



## Tarea 4

### Análisis y Diseño de Algoritmos

#### Instrucciones para envío en Blackboard

Guarda tus archivos con el nombre según la siguiente regla:

- M<matrícula>pX.cpp, donde <matrícula>corresponde a los 6 dígitos de su matrícula UDEM y X al número de programa.
- Ejemplo: Si tu matrícula es 123456, el archivo para el problema 2, se deberá llamar M123456p2.cpp o .java
- Incluye el Código de Honor en cada programa.
- Si se realizan los programas en pareja, sólo un estudiante deberá subir el archivo nombrado con su matrícula. Importante poner **ambas matrículas y nombres** en el código fuente y/o en los comentarios al subir los programas en Blackboard.
- **\*\*Nota:** Sino sigues esta regla, tus programas NO serán calificados.

*Resolver los siguientes problemas de programación. Se sugiere, pero no se limita a resolverlos con la técnica de algoritmos voraces.*

## Problem 1. *Servidoreess*

Standard Input

Time limit: 1 seconds

Una empresa de software utiliza servicios en la nube para el procesamiento de  $n$  tareas matutinas y  $n$  tareas vespertinas. Cada tarea requiere una cantidad específica de recursos computacionales, medidos en unidades de procesamiento.

La empresa cuenta con  $n$  servidores, y cada servidor debe encargarse de una tarea en la mañana y otra en la tarde. Cada servidor puede procesar hasta  $u$  unidades de trabajo en total sin costo adicional por parte del proveedor.

Sin embargo, si la carga total de ambas tareas asignadas a un servidor excede este límite, se genera un costo adicional por cada unidad extra procesada. Este costo se cobra a razón de  $d$  dólares por unidad adicional.

Tu objetivo es asignar las tareas matutinas y vespertinas a los servidores de forma que el costo total adicional que la empresa deba pagar al proveedor sea el menor posible.

### Input

- La primera línea de cada caso de prueba contiene tres números enteros:  $n$  ( $1 \leq n \leq 1000$ ) : Número de servidores disponibles (y también el número de tareas matutinas y vespertinas);  $u$ : ( $1 \leq u \leq 10^5$ ) Capacidad diaria máxima de procesamiento por servidor sin costo adicional;  $d$  ( $1 \leq d \leq 1000$ ): Costo en dólares por cada unidad adicional que supere el límite.
- La segunda línea contiene  $n$  números enteros separados por espacios, que representan las cargas de trabajo de las tareas matutinas (en unidades de procesamiento).
- La tercera línea contiene  $n$  números enteros separados por espacios, que representan las cargas de trabajo de las tareas vespertinas (en unidades de procesamiento).

### Output

Para cada caso de prueba, imprime el costo mínimo total que la empresa deberá pagar por concepto de unidades de procesamiento adicionales.

### Sample Input

```
2 30 10
15 17
15 17
3 10 20
2 4 6
8 10 10
0 0 0
```

### Sample Output

```
40
200
0
```

## Problem 2. *DJ*

Standard Input

Time limit: 1 seconds

Cierto DJ famoso siempre prepara su playlist para sus conciertos de una manera muy peculiar. En su consola tiene acceso a una biblioteca de **loops musicales** pregrabados, cada uno con una duración específica en minutos.

Estos loops son **indivisibles**, es decir, no se pueden cortar en partes más pequeñas. Sin embargo, el DJ puede repetir un mismo loop varias veces si es necesario para completar su playlist.

El DJ puede elegir libremente cuántos minutos quiere que dure su presentación, siempre que pueda construir ese tiempo utilizando los loops disponibles siguiendo un proceso específico, siguiendo las siguientes reglas:

- El DJ selecciona el **loop más largo que pueda caber** en el tiempo restante que tiene, digamos  $X$ .
- Ese loop se agrega a la playlist (no importa que ya haya sido utilizado antes).
- Este tiempo se descuenta del total disponible  $X$ .
- El DJ repite este proceso con el tiempo restante hasta que no pueda agregar más loops.

El DJ puede elegir cualquier cantidad de minutos para su presentación, pero su objetivo es maximizar la cantidad de loops diferentes que pueda incluir en una sola playlist. ¿Puedes ayudarlo?

### Input

La entrada comienza con una línea que contiene un número entero  $n$  ( $1 \leq n \leq 1000$ ) que representa el número total de loops disponibles.

La siguiente línea contiene  $n$  números enteros separados por espacios que indican la duración en minutos de cada uno de los loops musicales. Puedes asumir que cada número es diferente y con valores entre 1 y  $10^9$ .

El final de la entrada se indica con una línea que contiene un cero.

### Output

Para cada caso de prueba, se debe imprimir un solo número entero que represente el máximo número de loops musicales diferentes que el DJ puede incluir en su playlist.

### Sample Input

```
6
8 1 16 4 32 2
6
1 3 6 20 8 15
0
```

### Sample Output

```
6
4
```