### Input:

Key ban đầu, key cần tạo

Số lượng key để ghép (N <= 10^3)

Danh sách key để ghép (a[i] <= 10^5)

### Output:

Số lần ghép tối thiểu

### Sample testcase:

3 30

3

2 5 7

3 => 3\*2 (6) => 6\*5 (30)

output: 2

3 => 6, 15, 21

6 => 12, 30. 42

### Code

function main() {

input(Sour, Dest)

input(n)

input(a[1], a[2], a[3], .. , a[n])

// Kiểm tra đỉnh được thăm hay chưa

visited[](0)

// Độ dài đường đi

dist[](-1)

// Danh sách các đỉnh chờ được duyệt

queue

queue.add(Sour)

visited[Sour] = 1

dist[Sour] = 0

// BFS

while(queue not empty) {

var u = queue.top()

queue.pop()

for(i: 1 -> n) {

var v = u\*a[i] % 100000

// if(visited[v] == 0)

// Cải tiến

if(dist[v] == -1) {

queue.add(v);

visited[v] = 1;

dist[v] = dist[u] + 1

if(v == Dest) break;

}

}

}

output(dist[Dest])

}

### Độ phức tạp: O(|E| + |V|) = O(10^5\*N + 10^5) = O(10^5 \* N)