PROBLEM STATEMENT

Sherlock is stuck. He has an array ***A****1,****A****2, … ,****A****N*. He wants to know if there exists a subset, ***B****= {****A****i1,****A****i2, … ,****A****ik* where *1 ≤****i1****<****i2****< … <****ik****≤****N****,* of this array which follows the property

* *B* is non-empty.
* All elements of ***B*** are relatively prime, i.e. there exists no integer ***x****(****x****> 1)* that evenly divides all elements of ***B***.  
  Note that ***x*** may or may not be an element of ***A***.

**Input**  
The first line contains a single integer ***T***, the number of test cases. Each test case consists of two lines: a single integer ***N*** on the first line and ***N*** integers denoting the array ***A*** on the second.

**Output**  
Print the string YES if there exists any such subset, and or NO, if not.

**Constraints**  
*1 ≤****T****≤ 10  
1 ≤****N****≤ 100  
1 ≤****Ai****≤ 100,000  1≤****i****≤ N*

**Sample input**

2

3

1 2 3

2

2 4

**Sample output**

YES

NO

**Explanation**  
In first test case, *S={1}, S={1,2}, S={1,3}, S={2,3}*and *S={1,2,3}* are subsets where no integer greater than 1 divides the elements.  
In second test case, no non-empty subset exists which satisfies the given condition.

#include <iostream>

using namespace std;

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

int T,N;

cin>>T;

for (int i=0;i<T;i++)

{

cin>>N;

int A[N];

for(int j=0;j<N;j++)

{

cin>>A[j];

}

int count =0;

int min =A[0];

for(int k=1;k<N;k++)

{

if(A[k]<min)

min=A[k];

}

for(int k=0;k<N-1;k++)

{

if(min==1)

{

cout<<"YES"<<endl;

count++;

break;

}

int n =2;

int flag =0;

while(n<=min)

{

if(A[k]%n==0 && A[k+1]%n==0)

{

flag=1;

break;

}

else

{

n++;

}

}

if(flag==1)

{

continue;

}

else{cout<<"YES"<<endl;

count++;

break;}}

if(count==0){cout<<"NO"<<endl;}

} return 0; }