

Ejercicio 1: Utilizar y modificar los algoritmos

Su hermano chico tomó un curso de programación en el colegio y tiene una tarea que le entregó su profesora. La tarea consiste en ordenar y buscar elementos. Tú decides ayudarlo, emocionado o con tus conocimientos aprendidos en el taller OCILabs.

Primera pregunta

Utilizando la función implementada en C++, tú debes ordenar un arreglo de números en orden mayor a menor.

En la primera línea se encuentra un entero n , correspondiente a la cantidad de elementos en la secuencia. Luego, en la siguiente línea se encuentra n enteros separados por un espacio.

Input de ejemplo

```
1 10
2 3 2 10 -2 5 8 7 1 2 -3
```

Segunda pregunta

Ahora que sabes como ordenar una función utilizando la función de C++, utilice el código de la función descrita en la cápsula sobre búsqueda binaria de un buen taller llamado OCILabs, para poder buscar si es que algún elemento esta o no.

NOTA: en la cápsula se enseña como buscar elementos en un arreglo ordenado de menor a mayor, modifique la condición del código para poder buscar un elemento en un arreglo ordenado de mayor a menor.

En la primera línea se encuentra un entero n , correspondiente a la cantidad de elementos en la secuencia. Luego, en la siguiente línea se encuentra n enteros separados por un espacio.

Por último, se encuentra un entero que corresponde al elemento a buscar.

Input de ejemplo

```
1 10
2 3 2 10 -2 5 8 7 1 2 -3
3 5
```

Muestre por pantalla, “Si, si esta el elemento” si el elemento esta en el arreglo. Muestre por pantalla “No, no esta” si el elemento no esta en el arreglo.

Output de ejemplo

```
1 Si, si esta el elemento
```

C - Find the Median

La mediana de una lista de números es esencialmente su elemento medio después de ordenar. El mismo número de elementos aparece después que antes. Dada una lista de números con un número impar de elementos, encuentre la mediana?

Por ejemplo,

$arr = [5, 3, 1, 2, 4]$

El arreglo ordenado sería $[1, 2, 3, 4, 5]$. El elemento de la mitad y la mediana es 3.

Input

La primera línea contiene el entero n , del tamaño de arr .

La segunda línea contiene n enteros separados por espacios arr .

Restricciones

- $1 \leq n \leq 1000001$
- n es impar
- $-10000 \leq arr[i] \leq 10000$

Output

Un entero que corresponda a la mediana del arreglo.

Ejemplo

Input

```
1 5
2 5 3 1 2 4
```

Output

```
1 3
```

B - Chef and Eid

Durante el Eid, es una tradición que cada padre les dé a sus hijos dinero que pueden gastar en entretenimiento.

Chef tiene N monedas denotemos el valor de la moneda i por v_i . Dado que hoy es Eid, Chef le dará una moneda a cada uno de sus dos hijos. Quiere que el valor absoluto de la diferencia entre los valores de las monedas entregadas a los dos niños sea lo más pequeño posible, para que él sea lo más justo posible.

Ayude al Chef diciéndole la mínima diferencia posible entre los valores de las monedas entregadas a los dos niños. Por supuesto, el Chef no puede dar la misma moneda a ambos niños.

Input

- La primera línea de la entrada contiene un solo entero T que indica el número de casos de prueba. A continuación, se muestra la descripción de los casos de prueba T .
- La primera línea de cada caso de prueba contiene un único entero N .
- La segunda línea contiene N enteros separados por espacios v_1, v_2, \dots, v_N .

Output

Por cada caso de prueba, muestre por pantalla una única línea que contenga un entero - la menor diferencia posible.

Restricciones

- $1 \leq T \leq 100$
- $2 \leq N \leq 10^5$
- la suma de N en todos los casos no excede $5 \cdot 10^5$
- $1 \leq v_i \leq 10^6$ por cada i válido

Subtareas

Subtarea #1 (30 puntos): la suma de N en todos los casos no excede 2000

Subtarea #2 (70 puntos): las restricciones originales

Ejemplo

Input

```
1 2
2 3
3 1 4 2
4 3
5 1 3 3
```

Output

```
1 1
2 0
```

Explicación

Ejemplo caso 1: Chef le da la moneda con valor 1 a su primer hijo y la moneda con valor 2 al segundo hijo, por lo que la respuesta es $2 - 1 = 1$.

Ejemplo caso 2: El chef le da a cada uno de sus hijos una moneda de valor 3, por lo que la diferencia es 0.

HINT: Ordene los elementos primero.

C - Where is the Marble?

A Raju y Meena les encanta jugar con las canicas. Tienen muchas canicas con números escritos en ellas. Al principio, Raju colocaba las canicas una tras otra en orden ascendente de los números escritos en ellas. Entonces Meena le pedía a Raju que encontrara la primera canica con un número determinado. Contaría 1 ... 2 ... 3. Raju obtiene un punto por respuesta correcta, y Meena obtiene el punto si Raju falla. Después de un número fijo de intentos, el juego termina y el jugador con el máximo de puntos gana. Hoy es tu oportunidad de jugar como Raju. Siendo el niño inteligente, estarías tomando el favor de una computadora. Pero no subestimes a Meena, ella había escrito un programa para llevar un registro de cuánto tiempo estás tomando para dar todas las respuestas. Así que ahora tienes que escribir un programa que te ayudará en tu papel de Raju.

Input

Puede haber varios casos de prueba. El número total de casos de prueba es menor que 65. Cada caso de prueba consta de 2 números enteros: N el número de canicas y Q el número de consultas que haría Mina. Las siguientes N líneas contendrían los números escritos en las N canicas. Estos números de canicas no vendrán en ningún orden en particular. Las siguientes líneas de preguntas tendrán consultas. Tenga la seguridad de que ninguno de los números de entrada es mayor que 10000 y ninguno de ellos es negativo.

El input termina cuando el caso de prueba sea $N = 0$ y $Q = 0$.

Output

Para cada caso de prueba, envíe el número de serie del caso.

Para cada una de las consultas, muestra por pantalla una línea. El formato de esta línea dependerá de si el número de consulta está escrito o no en alguna de las canicas. Los dos formatos diferentes se describen a continuación:

- “x found at y”, si la primera canica con el número x se encontraba en position y . Las posiciones están numeradas 1, 2, ..., N .
- “x not found”, si la canica con el número x no está presente.

Mire la salida de entrada de muestra para obtener detalles.

Ejemplo

Input

```
1 4 1
2 2
3 3
4 5
5 1
6 5
7 5 2
8 1
9 3
10 3
11 3
12 1
13 2
14 3
15 0 0
```

Output

```
1 CASE# 1:  
2 5 found at 4  
3 CASE# 2:  
4 2 not found  
5 3 found at 3
```

D - Aggressive cows

El granjero John ha construido un granero largo nuevo, con N ($2 \leq N \leq 100000$) puestos. Los puestos están ubicados a lo largo de una línea recta en las posiciones x_1, \dots, x_N ($0 \leq x_i \leq 1000000000$).

A sus vacas C ($2 \leq C \leq N$) no les gusta este diseño de establo y se vuelven agresivas unas con otras una vez que las colocan en un establo. Para evitar que las vacas se lastimen entre sí, FJ quiere asignar las vacas a los establos, de modo que la distancia mínima entre dos de ellas sea la mayor posible. ¿Cuál es la distancia mínima más grande?

Input

- t : el número de casos de prueba, luego sigue t casos de prueba.
- Línea 1: Dos enteros separados por espacios: N y C
- Líneas 2.. N +1: La línea i +1 contiene una ubicación de pérdida de números enteros, x_i

Output

Para cada caso de prueba, muestre por pantalla un número entero: la distancia mínima más grande.

Ejemplo

Input

```
1 1
2 5 3
3 1
4 2
5 8
6 4
7 9
```

Output

```
1 3
```