```c
/* Given an already sorted array of positive integers, design an algorithm and implement it using a
program to find whether a given key element is present in the sorted array or not. For an array
arr[n], search at the indexes arr[0], arr[2], arr[4],.....,arr[2k] and so on. Once the interval (arr[2k]<
key < arr[ 2k+1] ) is found, perform a linear search operation from the index 2k
to find the element key. (Complexity < O(n), where n is the number of elements need to be scanned for
searching):
Jump Search
*/

#include <stdio.h>
#include <math.h>
int jumpsearch(int arr[], int n, int key)
{
    int step = sqrt(n);
    int prev = 0;
    int min = (step < n)? step : n;
    while (arr[min - 1] < key)
    {
        prev = step;
        step = step +sqrt(n);
        if (prev >= n)
            return -1;
    }
    for (int i = prev; i < min; i++)
    {
        if (arr[i] == key)
            return i;
    }
    return -1;
}
int main()
{
    int T;
    scanf("%d", &T);
    while (T--)
    {
        int n;
        printf("enter size of array: ");
        scanf("%d", &n);
        int arr[n];
        printf("enter elements in array: ");
        for (int i = 0; i < n; i++)
        {
            scanf("%d", &arr[i]);
        }
        int key;
        scanf("%d", &key);
        int result = jumpsearch(arr, n, key);
         if (result != -1)
         printf("Present ");
         else
         printf( "Not Present ");
    }
    return 0;
}
```