```
// LIBS
#include <stdio.h>
#include "header1.h" // custom header file

// FUNCTION DECLARATION
int minDistance(int *array, int N, int a, int b);

// MAIN FUNCTION
int main() {
  // Number of test cases
  int T;
  scanf("%d", &T);

  // Test cases begin
  for (int case_index = 0; case_index < T; case_index++) {
     // Length of array
     int N;
     scanf("%d", &N);

     // Initializing Array
     int *array;
     array = (int*) malloc(sizeof(int) * N);

     // Error Handlings
     if (array == NULL) {
     return 1;
     }

     // Inputting Array while checking for errors
     if (inputArray(array, N) != 0) {
     return -1;
     }

     int a, b;
     scanf("%d %d", &a, &b);

     int min_distance;

     min_distance = minDistance(array, N, a, b);

     if (min_distance <= 0) {
     return -1;
     }
```

```
    printf("%d\n", min_distance);

    free(array);
  }

  return 0;
}

// FUNCTION DEFINITION
int minDistance(int *array, int N, int a, int b) {
  int last_a = -1;
  int last_b = -1;
  int min_distance = N;

  for (int index = 0; index < N; index++) {
    if (array[index] == a) {
    if (last_b != -1) {
    min_distance = ((index - last_b) < min_distance) ? (index
- last_b) : min_distance;
    }
    last_a = index;
    }
    else if (array[index] == b) {
    if (last_a != -1) {
    min_distance = ((index - last_a) < min_distance) ? (index
- last_a) : min_distance;
    }
    last_b = index;
    }
  }

  return (min_distance == N) ? -1 : min_distance;
}
```