

Proyecto 1: V de AVES

Es muy conocido que algunas especies de aves vuelan creando una formación en V. La razón es simple: de esa forma las aves que van detrás aprovechan el rebufo de las que van delante. Esto implica que el pájaro que hace más esfuerzo es aquél que va primero ocupando el vértice de la V, por lo que es importante elegir bien quién es.



Algunas especies lo que hacen es colocarse en orden de edad de izquierda a derecha, de forma que el pájaro que queda en el centro es el que no es ni demasiado joven ni demasiado viejo. De esta forma evitan que el que abre camino esté poco desarrollado o demasiado cansado para tirar del grupo. Cuando a la bandada se van uniendo más pájaros, ocupan su lugar y, si es necesario, el pájaro que abre el camino cambia.

Por ejemplo, imaginemos que hay una bandada en vuelo con pájaros de edades 10, 20 y 30 meses. En ese caso, el que ocupará el primer lugar es el de 20 meses. Si ahora llegan dos pájaros nuevos, uno de 25 meses y otro de 35, ambos ocuparán su sitio y el primero pasará a ser el de 25 meses. Si posteriormente llega un abuelo de 40 meses con su nieto de 5, ocuparán los extremos y el primer puesto no variará.

La Asociación Venezolana de Estudios Superiores (AVES) ha solicitado ayuda a los alumnos de Algoritmos y Estructura de Datos para ayudarles a estudiar el comportamiento de este tipo de aves. Lo que se quiere es simular la formación de una de estas bandadas de pájaros que comienza con un único ejemplar y a la que se van añadiendo y eliminando nuevas parejas. Su programa deberá ser capaz de simular este proceso y determinar rápidamente cual es el ave que ocupa la punta de la bandada.

Entrada:

La entrada será por entrada estándar, la primera línea contendrá 2 enteros A y Q indicando la edad del ave inicial y el número de consultas o acciones a realizar. Luego vendrán Q líneas, cada una tendrá un entero al inicio que será 1, 2, 3, o 4, si es un 1, significa que se quiere saber cuál es el ave que está en la punta de la bandada, si es un 2 entonces a continuación vendrán 2 enteros indicando las edades de 2 nuevas aves que entraran en el grupo, el 3 indica que se desea saber todas las aves que están en la bandada y por último el 4 que también tendrá 2 enteros indicando la edad de las aves que se retiran de la bandada.

Salida:

El programa debe dar respuesta a cada consulta del tipo 1, indicando en una línea para cada consulta, la edad del ave que está en la punta de la bandada y para cada consulta de tipo 3 deberá imprimir todas las aves en una línea ordenadas de menor a mayor edad.

Ejemplo:

Entrada	Salida
30 8	30
1	20
2 10 20	25
1	25
2 35 25	5 10 20 35 40
1	20
2 5 40	
1	
4 25 30	
3	
1	

Consideraciones:

- Todas las edades de las aves son distintas, es decir no hay 2 aves que tengan la misma edad.

- La edad de las aves se representara con un entero positivo no mayor a 1.000.000.000
- Se asegura que las aves que salen de la bandada son válidas, es decir no se indicara la salida de un ave que no exista en la bandada.
- La entrada y salida debe ser estricta, es decir su programa debe leer los datos tal y como se especifican, e imprimir las respuestas tal como se especifica en el ejemplo. Cualquier información adicional que solicite o proporcione será penalizada.
- La consulta del tipo 1 debe ser realizada en tiempo constante es decir $O(1)$.
- Debe realizar una implementación lo más eficiente posible, puesto que su programa se pondrá a prueba con casos que pueden causar que este tarde más de lo necesario.
- La lectura y escritura de datos debe ser hecho usando la Entrada/Salida estándar de C++.
- El programa debe ser realizado en C++ sin el uso de STL para las estructuras de datos, a excepción de la clase **string** provista por C++.
- Debe emplear Programación Orientada a Objetos en su programa.
- El código fuente debe estar intradocumentado adecuadamente. Se debe entregar un informe en formato PDF con la explicación de la solución de su proyecto (sin incluir código), explicación de las estructuras usadas, la complejidad de cada una de las funciones descritas y la identificación del autor. La ausencia de este archivo califica al proyecto como no entregado.
- Será compilado y evaluado en el compilador g++ (Linux). Si el proyecto no compila no será corregido.
- El proyecto debe ser realizado de manera individual.
- La revisión del proyecto no es obligatoria y solo se revisaran proyectos que el grupo docente considere necesario.
- El proyecto debe entregarse el día 29 de julio de 2016 a través de la página de la asignatura, hasta las 11:59 pm como hora límite. No se aceptarán proyectos fuera de las horas establecidas para ello, y se considerarán como no entregados.

Las copias entre proyectos tendrán una calificación de 0 puntos, además de una sanción para todos los involucrados