Fast and Last Index in C++

```
#include <iostream>
using namespace std;

void findFirstAndLastIndex(int arr[], int n,
int d) {
   int low = 0;
   int high = n - 1;
   int firstIndex = -1;
   int lastIndex = -1;

   // Finding the first occurrence
   while (low <= high) {
      int mid = low + (high - low) / 2;
      if (d > arr[mid]) {
         low = mid + 1;
      } else if (d < arr[mid]) {
         high = mid - 1;
      } else {
         firstIndex = mid;
         high = mid - 1;
      }
   }

   // Finding the last occurrence
   low = 0;
   high = n - 1;
   while (low <= high) {
      int mid = low + (high - low) / 2;
      if (d > arr[mid]) {
         low = mid + 1;
      } else if (d < arr[mid]) {
         high = mid - 1;
      } else {
         lastIndex = mid;
         low = mid + 1;
      }
   }

   cout << "First Index: " << firstIndex <<
endl;
   cout << "Last Index: " << lastIndex <<
endl;
}

int main() {
   int arr[] = {1, 5, 10, 15, 22, 33, 33, 33, 33,
33, 40, 42, 55, 66, 77, 33};
   int n = sizeof(arr) / sizeof(arr[0]);
   int d = 33;

   findFirstAndLastIndex(arr, n, d);

   return 0;
}
```

Dry Run Example (on sorted array):

Sorted version of the array:

{1, 5, 10, 15, 22, 33, 33, 33, 33, 33, 33, 40, 42, 55, 66, 77}

We want to find **first and last index of 33**.

First Occurrence:

Iteration	low	high	mid	arr[mid]	firstIndex	high (updated)
1	0	15	7	33	7	6
2	0	6	3	15		
3	4	6	5	33	5	4
4	4	4	4	22		

➡ First index = **5**

Last Occurrence:

Iteration	low	high	mid	arr[mid]	lastIndex	low (updated)
1	0	15	7	33	7	8
2	8	15	11	40		
3	8	10	9	33	9	10
4	10	10	10	33	10	11

➡ Last index = **10**

🖶 **Final Output:**

First Index: 5
Last Index: 10

First Index: 5 Last Index: 10	