

## GPC-UNI Resto del Mundo - Contest 03

04 de marzo de 2020

## Problem A. Buenos dominos

Author: racsosabe  
Input file: standard input  
Output file: standard output  
Time limit: 4 seconds  
Memory limit: 256 megabytes

Los dominos son piezas con una cara partida a la mitad (una matriz de  $1 \times 2$ ) que tiene enteros  $a_i$  en el lado izquierdo y  $b_i$  en el lado derecho.

La suma de dígitos de un número  $x$  en base  $b \geq 2$  es la suma de sus dígitos cuando este es escrito en base  $b$ .

La raíz digital de un número  $x$  en base  $b \geq 2$  se obtiene al reemplazarlo repetidamente por su su suma de dígitos en base  $b$  hasta que halla solo un número de un dígito en base  $b$ . La denotaremos como  $f_b(x)$ .

Por ejemplo, la raíz digital de 40 en base 3 es 2:

$$40 = \overline{1111}_{(3)} \rightarrow 4 = \overline{11}_{(3)} \rightarrow 2 = \overline{2}_{(3)}$$

Decimos que un par  $(k, b)$  es **bueno** si y solo si:

$$\{f_b(i \cdot k), \forall i = 0, 1, \dots, (b - 1)\}$$

es una permutación de  $(0, 1, \dots, (b - 1))$ .

Artem tiene  $n$  dominos, los cuales él puede rotar de  $(a_i, b_i)$  a  $(b_i, a_i)$ .

Tu tarea es calcular la máxima cantidad de buenos pares que se pueden obtener mediante la posible rotación de algunos dominos y obtener alguna distribución óptima.

### Input

La primera línea de entrada contiene un entero  $n$  ( $1 \leq n \leq 3 \cdot 10^5$ ) — La cantidad de dominos que tiene Artem.

Las siguientes  $n$  líneas de entrada contienen dos enteros  $a_i$  y  $b_i$  ( $1 \leq a_i, b_i \leq 10^{15}$ ) — El  $i$ -ésimo domino tiene los valores  $a_i$  y  $b_i$  en sus lados.

### Output

En la primera línea imprime la máxima cantidad de buenos pares que se pueden obtener.

Luego, imprime  $n$  líneas con la distribución óptima de los dominos:

En la  $i$ -ésima línea, imprime los valores  $a'_i$  y  $b'_i$ , los cuales deben ser el resultado de rotar (o no) los valores  $a_i$  y  $b_i$ .

**Advertencia:** Recuerda que el juez solo verificará si  $(a'_i, b'_i)$  es un buen par, así que ten cuidado con tu formato de salida.

### Scoring

El puntaje estará dividido en los siguientes subtasks:

#### Subtask 1 (10 puntos):

- $1 \leq a_i, b_i \leq 10$

#### Subtask 2 (25 puntos):

- $1 \leq a_i, b_i \leq 1000$

**Subtask 3 (65 puntos):**

- $1 \leq a_i, b_i \leq 10^{15}$

Para todos los subtasks se cumplirá que:

- $1 \leq n \leq 3 \cdot 10^5$

**Examples**

standard input	standard output
2	2
10 7	7 10
7 10	7 10
1	1
4 8	8 4

**Note**

Para el primer caso de prueba, notemos que  $(10, 7)$  no es un buen par:

$$f_7(0) = 0$$

$$f_7(10) = 4$$

$$f_7(20) = 2$$

$$f_7(30) = 6$$

$$f_7(40) = 4$$

$$f_7(50) = 2$$

$$f_7(60) = 6$$

Lo cual no es una permutación de  $\{0, 1, 2, 3, 4, 5, 6\}$ .

Sin embargo,  $(7, 10)$  sí es un buen par:

$$f_{10}(0) = 0$$

$$f_{10}(7) = 7$$

$$f_{10}(14) = 5$$

$$f_{10}(21) = 3$$

$$f_{10}(28) = 1$$

$$f_{10}(35) = 8$$

$$f_{10}(42) = 6$$

$$f_{10}(49) = 4$$

$$f_{10}(56) = 2$$

$$f_{10}(63) = 9$$

Que sí es una permutación de  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ .

## Problem B. Testing Cuidadoso

Author: racsosabe  
Input file: standard input  
Output file: standard output  
Time limit: 5 seconds  
Memory limit: 512 megabytes

racsosabe está muy ocupado seteando los problemas para el “GPC-UNI Resto del Mundo Contest - 03”. Esta vez está testeando las soluciones para uno de los problemas.

El problema es sobre obtener sumas en rango de un arreglo estático (los valores no cambiarán durante las consultas). Quien propuso este problema fue TISparta y su solución modelo usa, dado que él es un amante de las estructuras de datos, una estructura de datos llamada *Fenwick Tree*.

El Fenwick Tree responde a una consulta en rango usando como subrutina a la función  $getSum(pos)$ , la cual obtiene la suma de todos los valores en el rango de índices  $[0, pos]$ . Así, la función  $query(L,R)$  devuelve  $getSum(R) - getSum(L-1)$ .

El pseudocódigo de la función  $getSum$  es el siguiente:

```
long long getSum(int pos){
    pos += 1;
    long long res = 0LL;
    while(pos > 0){
        res += ft[pos];
        pos &= pos-1;
    }
    return res;
}
```

Sin embargo, TISparta se olvidó cómo era esta función y usó la siguiente:

```
long long getSum(int pos){
    pos += 1;
    long long res = 0LL;
    while(pos > 0){
        res += ft[pos];
        pos -= 1<<(31 - __builtin_clz(pos));
    }
    return res;
}
```

Dados los valores actuales del arreglo  $ft$  de tamaño  $(n + 1)$  y todas las consultas que la solución modelo debe responder, debes obtener el número de respuestas incorrectas entre todos los casos.

**Puedes asumir que todos los valores  $ft_i$  han sido obtenidos de manera correcta.**

### Input

La primera línea de entrada contiene un entero  $n$  ( $1 \leq n \leq 10^5$ ) — La cantidad de elementos en el arreglo original  $a$ , el tamaño del arreglo  $ft$  del Fenwick Tree es  $(n + 1)$ .

La segunda línea contiene  $(n + 1)$  enteros  $ft_i$  ( $0 \leq |ft_i| \leq 10^9$ ) — El  $i$ -ésimo entero es el  $i$ -ésimo valor del arreglo del Fenwick Tree.

La tercera línea contiene un entero  $q$  ( $1 \leq q \leq 10^5$ ) — La cantidad de consultas que deben ser analizadas.

Las siguientes  $q$  líneas contienen dos enteros  $l_i$  y  $r_i$  ( $0 \leq l_i \leq r_i < n$ ) — La  $i$ -ésima línea contiene los límites de la  $i$ -ésima consulta, para la cual uno debe responder la suma de todos los valores  $a_l, \dots, a_r$  del arreglo original.

## Output

Imprime una sola línea — La cantidad de consultas incorrectas usando el código de TISparta.

## Scoring

El puntaje estará dividido en los siguientes subtasks:

### Subtask 1 (40 puntos):

- $1 \leq n, q \leq 10^4$

### Subtask 2 (60 puntos):

- $1 \leq n, q \leq 10^5$

Para todos los subtasks se cumplirá que:

- $0 \leq |ft_i| \leq 10^9$

## Example

standard input	standard output
5 0 1 -9 2 4 7 3 0 3 2 4 4 4	2

## Problem C. Tercio Palindrómico

Author: racsosabe  
Input file: standard input  
Output file: standard output  
Time limit: 1 second  
Memory limit: 256 megabytes

Lord El-Melloi II es un mago detective. Él estaba intentando resolver un misterio sobre un mensaje encriptado en la casa de su pupila Gray. Ellos encontraron una cadena de caracteres que olo consistia de letras minúsculas.

Ellos tuvieron que hacer un ritual en el que la cadena original sería transformada en otra que solo tenga las letras  $a$ ,  $b$  y  $c$ . Además, cada par de letras consecutivas son diferentes.

Para el paso final, se requería que ellos determinaran una subsecuencia palíndroma de la cadena anterior, pero con la siguiente condición:

- Sea la cadena  $s$  y la subsecuencia palíndroma  $t$ , entonces:

$$|t| \geq \left\lceil \frac{|s|}{3} \right\rceil$$

Ayuda a Lord El-Melloi II y a Gray a encontrar alguna subsecuencia válida para resolver el misterio.

**Definición:** Una subsecuencia  $t$  de una cadena  $s$  es una cadena que se puede obtener eliminando algunas posiciones de  $s$  y concatenando las que no fueron eliminadas de izquierda a derecha.  $abc$  es subsecuencia de  $adbca$ , pero  $abcd$  no lo es.

### Input

La primera línea de entrada contiene una cadena  $s$  ( $3 \leq |s| \leq 10^5$ ) — La cadena del último paso que consiste solo de letras  $a$ ,  $b$  y  $c$ .

### Output

Imprime una sola línea — Una subsecuencia palíndroma válida. Está garantizado que existe una respuesta.

### Scoring

El puntaje estará dividido en los siguientes subtasks:

#### Subtask 1 (15 puntos):

- $1 \leq |s| \leq 100$

#### Subtask 1 (35 puntos):

- $1 \leq |s| \leq 10^4$

#### Subtask 1 (50 puntos):

- $1 \leq |s| \leq 10^5$

### Example

standard input	standard output
abc	c

## Note

Para el caso de prueba, necesitamos una subsecuencia palíndroma de longitud no menor a 1, así que las posibles respuestas son:

- $a$
- $b$
- $c$

## Problem D. Semana Deportiva UNI

Author: racsosabe  
Input file: `standard input`  
Output file: `standard output`  
Time limit: 1 second  
Memory limit: 256 megabytes

¡La semana deportiva ha llegado! Hay  $n$  facultades (identificadas con ids desde 1 hasta  $n$ ) participando en un Torneo de Voleibol. El torneo se lleva a cabo usando Formato Liga, así que cada facultad tiene un partido contra todas las demás en el coliseo (así que no hay equipo local o visitante) y los resultados finales se obtienen usando los puntos ganados y la cantidad de sets ganados.

Por simplicidad (y debido a que no tenemos suficiente dinero para pagarle a los árbitros), nuestros partidos de voley los gana el mejor de 3 sets. De esa manera, solo hay 2 posibles resultados ( $2-0$  y  $2-1$  en sets). Finalmente, el ganador obtiene 2 puntos y el perdedor solo 1.

El ranking final se da de acuerdo a la siguiente prioridad:

1) Puntos acumulados.

2) Cociente:  $\frac{\text{Sets ganados}}{\text{Sets perdidos}}$

El **Campeonato** se le entrega a todos los equipos **tales que no existe otro equipo que tiene un resultado estrictamente mejor que ellos**.

Han habido  $m$  partidos jugados hasta ahora, así que deseas saber cuántos resultados generales diferentes existen de forma que tu equipo obtenga el Campeonato, solo para saber si seguirás apoyando o no.

### Input

La primera línea de entrada contiene 3 enteros  $n$ ,  $w$  y  $m$  ( $2 \leq n \leq 16, 1 \leq w \leq n, 0 \leq \frac{n(n-1)}{2} - m \leq 10$ )  
— La cantidad de facultades, el id de tu facultad y la cantidad de partidos jugados hasta ahora.

Las siguientes  $m$  líneas contienen 4 enteros  $a_i$ ,  $b_i$ ,  $pa_i$  and  $pb_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i, 0 \leq pa_i, pb_i \leq 2$ )  
— Los ids de las facultades que jugaron el  $i$ -ésimo partidos y los sets ganados por cada una de ellas, respectivamente.

### Output

Imprime una sola línea — “Just go home” si no existe ningún resultado general que le dé el Campeonato a tu facultad o la cantidad de dichos resultados si existe alguno.

### Scoring

El puntaje estará dividido en los siguientes subtasks:

**Subtask 1 (100 puntos):**

- $2 \leq n \leq 16$
- $1 \leq w \leq n$
- $0 \leq \frac{n(n-1)}{2} - m \leq 10$

### Example

standard input	standard output
3 1 2 1 2 2 0 2 3 0 2	2



## Note

Para el caso de prueba, notemos que los rankings actuales son:

Facultad	Puntos	Sets Ganados	Sets Perdidos
1	2	2	0
3	2	2	0
2	2	0	4

Y solo hay 1 partido restante, entre 1 y 3, así que si queremos que 1 obtenga el campeonato, solo hay 2 posibles resultados: Vencer 2 – 0 o vencer 2 – 1.

## Problem E. Emergencia XOR

Author: racsosabe  
Input file: `standard input`  
Output file: `standard output`  
Time limit: 3 seconds  
Memory limit: 512 megabytes

Knuckles ha aprendido Bitwise XOR ( $\wedge$  in C++) en la clase de hoy; dado que él es muy bueno en matemática, él puede calcular la respuesta para cualquier conjunto de números en un abrir y cerrar de ojos. Su profesor quiso darle un reto, así que le dio un conjunto de números distintos y algunas consultas. Luego le preguntó, por cada consulta, la suma de los Bitwise XOR de los elementos para cada subconjunto de la consulta dada.

Como tú conoces el camino, él te pidió tu ayuda en este problema.

**Nota:** El Bitwise XOR del conjunto vacío es 0.

### Input

La primera línea de entrada contiene 2 enteros  $n$  y  $q$  ( $1 \leq n \leq 18, 1 \leq q \leq 10^5$ ) — La cantidad de valores y la cantidad de consultas.

La segunda línea de entrada contiene  $n$  enteros  $a_i$  ( $0 \leq a_i \leq 10^9$ ) — El  $i$ -ésimo entero es el  $i$ -ésimo valor.

Las siguientes  $q$  líneas contienen una cadena  $s$  ( $|s| = n, s_i \in \{0, 1\}$ ) — El  $i$ -ésimo carácter es 1 si el  $i$ -ésimo número debe ser considerado en la consulta y 0 en caso contrario.

### Output

Imprime  $q$  líneas — La  $i$ -ésima línea debe tener la respuesta a la  $i$ -ésima consulta.

### Scoring

El puntaje estará dividido en los siguientes subtasks:

#### Subtask 1 (30 puntos):

- $1 \leq n \leq 9$

#### Subtask 2 (70 puntos):

- $1 \leq n \leq 18$

Para todos los subtasks, se cumple:

- $1 \leq q \leq 10^5$
- $0 \leq a_i \leq 10^9$

### Example

standard input	standard output
3 3	12
1 2 3	6
111	1
101	
100	

## Note

Para la primera consulta, tenemos  $\{1, 2, 3\}$ , el cual tiene los subconjuntos  $\{1, 2, 3\}$ ,  $\{1, 2\}$ ,  $\{1, 3\}$ ,  $\{2, 3\}$ ,  $\{1\}$ ,  $\{2\}$ ,  $\{3\}$  y  $\phi$ , su suma es:

$$0 + 3 + 2 + 1 + 1 + 2 + 3 + 0 = 12$$

Para la segunda consulta, tenemos  $\{1, 3\}$ , el cual tiene los subconjuntos  $\{1, 3\}$ ,  $\{1\}$ ,  $\{3\}$  y  $\phi$ , su suma es:

$$2 + 1 + 3 + 0 = 6$$

Para la tercera consulta, tenemos  $\{1\}$ , el cual tiene los subconjuntos  $\{1\}$  y  $\phi$ , su suma es:

$$1 + 0 = 1$$

## Problem F. Inki

Author: Valgt  
Input file: `standard input`  
Output file: `standard output`  
Time limit: 4 seconds  
Memory limit: 256 megabytes

Inki es un cazador que se está preparando para el invierno, por lo que necesita almacenar la mayor cantidad de carne posible. Para poder llegar a la zona de caza el tiene que cruzar un río y la única forma de hacerlo es a través de un viejo puente colgante compuesto por  $n$  tablas, la  $i$ -ésima tabla tiene una capacidad  $c_i$  que es el peso extra que puede llevar consigo Inki sin que la tabla se rompa.

Además Inki tiene  $m$  monedas que puede usar en reparar algunas tablas. El puede reparar la  $i$ -ésima tabla y aumentar su capacidad en 1 gastando  $k_i$  monedas.

Ayuda a Inki a usar sus monedas sabiamente, para que pueda conseguir la mayor cantidad de carne posible.

### Input

La primera línea de entrada contiene dos enteros  $n$  y  $m$  ( $1 \leq n \leq 3 \cdot 10^5, 0 \leq m \leq 10^{12}$ ) — La cantidad de tablas que tiene el puente colgante y las monedas que tiene Inki.

La segunda línea contiene  $n$  enteros  $c_i$  ( $1 \leq c_i \leq 10^5$ ) — La capacidad de las  $n$  tablas.

La tercera línea contiene  $n$  enteros  $k_i$  ( $1 \leq k_i \leq 10^5$ ) — El costo de las  $n$  tablas, puedes gastar  $k_i$  en reparar la  $i$ -ésima tabla para incrementar su capacidad en 1

### Output

Imprime una sola línea — La máxima cantidad de carne que puede conseguir Inki.

### Scoring

El puntaje estará dividido en los siguientes subtasks:

#### Subtask 1 (30 puntos):

- $1 \leq n \leq 1000$
- $0 \leq m \leq 10^5$

#### Subtask 2 (70 puntos):

- $1 \leq n \leq 10^5$
- $0 \leq m \leq 10^{12}$

### Examples

standard input	standard output
5 12 1 5 10 6 4 3 2 2 1 1	4
3 12 1 1 1 1 2 3	3

## Note

Para el primer caso de prueba reparamos la primera tabla 3 veces haciendo que su nueva capacidad sea 4.

Para el segundo caso de prueba reparamos cada tabla 2 veces haciendo que la capacidad de todas las tablas sea de 3.

## Problem G. Suma de dados

Author: TISparta  
Input file: `standard input`  
Output file: `standard output`  
Time limit: 3 seconds  
Memory limit: 256 megabytes

TISparta esta jugando un juego con dados. Un dado es un cubo, sus caras contienen distintos enteros desde el 1 hasta el 6 como puntos negros. En este juego él escogerá  $n$  dados y un entero positivo  $target$ , luego el lanzará los  $n$  dados y gana el juego si la suma de los números obtenidos es  $target$ .

Además, TISparta considera que dos jugadas son la misma si al ordenar los números obtenidos dan la misma secuencia. Por ejemplo, si en un juego con  $n = 3, target = 6$  él obtiene estos valores ( $dado_1 = 3, dado_2 = 2, dado_3 = 1$ ), él considera que esa jugada es la misma a haber obtenido los valores ( $dado_1 = 1, dado_2 = 2, dado_3 = 3$ ) con los dados (debido a que al ordenar ambos resultados nos da la secuencia  $\{1, 2, 3\}$ ).

Ahora, él se está preguntando de cuántas maneras puede ganar. ¿Puedes ayudarlo?

### Input

La única línea de entrada contiene dos enteros  $n, m$  ( $1 \leq n \leq 50, 1 \leq m \leq 6 \cdot n$ ) — El número de dados y la suma deseada.

### Output

Imprime una sola línea — El número de maneras en las que TISparta puede ganar.

### Scoring

El puntaje estará dividido en los siguientes subtasks:

#### Subtask 1 (30 puntos):

- $1 \leq n \leq 10$

#### Subtask 2 (70 puntos):

- $1 \leq n \leq 50$

Para todos los subtasks, se cumple:

- $1 \leq m \leq 6 \cdot n$

### Examples

standard input	standard output
1 6	1
2 4	2

### Note

Para el primer caso de prueba, el único resultado posible es obtener un 1.

Para el segundo caso de prueba, los únicos resultados posibles son obtener  $\{1, 3\}$  y  $\{2, 2\}$ .

## Problem H. Secuencia de divisores

Author: TISparta  
Input file: `standard input`  
Output file: `standard output`  
Time limit: 3 seconds  
Memory limit: 1024 megabytes

Leonidas te está dando esta secuencia:

$$f_0 = 0$$
$$f_n = f_{n-1} + \sigma_0(n)$$

Donde  $\sigma_0(n)$  = número de divisores de  $n$ .

Ahora, él quiere evaluar tu aprendizaje durante este entrenamiento de verano. Así, él te está dando  $Q$  consultas, en cada consulta recibes dos enteros positivos  $l$ , y por cada consulta debes imprimir cuando números primos hay en  $f_l, f_{l+1}, \dots, f_r$ .

### Input

La primera línea de entrada contiene un entero  $Q$  ( $1 \leq Q \leq 10^5$ ) — El número de consultas.

Las siguientes  $Q$  líneas de la entrada contienen dos enteros  $l$  y  $r$  ( $1 \leq l \leq r \leq 10^6$ ) — La  $i$ -ésima consulta.

### Output

Imprime  $Q$  enteros — el  $i$ -ésimo entero es la respuesta de la  $i$ -ésima consulta.

### Scoring

El puntaje estará dividido en los siguientes subtasks:

#### Subtask 1 (10 puntos):

- $1 \leq Q \leq 10^3$
- $1 \leq l \leq r \leq 10^3$

#### Subtask 2 (20 puntos):

- $1 \leq Q \leq 10^4$
- $1 \leq l \leq r \leq 10^5$

#### Subtask 3 (30 puntos):

- $1 \leq Q \leq 10^5$
- $1 \leq l \leq r \leq 10^5$

#### Subtask 4 (40 puntos):

- $1 \leq Q \leq 10^5$
- $1 \leq l \leq r \leq 10^6$

## Examples

standard input	standard output
3 1 1 3 9 49 68	0 2 4
1 1 1000	132

## Note

Los primeros  $f_i$  son:

$$f_1 = 1$$

$$f_2 = 3$$

$$f_3 = 5$$

$$f_4 = 8$$

$$f_5 = 10$$

$$f_6 = 14$$

$$f_7 = 16$$

$$f_8 = 20$$

$$f_9 = 23$$

$$f_{10} = 27$$