

```
1 // https://toph.co/p/is-it-a-square
2 #include<bits/stdc++.h>
3 using namespace std;
4 #define mx 100005
5 #define mp make_pair
6 #define pb push_back
7 const int BLOCK_SIZE = 320;
8 bool flag[mx], ans[mx];
9 int zero, neg, XOR;
10 int a[mx];
11 vector< pair<int,int> >factor[mx];
12 vector< pair< int, pair<int,int> > > qr[BLOCK_SIZE];
13 void sieve(){
14     for(int i=2; i<=100000; i++){
15         if(!flag[i]){
16             for(int j=i; j<=100000; j+=i){
17                 int k=0, v=j;
18                 while(v%i==0){
19                     k++;
20                     v/=i;
21                 }
22                 factor[j].pb(mp(i,k));
23                 flag[j]=true;
24             }
25         }
26     }
27 }
28
29 void Add(int x){
30     int v = a[x];
31
32     if(v==0)zero++;
33     else if(v<0)neg++;
34
35     if(v<0)v*=(-1);
36     if(v==0 || v==1) return;
37
38     for(int i=0; i<factor[v].size(); i++){
39         int p = factor[v][i].first;
40         int k = factor[v][i].second;
41         if(k&1)XOR^=p;
42     }
43 }
44
45 void Remove(int x)
46 {
47     int v = a[x];
48
49     if(v==0)zero--;
50     else if(v<0)neg--;
51
52     if(v<0)v*=(-1);
53     if(v==0 || v==1) return;
54
55     for(int i=0; i<factor[v].size(); i++){
56         int p = factor[v][i].first;
57         int k = factor[v][i].second;
58         if(k&1)XOR^=p;
59     }
60 }
```

```
61
62 int main()
63 {
64     sieve();
65
66     int n,q; scanf("%d%d",&n,&q);
67     for(int i=1; i<=n; i++) scanf("%d",&a[i]);
68
69     int block = sqrt(n);
70
71     for(int i=0; i<q; i++){
72         int x,y; scanf("%d%d",&x,&y);
73         int block = x/BLOCK_SIZE;
74         qr[block].pb(mp(y,mp(x,i)));
75     }
76
77     for(int i=0; i<BLOCK_SIZE; i++){
78         sort(qr[i].begin(),qr[i].end());
79     }
80
81     zero=neg=XOR=0;
82     int x=1,y=0;
83
84     for(int k=0; k<BLOCK_SIZE; k++){
85         for(int i=0; i<qr[k].size(); i++){
86             int left = (qr[k][i].second).first;
87             int right = qr[k][i].first;
88             int idx = (qr[k][i].second).second;
89
90             while(x<left) Remove(x++);
91             while(x>left) Add(--x);
92             while(y<right) Add(++y);
93             while(y>right) Remove(y--);
94
95             if(zero>0)ans[idx]=true;
96             else if(neg&1)ans[idx]=false;
97             else if(XOR==0)ans[idx]=true;
98             else ans[idx]=false;
99         }
100    }
101
102    for(int i=0; i<q; i++){
103        if(ans[i])printf("Yes\n");
104        else printf("No\n");
105    }
106
107    return 0;
108 }
```

Is It A Square?

Limits: 1.5s, 256 MB

Shikamaru is the most brilliant student of his class. He is very good at Mathematics. Mr. Asuma is their math teacher. Today he taught the class about square numbers. A square number is a **non-negative** integer which is the product of some integer with itself. For example, **0, 1, 4, 9, 16, 25** are square numbers, but **2, 3, 5, 6** are not. Mr. Asuma wanted to test whether his students can identify square numbers or not. So he gave them a homework.

He gave them an array **A[]** consisting of **n** integers and asked **q** questions. In each question, he gave them two integers **l** and **r** which denotes two positions of the array. The students have to tell whether the product of all the integers between the two positions (**l** and **r** inclusive) is a square number or not. In other words, they have to compute the value of **P** represented as the following

$$P = \prod_{i=l}^r A_i$$

and check whether it is a square or not. The array follows 1-based indexing.

After returning home, Shikamaru started doing his homework. He was able to answer some of the questions correctly, but later some of the products **P** became so huge and he could not identify whether they are square or not. Can you help Shikamaru doing his homework?

Input

The first line contains two integers **n** ($1 \leq n \leq 10^5$) and **q** ($1 \leq q \leq 10^5$).
The second line contains **n** integers **A₁, A₂, ..., A_n** ($-10^5 \leq A_i \leq 10^5$).
Each of the next **q** lines contains two integers **l** and **r** ($1 \leq l \leq r \leq n$).

Output

For each question print “Yes” if the product is a square, otherwise print “No” in a separate line.

Samples

Input	Output
6 4	Yes
4 1 2 0 -3 -3	No
1 1	Yes
2 3	Yes
4 4	
5 6	

Question 1 : Product = **4**. It is a square number.

Question 2 : Product = **1 x 2 = 2**. It is not a square number.

Question 3 : Product = **0**. It is a square number.

Question 4 : Product = **(-3) x (-3) = 9**. It is a square number.