Question:

[IPv4](https://en.wikipedia.org/wiki/IPv4) was the first publicly used Internet Protocol. It used *4*-byte addresses and permitted *232* distinct values. The typical format for an *IPv4* address is *A.B.C.D* where *A*, *B*, *C*, and *D*are integers in the inclusive range between *0* and *255*.

[IPv6](https://en.wikipedia.org/wiki/IPv6), with *128* bits, was developed to permit the expansion of the address space. These addresses are represented by eight colon-separated sixteen-bit groups, where each sixteen-bit group is written using *1* to *4* hexadecimal digits. Leading zeroes in a section are often omitted from an address, meaning that the groups *0* is identical to *0000* and group *5* is identical to *0005*. Some examples of valid *IPv6* addresses are 2001:0db8:0000:0000:0000:ff00:0042:8329 and 3:0db8:0:01:F:ff0:0042:8329.

Given *n* strings of text that *may or may not be* valid Internet Protocol (IP) addresses, we want to determine whether each string of text is:

* An *IPv4* address.
* An *IPv6* address.
* *Neither* an *IPv6* address nor an *IPv4* address.

Complete the *checkIPs* function in the editor below. It has one parameter: an array of strings, *ip\_array*, where each element *i* denotes a string of text to be checked. It must return an array of strings where each element *i* contains the answer for *ipi*; each answer must be whichever of the following case-sensitive terms is appropriate:

* IPv4 if the string is a valid *IPv4* address.
* IPv6 if the string is a valid *IPv6* address.
* Neither if the string is not a valid *IPv4* or *IPv6* address.

**Input Format**

Locked stub code in the editor reads the following input from stdin and passes it to the function:

The first line contains an integer, *n*, denoting the number of elements in *ip\_array*.

Each line *i* of the *n* subsequent lines (where *0 ≤ i < n*) contains a string describing *ipi*.

**Constraints**

* *1 ≤ n ≤ 50*
* *1 ≤ ipi ≤ 500*
* It is guaranteed that any string containing a valid *IPv4* or *IPv6* address has no leading or trailing whitespace.

**Output Format**

The function must return an array of strings where each element *i* contains the string IPv4, IPv6, or Neither, denoting that *ipi* was an *IPv4* address, an *IPv6* address, or *Neither*(i.e., not an address at all). This is printed to stdout by locked stub code in the editor.

**Sample Input 0**

2

This line has junk text.

121.18.19.20

**Sample Output 0**

Neither

IPv4

**Explanation 0**

We must check the following *n = 2* strings:

1. *ip0 = "This line has junk text."* is not a valid *IPv4* or *IPv6* address, so we return Neither in index *0* of our return array.
2. *ip1 = "121.18.19.20"* is a valid *IPv4* address, so we return IPv4 in index *1* of our return array.

**Sample Input 1**

1

2001:0db8:0000:0000:0000:ff00:0042:8329

**Sample Output 1**

IPv6

**Explanation 1**

We only have *n = 1* value to check. Because *ip0 = "2001:0db8:0000:0000:0000:ff00:0042:8329"* is a valid *IPv6* address, we return IPv6 in index *0* of our return array.