B - Express as a power b

Given a number, check if the number can be expressed as pow(a, b) where both a and b should be greater than 1.

**Input Format**

First line of input contains T - number of test cases. Its followed by T lines, each line contains a single integer N.

**Constraints**

30 points  
1 <= T <= 104  
1 <= N <= 106

70 points  
1 <= T <= 106  
1 <= N <= 108

**Output Format**

For each test case, print "Yes" if N can be expressed as pow(a, b), print "No" otherwise, separated by newline.

**Sample Input 0**

5

2

16

31

8880

961

**Sample Output 0**

No

Yes

No

No

Yes

**Explanation 0**

Self Explanatory

Logic:

Given N, can it be expressed as AB

where A and B both should be > 1

1 <= T <= 106  
1 <= N <= 108

# Working Out A Solution First (without worrying about complexity)

How can we check if N can be written as ABWhat if we can compare N to every possible number of the form A power B.

* Since max value of N can be 108, **a cannot exceed 10000**, as (10000)2 will yield 108 when b=2.
* The value of b cannot exceed 27.
  + 220 ~= 210 \* 210 ~= 1024 \* 1024 ~= 1000 \* 1000 ~= 103 \* 103 ~= 106
  + To make it 108, we need to add another 102 == 100 ~= 128 == 27.
  + So value of b cannot exceed 27.

bool isAPowerB(int n) {

for (auto a = 2; a <= 10000; a++) {

auto b = 2;

while (true) {

auto p = *pow*(a, b);

if (p == n)

return true;

if (p > n)

break;

b++;

}

}

return false;

}

# Optimizing Solution

* How about generating all the numbers power(a, b) and storing them in set once.

#include <iostream>

#include <set>

#include <cmath>

#include <algorithm>

using namespace *std*;

constexpr int N = 1E8;

int main(void) {

*std*::*set*<long long int> values;

for (auto a = 2; a <= 100000; a++) {

auto b = 2;

while (true) {

long long int p = *pow*(a, b);

if (p > N || p < 0)

break;

values.*insert*(p);

b++;

}

}

*ios*::*sync\_with\_stdio*(0);

*cin*.*tie*(0);

*cout*.*tie*(0);

int t; *cin* >> t;

while (t--) {

long long int n; *cin* >> n;

if(values.*find*(n)!=values.*end*())

*cout* << "Yes\n";

else

*cout* << "No\n";

}

return 0;

}