
Examen práctica PRO2 Q1-2019-2020–Problema 1**X68868_es**

1. El peso de este ejercicio en la nota del exámen de la práctica es de un 60%
2. La nota manual es un 1/3 de la nota de este problema (3,33... puntos/10), y la nota automática son 2/3 de la nota (6,66... puntos/10).
3. El peso de los juegos de pruebas público y privados en el cálculo de la nota automática es idéntico para todos ellos (cada juego de pruebas superado 2.5/10 puntos de la nota automática de este problema).

Queremos escribir un programa que procesa instrucciones de unión y de intersección de conjuntos. Con cada instrucción se reciben dos subconjuntos finitos de enteros y se habrá de escribir el subconjunto resultante de hacer la unión o la intersección de los dos subconjuntos dados.

En vuestro programa usaréis las siguientes definiciones:

```
// un subconjunto finito de enteros se representa mediante una
// set de int en orden estrictamente creciente
typedef set<int> subcjt;

// Pre: a y b son sets representando dos conjuntos de enteros
//      a = {a0, a1, ... }, b = {b0, ... }
// Post: c = set que representa a A U B, esto es, la unión de los
//      conjuntos representados por a y b
void unir(const subcjt& a, const subcjt& b, subcjt& c);

// Pre: a y b son sets representando dos conjuntos de enteros
//      a = {a0, a1, ... }, b = {b0, ... }
// Post: c = set que representa a A ^ B, esto es, la intersección
//      de los conjuntos representados por a y b
void intersectar(const subcjt& a, const subcjt& b, subcjt& c);
```

Se os suministra un módulo (ficheros `subset-IO.o` y `subset-IO.hh`) con, entre otras cosas, operaciones de entrada/salida para subconjuntos de enteros. En particular, proporciona un operador `>>` para leer un conjunto de números (que está escrito siguiendo el formato del apartado de **Entrada** y pondrá los elementos en orden si es necesario), y un operador `<<` para imprimir un conjunto de números (según el formato de **Salida**).

Con todo ello completaréis un pequeño programa que lee una tras otra diferentes instrucciones y los conjuntos a operar, ejecuta cada instrucción e imprime los conjuntos resultantes. Utilizad el fichero `plantilla_solucion.cc.txt` para escribir vuestra solución. Vuestra solución debe hallarse en el fichero `solution.cc`, que es el único fichero a enviar (una buena idea es renombrar el fichero `plantilla_solucion.cc.txt` como `solution.cc`).

Entrada

La entrada consiste en una secuencia de instrucciones. Cada instrucción comienza con una letra (\cup , \cap , o \mathcal{Q}), y si la letra es \cup o \cap a continuación se dan dos subconjuntos de enteros. Los conjuntos (operandos) se introducen delimitando sus contenidos mediante llaves de apertura y cierre '{' '}'. Debe haber al menos un blanco entre la llave de apertura y el primer elemento, y de igual modo debe existir al menos un blanco entre el último elemento y la llave de cierre. Las letras \cup e \cap denotan uniones e intersecciones, respectivamente. Si la instrucción comienza con la letra \mathcal{Q} se indica el final de la entrada (Quit).

Salida

Para cada instrucción (excepto \mathcal{Q}) se escribe el conjunto de números resultante de hacer la unión (\cup) o la intersección (\cap) de los dos conjuntos dados en la instrucción. Los conjuntos se imprimen utilizando '{' y '}' para delimitar su comienzo y su final, y se usan blancos para separar los elementos del conjunto, que han de escribirse en orden estrictamente creciente. Después de imprimir la salida de cada instrucción se imprime un salto de línea.

Ejemplo de entrada 1

```
 $\cup$  { 1 8 } { 9 2 -3 }  
 $\cap$  { 1 8 -3 12 } { 9 1 -3 }  
 $\cup$  { } { 2 3 }  
 $\cap$  { 1 5 7 9 } { 4 -2 8 6 }  
 $\mathcal{Q}$ 
```

Ejemplo de salida 1

```
{ -3 1 2 8 9 }  
{ -3 1 }  
{ 2 3 }  
{ }
```

Información del problema

Autoría: Profesores de PRO2

Generación: 2026-01-25T17:45:13.051Z

© *Jutge.org*, 2006–2026.
<https://jutge.org>