

Problema 1

(<https://olimpiada-informatica.madrid/archive/2021/OIM2021.pdf>)



¿En qué volumen?

En muchos jueces on-line (*¡Acepta el reto!* entre ellos) cada problema tiene un identificador único para poderlo referenciar de manera unívoca dentro del sistema. Los identificadores son números naturales correlativos, y el primer problema recibe el número 100.

Empezar en 100, en lugar de en 1 (o en 0), no es un capricho. Los problemas se “archivan” en volúmenes, cada uno compuesto por 100 problemas. Al asignar el número 100 al primer problema, es fácil saber en qué volumen está cualquier problema a partir de su identificador. En concreto, el primer volumen de problemas contiene a aquellos que tienen como identificador los números entre 100 y 199, el volumen 2 contiene los problemas con identificadores 200...299, etcétera.



Dado un problema, ¿en qué volumen está?

Entrada

La entrada comienza con un número que indica cuántos casos de prueba vendrán a continuación. Cada uno será un número entre 100 y 999.999.

Salida

Para cada caso de prueba, el programa deberá escribir a qué volumen pertenece el problema con ese identificador.

Entrada de ejemplo

```
2
100
306
```

Salida de ejemplo

```
1
3
```

Problema 2

(<https://www.oia.unsam.edu.ar/wp-content/uploads/2023/04/cadenas1.pdf>)

Cadenas de divisores

Contribución de Matías Raimundez

Descripción del problema

Agustín estaba pensando en el número 60, que es un número con muchos divisores. De repente se le ocurrió crear una cadena de divisores, en la que el primer elemento sea 60 y a partir de ahí cada elemento sea un divisor del anterior.

Por ejemplo, una posible cadena sería 60 - 20 - 10 - 2.

A Agustín le interesa que la suma de los elementos de la cadena sea lo más grande posible. Probó todas las cadenas de divisores que comienzan en 60, y descubrió que la cadena de divisores encabezada por 60 **con la mayor suma posible de sus elementos** es 60 - 30 - 15 - 5 - 1, cuya suma es 101.

Agustín comenzó con un número $a = 60$ en el ejemplo anterior, pero la cadena podría comenzar con cualquier entero positivo a .

Debes escribir una función que calcule la máxima suma posible en una cadena de divisores, dado el número inicial a de la cadena.

Descripción de la función

Debes implementar la función `cadenas(a)`

El único parámetro indica el número inicial a de la cadena. La función debe retornar la máxima suma posible de los elementos de una cadena que comience con el número a .

Evaluador

El evaluador local lee de la entrada estándar un único entero, que indica el parámetro a para la función `cadenas`.

El evaluador local escribe a la salida estándar un único entero, con el valor retornado por la llamada `cadenas(a)`.

Restricciones

- $1 \leq a \leq 1.000.000$

Ejemplo

Si se invoca al evaluador con la siguiente entrada:

60

Para un programa correcto, la salida será:

101

Si en cambio la entrada es:

33

Para un programa correcto, la salida será:

45

Subtareas

1. $a \leq 10$ (10 puntos)
2. $a \leq 5.000$ y a tiene a lo sumo 4 divisores (15 puntos)
3. a tiene a lo sumo 4 divisores (20 puntos)
4. Sin más restricción (55 puntos)

Problema 3

(<https://olimpiada-informatica.madrid/archive/2021/OIM2021.pdf>)

La carnicería de mi barrio trae el mejor producto de la zona, por lo que siempre está llena de gente. Claudio, el dueño, hace tiempo que decidió colocar un expendedor de números para poder atender a los clientes ordenadamente y sin que las discusiones entre ellos perturbasen su quehacer diario. Como no tenía suficiente dinero, no ha podido comprar además un contador digital, así que confía en que los propios clientes formen una cola respetando la numeración y no suele pedir el número cuando les atiende; circunstancia que estos han aprovechado para colarse en numerosas ocasiones.



Por ello, le ha pedido a su hija Claudia que haga una inspección sorpresa diaria. En el momento de la inspección, Claudia les pide a todos los que están en ese momento en la cola que enseñen su número y a los que está segura de que se han colado los expulsa de la tienda. Luego, cuando cierran la tienda, le cuenta a su padre a cuántos ha expulsado ese día.

Entrada

La primera línea contiene un número que indica el número de casos de prueba que aparecen a continuación.

Cada caso de prueba se compone de dos líneas. En la primera aparece un único entero con el número de personas que están en la cola cuando Claudia hace la inspección (entre 1 y 500.000). En la segunda aparecen los números que tienen cada una de esas personas según están en la cola, números entre 1 y 10^6 , todos distintos (nadie ha llegado aún hasta el extremo de falsificar los números). Eso sí, pueden faltar números, porque hay personas que ya han sido atendidas o personas que se cansaron de esperar y se marcharon.

Salida

Para cada caso de prueba, el programa escribirá el número de personas que Claudia ha expulsado.

Entrada de ejemplo

```
2
5
2 3 6 7 9
7
1 4 3 2 5 7 6
```

Salida de ejemplo

```
0
3
```

Problema 4

(<https://olimpiada-informatica.madrid/archive/2021/OIM2021.pdf>)



Comienza la temporada

En el club deportivo del colegio les gusta aprovechar las equipaciones de los chicos de una temporada para la siguiente. De esta forma tienen más dinero para el mantenimiento de las instalaciones y no necesitan incrementar las cuotas. Al final de temporada, todos los jugadores deben devolver al club sus equipaciones, debidamente lavadas, planchadas y retirado el dorsal. Al comenzar la siguiente temporada el club toma nota de las tallas que necesitan los chicos y procede a repartir las equipaciones entre ellos, que volverán a poner el dorsal que les corresponda.



Al comenzar la temporada tenemos alumnos que se incorporan nuevos, en concreto los del primer curso de alevín, y normalmente los jugadores del curso anterior han cambiado de talla. Por este motivo es raro el año en que no hay que comprar alguna nueva equipación. Desde el club deportivo quieren optimizar sus recursos y nos han pedido que calculemos el mínimo número de equipaciones que han de comprarse para que todos los alumnos puedan jugar, teniendo en cuenta que si no hay suficientes equipaciones de una talla los alumnos pueden utilizar también las de una talla más. Irán un poco más holgados, pero no les impide moverse ni se les llega a caer como ocurriría con las tallas más pequeñas o las todavía más grandes.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con una línea en que se indica el número N de jugadores de esta temporada y el número M de equipaciones con las que contamos ($1 \leq N, M \leq 100.000$). A continuación aparece dos líneas: la primera tiene N valores con la talla que necesita cada chico y la segunda tiene M valores con las tallas de las equipaciones con las que contamos. Las tallas son valores entre 1 y 100.

Salida

Para cada caso de prueba se escribe en una línea el mínimo número de equipaciones que debemos comprar.

Entrada de ejemplo

```
5 4
12 12 12 12 12
13 14 12 10
3 4
13 10 12
15 11 12 14
3 3
10 14 17
6 4 50
```

Salida de ejemplo

```
3
0
3
```


Problema 5

(<https://www.oia.unsam.edu.ar/wp-content/uploads/2023/04/tejedores.pdf>)

Tejiendo paños

Contribución de Natalia Perez

Descripción del problema

Para festejar el aniversario de una ciudad, la municipalidad quiere crear una gran bandera, formada por N paños consecutivos, que se disponen en orden de izquierda a derecha (según la parte de la bandera a la que corresponden) y se numeran desde 0 hasta $N - 1$, en orden de izquierda a derecha. El tiempo t_i que toma completar cada paño varía entre los distintos paños, por tener cada uno distinta complejidad.

Para la tarea se contrataron K tejedores, que pueden trabajar todos al mismo tiempo para completar la bandera lo más rápido posible. El problema es que **cada tejedor puede tejer solamente paños consecutivos**. Así por ejemplo, si a un tejedor se le asignan tanto el paño 10 como el paño 14, entonces necesariamente también debe encargarse de los paños 11, 12 y 13. Por más que un tejedor termine su tarea antes que los demás, no puede asistir al resto para tratar de terminar antes, porque se generaría mucha confusión y en el desorden se podría arruinar la bandera.

Debes implementar una función que dada la cantidad K de tejedores y el tiempo t_i que se tarda en tejer de cada paño, determine el menor tiempo posible para terminar la bandera, si la asignación de paños a tejedores se hace de manera óptima respetando las restricciones indicadas.

Descripción de la función

Debes implementar la función `tejedores(K, t)`

Sus parámetros son:

- K : Cantidad de tejedores
- t : Arreglo de N enteros, que indican los tiempos que toma completar cada paño. El paño i toma tiempo $t[i]$.

La función debe retornar en un entero, el mínimo tiempo posible para completar la bandera.

Evaluador local

El evaluador lee de la entrada estándar:

- Una línea con N , y K
- Una línea con los N números del arreglo t

El evaluador escribe a la salida estándar el entero retornado por la llamada `tejedores(K, t)`.

Restricciones

- $1 \leq K \leq N \leq 100.000$
- $1 \leq t_i \leq 1.000.000$