

EVALUACIÓN	Obligatorio	GRUPO	Todos	FECHA	30/8
MATERIA	Estructura de Datos y Algoritmos 2				
CARRERA	Ingeniería en Sistemas				
CONDICIONES	<p>- Puntos: Máximo: 30 Mínimo: 1</p> <p>LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40MB EN FORMATO ZIP, RAR O PDF.</p> <p>IMPORTANTE:</p> <ul style="list-style-type: none">- Inscribirse- Formar grupos de hasta dos personas.- Subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: "RECORDATORIO"				

Obligatorio

El obligatorio de Estructuras de Datos y Algoritmos 2, para el semestre 2 de 2022, está compuesto por un conjunto de **10** problemas a resolver.

El desarrollo del obligatorio:

- Se deberá realizar en grupos de **2 estudiantes**.
- Se podrá utilizar tanto C++ (C++11) como Java (jdk8).
Para asegurar que utilizan C++11 deberán tener instalado el compilador C++ de GNU e invocar la siguiente orden: **g++ -std=c++11 ...**
- No se puede usar ninguna librería, clase, estructura, función o variable no definida por el estudiante, a excepción de `<iostream>`, `<string>` y `<cstring>` para C++. y `System.in`, `System.out` y `String` para Java.
- Si se quiere usar algo que no sea lo listado anteriormente, consultar previamente al profesor.
- Se deberá utilizar el formato de ejecución trabajado durante el curso:
`miSolucion < input > output`
- La cátedra proveerá un conjunto de casos de prueba, pares de entrada y salida esperada. Se deberá comparar la salida de la solución contra una salida esperada.
- **De no cumplir las condiciones anteriores el ejercicio se considera invalido, por lo tanto 0 puntos.**

La entrega:

- El formato de entrega se encuentra definido en [este enlace](#).

La corrección:

- La corrección implica la verificación contra la salida esperada, así también como la corrección del código fuente para verificar que se cumplan los requerimientos solicitados (órdenes de tiempo de ejecución, usos de estructuras o algoritmos en particular, etc.).
- Se verificarán TODOS los casos de prueba. Si el programa no termina para un caso de prueba, puede implicar la pérdida de puntos.
- Si el código fuente se encuentra en un estado que dificulta la comprensión, podrá perder puntos.
- Los códigos dados en clase como algoritmos encontrados de otras fuentes serán usados como referencia. **Considerándose copia el uso de ellos.**
- **Se utilizará MOSS para detectar copias.**

La defensa:

- Luego de la entrega, se realizará una defensa de autoría a cada estudiante **de manera individual.**

1. Emails

Nombre de archivo: ejercicio1.cpp/Ejercicio1.java

Letra

Se tiene una base de datos de usuario identificados por e-mail pero se empezó a notar de que existen usuarios duplicados (comparten el mismo e-mail) por lo cual se quiere calcular cuantos usuarios reales existen.

Input

La primera línea de la entrada contiene un número entero positivo N ($2 \leq N \leq 10^5$), el número de usuarios actuales del sistema.

Las siguientes N líneas contienen los email actuales del sistema

Salida

Imprime el único número entero: la cantidad real de usuarios (sin repetidos).

Restricciones

- Utilizar una tabla de **hash cerrado**.
- Resolver en orden temporal: $O(N)$ promedio, siendo N la cantidad de usuarios.

Ejemplos de entrada y salida

Entrada

```
5
abc@abc.com
cba@cba.com
abc@abc.com
abc@abc.com
aaa@aaa.com
```

Salida

```
3
```

2. SORTING Nombre de archivo: ejercicio2.cpp/Ejercicio2.java

Letra

Se desea ordenar un conjunto de números enteros, no acotados. Se solicita que implemente un algoritmo basado en el uso de un heap, en particular mediante el uso de la técnica de “heapificación” (**heapify** en inglés).

Input

La primera línea de la entrada contiene un número entero positivo N ($2 \leq N \leq 10^5$), el largo del conjunto de números.

Las siguientes N líneas contienen los números del conjunto.

Salida

Imprime N líneas con los números ordenados de menor a mayor

Restricciones

- Utilizar la técnica **heapify**.
- Resolver en orden temporal: $O(N \log N)$ promedio, siendo el tamaño del conjunto.
- Resolver en orden espacial auxiliar (sin contar el array de números): $O(1)$

Ejemplos de entrada y salida

Entrada

```
5
10
4
7
42
3
```

Salida

```
3
4
```

7 10 42

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

Obligatorios (Cap.IV.1, Doc. 220)

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. La entrega se realizará desde gestion.ort.edu.uy
2. Previo a la conformación de grupos cada estudiante deberá estar inscripto a la evaluación.
Sugerimos realizarlo con anticipación.
3. **Uno de los integrantes del grupo de obligatorio será el administrador del mismo y es quien formará el equipo y subirá la entrega**
4. Cada equipo debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar)
5. El archivo a subir debe tener **un tamaño máximo de 40mb**
6. Les sugerimos **realicen una 'prueba de subida' al menos un día antes**, donde conformarán el **'grupo de obligatorio'**.
7. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
8. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc)
9. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor pasar por la oficina del Coordinador o por Coordinación adjunta **antes de las 20:00hs.** del día de la entrega

Si tuvieras una situación particular de fuerza mayor, debes dirigirte con suficiente antelación al plazo de entrega, al Coordinador de Cursos o Secretario Docente.