**Kth distance**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)[Amazon](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Amazon&isCmp=1)

Given an unsorted array that may contain duplicates. Also given a number k which is smaller than size of array. Write a function that returns true if array contains duplicates within k distance.

**Input:**

The first line of input contains an integer T denoting the number of test cases.  
The first line of each test case is K and N,K is the distance and N is the size of array.  
The second line of each test case contains N input A[].  
  
**Output:**

Print True if array contains duplicates within k distance else False  
  
**Constraints:**

1 ≤ T ≤ 50  
1 ≤ N ≤ 100  
1 ≤ K ≤ N  
1 ≤ A[] ≤ 1000  
  
**Example:**

**Input:**  
4  
3 8  
1 2 3 4 1 2 3 4  
3 6  
1 2 3 1 4 5  
3 5  
1 2 3 4 5  
3 5  
1 2 3 4 4

**Output:**  
False  
True  
False  
True

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=477>

#include <stdio.h>

#include <iostream>

#include <map>

#include <vector>

using namespace std;

int main() {

    int t;

    scanf("%d", &t);

    while(t--) {

       int k,n;

       scanf("%d %d", &k, &n);

       int A[n];

       for(int i =0; i < n; i++) {

          scanf("%d", &A[i]);

       }

       bool duplicado = false;

       for(int i =0; i < n - k; i++) { //recorro hasta n - k

           std::map<int, int> m;

           //recorro una distancia de k y almaceno en el mapa cuantos hay de cada elemento

           for(int j=i; j <= i+k; j++) {

              //printf("%d ", A[j]);

              m[A[j]]++;

           }

           for(std::map<int,int>::iterator it = m.begin(); it != m.end(); it++)  {

               //si hay mas de 1 de un determinado elemento

               // significa que dentro de una distancia k hay duplicados

               if(it->second > 1) {

                    duplicado = true;

                    break;

                }

            }

            if(duplicado) {

                printf("True");

                break;

            }

           //printf("\n");

        }

        if(!duplicado) {

           printf("False");

        }

        printf("**\n**");

    }

    return 0;

}