1. Student Registration

There is a group of *N* friends who wish to enroll in a course together. The course has a maximum capacity of *M* students that can register for it. If there are *K* other students who have already enrolled in the course, determine if it will still be possible for all the *N* friends to do so or not.

Input Format:

The first line contains a single integer *T* - the number of test cases. Then the test cases follow.

Each test case consists of a single line containing three integers *N* , *M* and *K* - the size of the friend group, the capacity of the course and the number of students already registered for the course.

Output Format:
For each test case, output Yes if it will be possible for all the *N* friends to register for the course. Otherwise output No.
You may print each character of Yes and No in uppercase or lowercase (for example, yes, yEs, YES will be considered identical).

Constraints:

$1 \leq T \leq 1000$

$1 \leq N \leq M \leq 100$

$0 \leq K \leq M$

Sample I/O:

Input 1:
2
2 50 27
5 40 38

Output 1:
YES
NO

Input 2:
2
4 20 15
10 30 5

Output 2:
NO
YES

2. PDA

A positive number can be called as

- **Perfect Number**, if **Sum of it's proper factors (divisors) is EQUAL to that number**
- **Deficient Number**, if **Sum of it's proper factors (divisors) is LESS THAN that number**

- **Abundant Number**, if **Sum of it's proper factors (divisors) is GREATER THAN that number**

Note:
Proper factors (divisors) of a number N are the factors of N excluding N.
To give an idea, for number 6 factors are 1, 2, 3 and 6 But proper factors are only 1, 2 and 3.

Examples for Perfect, Deficient and Abundant Numbers respectively:

- 28 is a **Perfect Number** as the sum of it's proper factors (1+2+4+7+14=28) is EQUAL to 28.
- 5 is a **Deficient Number** as the sum of it's proper factors (1) is LESS THAN 5.
- 12 is an **Abundant Number** as the sum of it's proper factors (1+2+3+4+6=15) is GREATER THAN 12.

Given a number N, determine if it's **Perfect**, **Deficient** or **Abudant**.

Input Format:
Asingle line contains an integer N.

Output Format:
Print the output according to the Discription.

Sample I/O:

Input 1:
28
Output 1:
PERFECT

Input 2:
24
Output 2:
ABUNDANT

Input 3:
21
Output 3:
DEFICIENT