

Include Hazard

เวลาการทำงาน 1 วินาที, หน่วยความจำที่ใช้ได้ 16MB

เวลาเขียนโปรแกรมภาษา C เราสามารถเอาเนื้อหาของไฟล์หนึ่งๆ เข้ามาใส่ในไฟล์อีกไฟล์หนึ่งได้ด้วยการใช้คำสั่ง `#include` ยกตัวอย่างเช่น ถ้า `main.c` และ `lib.h` มีเนื้อหาดังต่อไปนี้

main.c	lib.h
<code>#include "lib.h"</code> <code>int main() { return f(); }</code>	<code>int f() { return 0; }</code>

เวลาคอมไพล์ คอมไพเลอร์ภาษา C จะเขียนเนื้อหาของ `main.c` ใหม่ โดยเอาเนื้อหาของ `lib.h` ไปแทรกไว้ที่คำสั่ง `#include "lib.h"` ใน `main.c` ดังนั้นเนื้อหาใหม่ของ `main.c` คือ

main.c (หลังถูกคอมไพเลอร์ประมวลผล)
<code>int f() { return 0; }</code> <code>int main() { return f(); }</code>

อนึ่งในไฟล์ที่ถูกไฟล์อื่น `include` อาจมีคำสั่ง `#include` อยู่ก็ได้ และในไฟล์นี้ อาจมีการ `include` ไฟล์อื่นๆ มากกว่าหนึ่งไฟล์ได้ ยกตัวอย่างเช่น

main.c	lib1.h
<code>#include "lib1.h"</code> <code>#include "lib3.h"</code> <code>int main() { return f1()+f3(); }</code>	<code>#include "lib2.h"</code> <code>int f1() { return f2(); }</code>
lib2.h	lib3.h
<code>int f2() { return 2; }</code>	<code>int f3() { return 3; }</code>

จะเห็นว่าเวลาคอมไพเลอร์ภาษา C จะเขียนเนื้อหาของ `main.c` ใหม่ ดังนี้

main.c (หลังถูกคอมไพเลอร์ประมวลผล)
<code>int f2() { return 2; }</code> <code>int f1() { return f2(); }</code> <code>int f3() { return 3; }</code> <code>int main() { return f1()+f3(); }</code>

อย่างไรก็ดีหากผู้เขียนโปรแกรมไม่ระมัดระวังก็อาจทำให้เกิดปัญหาได้สองประการคือ

1. ไฟล์เดียวกันถูก `include` มากกว่าหนึ่งครั้ง เช่น

main.c	lib1.h
<code>#include "lib1.h"</code> <code>#include "lib3.h"</code> <code>int main() { return f1()+f3(); }</code>	<code>#include "lib2.h"</code> <code>int f1() { return f2(); }</code>
lib2.h	lib3.h
<code>int f2() { return 2; }</code>	<code>#include "lib2.h"</code> <code>int f3() { return 3+f2(); }</code>

ในกรณีเมื่อคอมไพเลอร์ทำการคอมไพล์ main.c ไฟล์ lib2.c จะถูก include สองครั้ง ครั้งหนึ่งจากไฟล์ lib1.h และอีกครั้งจาก lib3.h ซึ่งทำให้ฟังก์ชัน f2() ถูกนิยามสองครั้ง ซึ่งอาจทำให้คอมไพล์ไม่ผ่านได้

2. ไฟล์ include กันเป็นวงกลม เช่น

main.c	lib1.h
#include "lib1.h"	#include "lib2.h"
int main() { return f1()+f3(); }	int f1() { return f2(); }
lib2.h	lib3.h
#include "lib3.h"	#include "lib1.h"
int f2() { return 2; }	int f3() { return 3+f2(); }

สังเกตว่าเมื่อคอมไพเลอร์ภาษา C คอมไพล์ไฟล์ main.c แล้วไฟล์ lib1.h จะ include ไฟล์ lib2.h ซึ่ง include ไฟล์ lib3.h ซึ่ง include ไฟล์ lib1.h และสามารถวนไปเช่นนี้เรื่อยๆ โดยไม่จำกัด

งานของคุณ

กำหนดไฟล์โปรแกรมภาษา C มาให้ N ไฟล์ แต่ละไฟล์จะถูกระบุด้วยตัวเลขตั้งแต่ 1 ถึง N พร้อมทั้งข้อมูลว่าไฟล์ใด include ไฟล์ได้บ้าง จงเขียนโปรแกรมเพื่อตอบคำถามว่าสำหรับไฟล์ทุกๆ ไฟล์ เมื่อคอมไพเลอร์ภาษา C ทำการคอมไพล์ไฟล์นั้นแล้วจะเกิดปัญหาใด ปัญหาหนึ่งในสองปัญหาที่กล่าวถึงข้างบนหรือไม่

ข้อมูลนำเข้า (อ่านจาก standard input)

บรรทัดแรกมีจำนวนเต็ม N ($1 \leq N \leq 1,000$)

บรรทัดที่ $i+1$ (สำหรับทุก ๆ $1 \leq i \leq N$) บอกว่าไฟล์ i ทำการ include ไฟล์ใดบ้าง ซึ่งบรรทัดที่ $i+1$ จะระบุในรูปแบบดังนี้

$$k \ a_1 \ a_2 \ \dots \ a_k$$

โดยที่ k และทุก ๆ a_j ที่ $1 \leq j \leq k$ เป็นจำนวนเต็มที่ไม่เป็นลบและมีค่าไม่เกิน N บรรทัดดังกล่าวมีความหมายว่าไฟล์ i ทำการ include ไฟล์ a_1, a_2, \dots , และ a_k เรารับประกันว่าเลข a_j จะมีค่าไม่ซ้ำกัน (ในไฟล์หนึ่งๆ จะไม่มีการ include ไฟล์เดียวกันซ้ำสองครั้ง) และในบรรทัดที่ $i+1$ จะไม่มี a_j ใดๆ ที่มีค่าเท่ากับ i (ไฟล์แต่ละไฟล์จะไม่ include ตัวเอง) นอกจากนี้รับประกันว่าจำนวนการ include ทั้งหมดจะไม่เกิน 3,000 ครั้ง

ข้อมูลส่งออก (แสดงออกทาง standard output)

มี N บรรทัด แต่ละบรรทัดมีข้อความ YES หรือ NO

ถ้าเมื่อคอมไพเลอร์ภาษา C คอมไพล์ไฟล์ i แล้วเกิดปัญหาใดปัญหาหนึ่งในปัญหาสองข้อข้างต้น ให้พิมพ์ YES ในบรรทัดที่ i หากไม่เกิดปัญหาใดขึ้นเลยให้พิมพ์ NO

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
2 1 2 0	NO NO
ข้อมูลนำเข้า	ข้อมูลส่งออก
3 1 2 1 3 1 1	YES YES YES
ข้อมูลนำเข้า	ข้อมูลส่งออก
4 2 2 3 1 4 1 4 0	YES NO NO NO

ข้อจำกัดเรื่องเวลาและหน่วยความจำ

โปรแกรมจะต้องทำงานเสร็จสิ้นในเวลา 1 วินาทีและใช้หน่วยความจำไม่เกิน 16 MB

การให้คะแนน

ในการให้คะแนนจะมีข้อมูลชุดทดสอบหลายชุด คะแนนสำหรับแต่ละชุดทดสอบจะเป็น 100% ถ้าคำตอบถูกต้อง และเป็น 0% ถ้าคำตอบผิด