**The Psychic Type**

Attempted by: **3002**

/

Accuracy: **94%**

/

Maximum Score: **20**

/

48 Votes

Tag(s):

Ad-Hoc, Basic Programming, Easy, Implementation

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

Alakazam has the unique ability to teleport itself during fights. He has realized that he can use this ability not just in Poke'mon fights, but also during real time manipulation of arrays. Abra, his non-evolution form has an interesting array of *N* integers, A1, A2, ... , AN-1, AN.

Since, Alakazam wants to try his teleporation power on arrays too, he decides to formalize it: he can move from any position *i* to a position A[i] in any array of the world. Now, he is currently standing at a position he calls *S*. He wants you to figure out if he can reach the position *E*.

**Input format:**  
The first line consists of a single integer *N*. The second line consists of *N* integers A1, A2, ... , AN-1, AN, which are separated by a space. The next line consists of two integers *S* and *E*.

**Output format:**   
Print **Yes** if he can reach the position *E* starting from position *S*, else print **No**.

**Constraints:**  
1 ≤ *N* ≤ 1000  
1 ≤ *A*i ≤ N  
1 ≤ *S*, *E* ≤ N

**Reference:**  


**SAMPLE INPUT**

5

3 4 2 5 5

1 4

**SAMPLE OUTPUT**

Yes

**Explanation**

From 1, he can go to 3 since A[1]=3.  
From 3, he can go to 2 since A[3]=2.  
From 2, he can go to 4 since A[2]=4.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Swift-4.1, Visual Basic

#include <iostream>

using namespace std;

int main()

{

int n,i,s,e,flag=1;

cin>>n;

int a[n+1];

for(i=1;i<=n;i++)

{

cin>>a[i];

}

cin>>s>>e;

i=s;

while(n--)

{

if(a[i]==e||s==e)

{

flag=0;

cout<<"Yes"<<endl;

break;

}

else

i=a[i];

}

if(flag!=0)

cout<<"No";

return 0;

}