***Solution***

***QNO 1***

#include <iostream>

using namespace std;

int binarySearch(int arr[], int n, int target) { int left = 0;

int right = n - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (arr[mid] == target) { return mid;

} else if (arr[mid] < target) { left = mid + 1;

} else {

right = mid - 1;

}

}

return -1;

}

int main() {

int arr[] = {1, 3, 5, 7, 9, 11, 13};

int n = sizeof(arr) / sizeof(arr[0]); int target = 7;

int index = binarySearch(arr, n, target);

if (index != -1) {

cout << "Target found at index: " << index << endl;

} else {

cout << "Target not found in the array." << endl;

}

return 0;

}

***SS:***



***QNO 2***

#include <iostream>

using namespace std;

int firstOccurrence(int arr[], int n, int target) { int left = 0;

int right = n - 1; int result = -1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (arr[mid] == target) { result = mid;

right = mid - 1;

} else if (arr[mid] < target) { left = mid + 1;

} else {

right = mid - 1;

}

}

return result;

}

int main() {

int arr[] = {1, 3, 5, 7, 7, 7, 9, 11, 13};

int n = sizeof(arr) / sizeof(arr[0]); int target = 7;

int index = firstOccurrence(arr, n, target);

if (index != -1) {

cout << "First occurrence of target at index: " << index << endl;

} else {

cout << "Target not found in the array." << endl;

}

return 0;

}

***SS:***



***QNO 3***

#include <iostream>

using namespace std;

int lastOccurrence(int arr[], int n, int target) {

int left = 0;

int right = n - 1; int result = -1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (arr[mid] == target) { result = mid;

left = mid + 1;

} else if (arr[mid] < target) { left = mid + 1;

} else {

right = mid - 1;

}

}

return result;

}

int main() {

int arr[] = {1, 3, 5, 7, 7, 7, 9, 11, 13};

int n = sizeof(arr) / sizeof(arr[0]); int target = 7;

int index = lastOccurrence(arr, n, target);

if (index != -1) {

cout << "Last occurrence of target at index: " << index << endl;

} else {

cout << "Target not found in the array." << endl;

}

return 0;

}

***SS:***



***QNO 4***

#include <iostream>

using namespace std;

int countOccurrences(int arr[], int n, int target) { int left = 0;

int right = n - 1;

int firstOccur = -1; int lastOccur = -1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (arr[mid] == target) { firstOccur = mid;

right = mid - 1;

} else if (arr[mid] < target) { left = mid + 1;

} else {

right = mid - 1;

}

}

left = 0;

right = n - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (arr[mid] == target) { lastOccur = mid;

left = mid + 1;

} else if (arr[mid] < target) { left = mid + 1;

} else {

right = mid - 1;

}

}

if (firstOccur != -1 && lastOccur != -1) return lastOccur - firstOccur + 1;

else

return 0;

}

int main() {

int arr[] = {1, 3, 5, 7, 7, 7, 9, 11, 13};

int n = sizeof(arr) / sizeof(arr[0]); int target = 7;

int occurrences = countOccurrences(arr, n, target);

if (occurrences != 0) {

cout << "Number of occurrences of target: " << occurrences << endl;

} else {

cout << "Target not found in the array." << endl;

}

return 0;

}

***SS:***

