

Solucionario: Nivelación Ronda 3

**Temática General: Estructuras de datos
e iteradores**





A

- **Nombre:** Palabras guardadas
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/334/detalle/es>
- **Análisis:** La finalidad del problema se resume en guardar palabras únicas y mostrarlas en orden léxico gráfico inverso. Además, de mostrar las palabras en mayúscula.
- **Implementación:** Lo primero a hacer, es leer las palabras y pasarlas directamente a mayúsculas, luego, guardarlo en un set, ya que este ordena y guarda cosas únicas, Por último, usando los iteradores inversos (rbegin, rend) mostramos los datos guardados en el set
- **Entradas:**
pan
ARROZ
PAN
- **Salidas:**
PAN
ARROZ





B

- **Nombre:** Matriz loca
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/21/detalle/es>
- **Análisis:** Seguir las instrucciones de la descripción.
 1. Llenamos la matriz
 2. Sumarle 1 a todas las casillas pares en las filas
 3. Sumarle 2 a todas las casillas pares en las columnas
 4. Restarle 3 a todas las casillas que se encuentran en una columna y una fila impar a la vez.
 5. Mostrar la matriz resultante





PASOS

• Paso 1: 1 3 5

2 4 6

4 8 2

• Paso 2

2 4 6

2 4 6

5 9 3

• Paso 3: 4 4 8

4 4 8

7 9 5

• Paso 4:

4 4 8

4 1 8

7 9 5





C

- **Nombre:** Diccionario
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/46/detalle/es>
- **Análisis:** Guardas palabras únicas (sin repetición) y mostrarlas en orden ascendente. Finalmente mostrar '-' para separar los casos.
- **Implementación:** Guardar las palabras en un set, ya que este guarda palabras únicas y en orden. Luego, mostrar su contenido y finalmente mostrar '-' para separar los casos de prueba





EJEMPLO C

- Entradas:

2

3

primera

segunda

tercera

4

palabras

sin

orden

alfabetico

- Salidas:

primera

segunda

tercera

-

alfabetico

orden

palabras

sin

-





D

- **Nombre:** Palabra mas común
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/49/detalle/es>
- **Análisis:** Dada N palabras, encontrar la que más se repite. Si todas tienen la misma cantidad de repeticiones, mostrar la palabra menor léxico gráficamente
- **Entradas:** 2
4 texto corto sin repeticiones
5 la letra de la palabra
- **Salidas:**
corta
la





E

- **Nombre:** Paréntesis
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/27/detalle/es>
- **Análisis:** Nos están dando una cadena de paréntesis '(' , ')' y debemos determinar si esta balanceado o no. Si todos los paréntesis que se abren, se cierran después. Entonces esta balanceado.
- **Ejemplo:** (()) CORRECTO
)(INCORRECTO





- **Ilustración:** Entrada: `()()`

Agregamos el '(' Luego hay un paréntesis de cierre ')', así que eliminamos el anterior. Luego, aparecen dos paréntesis que se abren seguidos, los agregamos '(' ('Luego, aparecen dos paréntesis de cierre ') ')', así que eliminamos los dos abiertos anteriormente.

- **Ilustración 2:** Entrada: `)((`

Aparece primero un paréntesis de cierre, pero como no hay nada a eliminar, toca agregar ese paréntesis de cierre. Ya que no puedo eliminar algo que no está. ')' Luego, aparecen dos paréntesis que se abren, los agregamos '(' '(' Como no hay más paréntesis, nuestra pila quedo con '(' Tercer elemento agregado '(' Segundo Elemento agregado ')' Primer elemento agregado Como quedó con elementos, quiere decir que no está balanceado los paréntesis





F

- **Nombre:** Torre de CDs
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/30/detalle/es>
- **Análisis:** Hacer uso de la pila, si se inserta un número > 0 agregarlo a la pila, si es 0, debemos mostrar el top de la pila y eliminarlo después, porque se entiende que este cd ha sido sacado. Si hay el número es 0 y no hay nada en la pila, no hacer nada..
- **Implementación:** Estructura de datos stack (pila)

Ejemplo

- | | | | |
|-------------|----------------|----------|----|
| • Entradas: | 1 | Salidas: | 30 |
| | 6 | | 20 |
| | 10 20 30 0 0 0 | | 10 |





G

- **Nombre:** Pila de strings
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/225/detalle/es>
- **Análisis:** Agregar una cadena o eliminarla, si la instrucción es PUSH la agregamos a nuestra pila. Si la instrucción es POP, mostramos el top (cima) de la pila y la eliminamos después, si no hay nada al momento de hacer POP, mostramos '-'
- **Implementación:** Estructura de datos stack (pila)

Ejemplo

Entradas:

```
7
PUSH string1
PUSH string2
POP
PUSH string3
POP
POP
POP
```

Salidas:

```
String2
String3
String1
-
```





H

- **Nombre:** Reporte de gestion
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/51/detalle/es>
- **Análisis:** Debemos guardar el nombre de un producto y la cantidad de clientes que tiene. Luego debemos mostrar el primer y el último producto, pero en orden léxico gráficamente, (Me refiero al nombre del producto)
- **Implementación:** Usar un mapa, ya que este me permite guardar dos datos a la vez. Además como ordena sus datos por la clave. Prácticamente ya nos soluciona el ejercicio. Ahora para mostrar el primer dato, este va a estar en la posición inicial, accedemos a este con begin. Ahora como debemos mostrar el último dato, podemos usar el iterador reverso rbegin para acceder al último dato. No funciona en iterador end ya que este retorna la posición posterior del último elemento.





EJEMPLO H

- Entradas:

virtualoffice 1221

priceapp 144

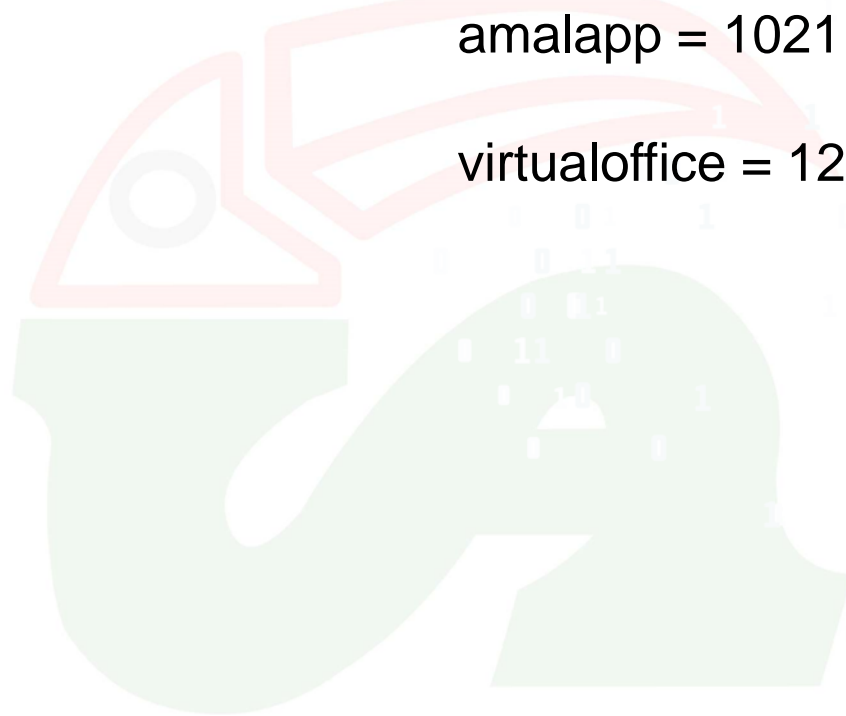
amalapp 1021

roundUp 932

- Salidas:

amalapp = 1021

virtualoffice = 1221





- **Nombre:** Estación de buses
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/25/detalle/es>
- **Análisis:** Debemos simular el registro y la salida de buses. Si la instrucción es 1, debemos agregar a la cola la placa del bus, si la instrucción es 2, debemos mostrar y eliminar el primer bus en la cola del registro.
- **Implementación:** Como es una simulación de una cola, usamos obviamente la cola ya que debemos agregar las placas de los registros y cuando uno finalice (instrucción 2) es porque es el bus que esta en el frente de la cola

Ejemplo

- | | | | |
|-------------------|----------------------------|----------------|----|
| • Entrada: | 1 10 //Se agrega a la cola | Salida: | 10 |
| | 2 0 //Finaliza el 10 | | 20 |
| | 1 20 //Se agrega a la cola | | 10 |
| | 1 10 //Se agrega a la cola | | |
| | 2 0 //Finaliza el 20 | | |
| | 2 0 //Finaliza el 10 | | |





- **Nombre:** Hola Mundo 2.0
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/392/detalle/es>
- **Análisis:** Debemos saludar a las personas, pero con la restricción de que si aparece un nombre repetido, este saludo se dirige al final.
- **Implementación:** Usar lista, ya que esta cuenta con la función remove, cuando un nombre este repetido lo removemos y lo agregamos al final.
¿Pero como sabemos si un nombre ha sido agregado antes? Usando la función count, si este conteo es 0, agregamos el nombre, si es 1, removemos el nombre y lo agregamos al final.
Al finalizar, simplemente mostramos el contenido de la lista.





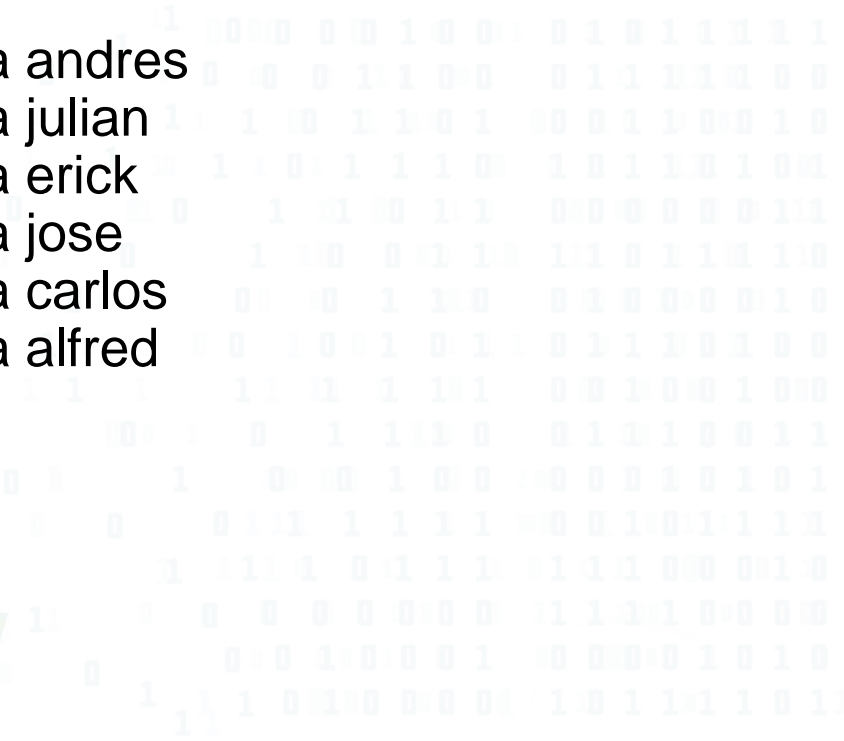
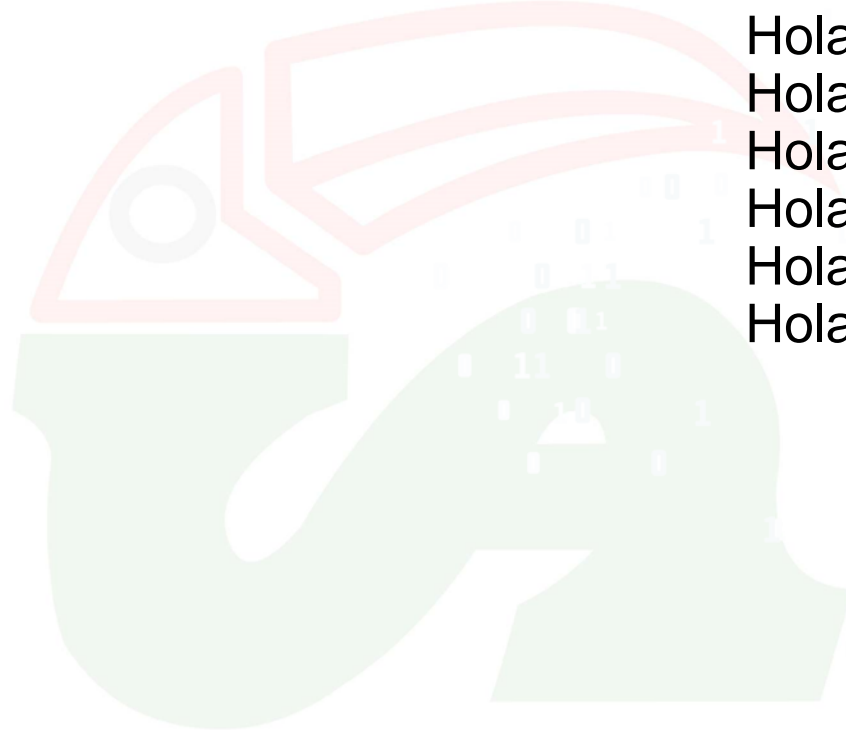
EJEMPLO H

- Entradas:

Alfred
erick
andres
Alfred
julian
erick
carlos
jose
carlos
alfred

- Salidas:

Hola andres
Hola julian
Hola erick
Hola jose
Hola carlos
Hola alfred





- **Nombre:** Carrito de compras
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/70/detalle/es>
- **Análisis:** Nos dan una lista de productos, con su respectivo costo, la tarea es determinar cuanto nos vale comprar ciertos productos de esa lista dada una cantidad a comprar..
- **Implementación:** Usar un mapa, ya que nos permite almacenar el nombre del producto y el costo, luego cuando vayamos a consultar el costo total por producto, iremos sumando el resultado de multiplicar la cantidad de ese producto por el valor almacenado en ese producto





- **Entrada:**

3 //Son los productos de la lista
leche 15 //La leche cuesta 15
pollo 5 //Cuesta 5
carne 8 //Cuesta 8
2 //Vamos a comprar 2 productos de la lista
leche 2 //La leche, una cantidad de 2. $2 \times 15 = 30$
carne 1 //La carne, una unidad, $1 \times 8 = 8$.
Total Acumulado = 38

- **Salida:**

38





- **Nombre:** Reporte de notas
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/282/detalle/es>
- **Análisis:** Nos dan un listado de nombres, con un código que lo identifica. Luego, nos dan el código con la nota de cada estudiante. Nuestra tarea es mostrar el nombre del estudiante con su nota. Pero ojo, el orden en que mostramos es dado por el código.
- **Implementación:** Uso de un mapa, la clave será la identificación y el contenido del mapa será un pair, el primer dato el nombre y el segundo dato la nota. Finalmente, mostramos el contenido del mapa





• Entrada:

Julian	11550
Luis	11512
Jarlin	11527
Elizabeth	11502
William	11565
11527	2.5
11550	3.5
11512	4.0
11565	0.0
11502	5.0

• Salida:

Elizabeth	5.0
Luis	4.0
Jarlin	2.5
Julian	3.5
William	0.0





M

- **Nombre:** Glosario
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/280/detalle/es>
- **Análisis:** Debemos leer N palabras, luego leer la palabra FIN. La tarea es ordenar las palabras léxico gráficamente, pero el problema es que algunas palabras al final tendrán ('.', ':', '?', '!'), entonces debemos eliminar eso, luego mostramos las palabras del glosario.
- **Implementación:** Usar un set, ya que ordena y guarda palabras únicas, en la entrada de las palabras eliminarles el último carácter si tiene algunos de los símbolos, luego mostrar '-' que indica el final del caso.

Ejemplo

Entradas:

1
11
hablo de programacion y de
programacion competitiva y
lenguajes de programacion.
FIN

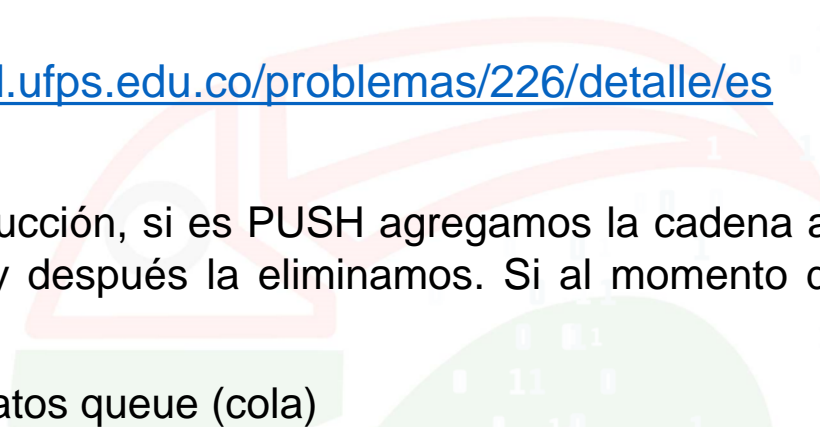
Salidas:

competitiva
de
hablo
lenguajes
programacion
y
-





N

- **Nombre:** Cola de strings
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/226/detalle/es>
- **Análisis:** Debemos leer una instrucción, si es PUSH agregamos la cadena a la cola. Si es pop, mostramos la cadena en el frente de la cola y después la eliminamos. Si al momento de hacer POP la cola esta vacía, mostramos '-'.

- **Implementación:** Estructura de datos queue (cola)

Entradas: 7

```
PUSH string1  
PUSH string2  
POP  
PUSH string3  
POP  
POP  
POP
```

Ejemplo

Salida:

```
String1  
String2  
String3  
-
```





- **Nombre:** JinJin
- **Link:** <https://trainingcenter.cloud.ufps.edu.co/problemas/224/detalle/es>
- **Análisis:** simular el juego JINJIN.
Si la instrucción es 1, el primer niño de la fila uno pasa de último a la fila dos
Si la instrucción es 2, el primer niño de la fila dos pasa al ultimo puesto de la fila uno.
si una fila esta vacía muestra '-'
- **Implementación:** Usar 2 queue(colas) , 2 list (listas) o deque (bicolos), ya que estas estructuras nos permiten acceder al primer dato y agregar en la ultima posición

Ejemplo

Entradas:

```
1 //casos
3 4 //3 niños, 4 instrucciones
juan david andres //Posiciones iniciales
1 1 2 1 //Instrucciones
```

Salida:

```
juan
david andres
```



