una única vez al inicio del programa. Recibirá los parámetros  $N$  y  $M$  del tipo ENTERO, que son la cantidad de llamadas que se realizarán en total a las funciones `Crear(A, B, PA, PB)` y `Cambiar(C, x)`, y el largo de los vectores respectivamente. No debe retornar nada.
- `Crear(A, B, PA, PB)`. Siendo  $A$  y  $B$  del tipo VECTOR DE ENTEROS y de largo  $M$  y  $PA$  y  $PB$  del tipo ENTERO LARGO. Debe retornar un ENTERO LARGO, correspondiente a la respuesta.
- `Cambiar(C, x)`. Siendo  $C$  del tipo VECTOR DE ENTEROS y de largo  $M$ , y  $x$  del tipo ENTERO LARGO. Esta función debe retornar un ENTERO LARGO, correspondiente a la respuesta.

El largo  $M$  es el mismo para todos los vectores de un mismo caso de prueba.

**Evaluador local**

El evaluador local primero leerá una línea con la cantidad de llamados a las funciones `Crear(A,B,PA,PB)` y `Cambiar(C,x)`,  $N$ , y el largo de los vectores  $M$  y va a llamar a la función `Incializar(N,M)`.

Luego, el evaluador local lee de la entrada estándar:

- Las  $N$  consultas. En cada una lee una primera línea un entero  $t$ , que indica el tipo de consulta.
- Si  $t = 1$ , lee en las 2 siguientes líneas los vectores  $A$  y  $B$ . Cada línea debe contener  $M$  enteros. Finalmente lee una última línea con los enteros largos  $PA$  y  $PB$ . Luego llama a `Crear(A,B,PA,PB)` y muestra el resultado por salida estándar.
- Si  $t = 2$ , lee en la siguiente línea el vector  $C$ , que deberá contener  $M$  enteros, y luego el entero largo  $x$ . Luego llama a `Cambiar(C,x)` y muestra el resultado por salida estándar.

Se garantiza que la primer consulta es de tipo 1.

**Cotas**

- $4 \leq N * M \leq 500.000$
- $0 \leq A_i, B_i, C_i \leq 9$
- En todo momento la población de cada pueblo es un entero entre 0 y  $10^{12}$

**Ejemplos**

Si el evaluador local recibe la siguiente entrada:

```
8 5
1
0 1 2 3 4
0 1 2 3 5
1000 1000
1
0 1 2 3 3
0 1 2 3 6
500 1000
1
0 1 2 3 2
0 1 2 3 7
2000 1000
2
0 1 2 3 3
9500
2
0 1 2 3 6
9000
1
0 0 0 0 1
9 9 9 9 8
1000000 1000000
1
0 0 0 0 0
9 9 9 9 9
1 1
2
0 0 0 0 0
-1
```

Una implementación correcta deberá devolver:

```
0
2000
3500
10000
10000
25000
2025000
0
```

En cambio, si recibe:

```
4 6
1
0 1 2 3 4 5
1 0 0 0 0 0
10 10
1
0 1 2 3 5 6
2 0 0 0 0 0
100 100
2
2 0 0 0 0 0
-99
1
0 0 9 9 9 9
3 0 0 0 0 0
100 100
```

Deberá devolver:

```
0
10
1
121
```

### Subtareas

1.  $4 \leq N * M \leq 5.000$ . (10 puntos)
2. La respuesta a todas las llamadas a la función `Crear(A,B,PA,PB)` es 0. (10 puntos)
3. En los llamados la función `Crear(A,B,PA,PB)` los vectores *A* y *B* solo difieren en la última posición. (20 puntos)
4. Los vectores *A*, *B* y *C* solo contienen los valores 0 y 1. (20 puntos)
5. Sin restricciones adicionales. (40 puntos)