

1. Linear Search Implementation

main.c	Output
<pre>1 #include <stdio.h> 2 3 int linearSearch(int arr[], int n, int target) { 4 for (int i = 0; i < n; i++) { 5 if (arr[i] == target) { 6 return i; // Target found at index i 7 } 8 } 9 return -1; // Target not found 10 } 11 12 int main() { 13 int arr[] = {4, 2, 9, 7, 5, 1}; 14 int target = 7; 15 int n = sizeof(arr) / sizeof(arr[0]); 16 17 int result = linearSearch(arr, n, target); 18 19 if (result != -1) { 20 printf("Target %d found at index %d\n", target, result); 21 } else { 22 printf("Target %d not found in the array\n", target); 23 } 24 25 return 0; 26 }</pre>	<pre>/tmp/WBPZxiGQtP.o Target 7 found at index 3 === Code Execution Successful ===</pre>

2. Binary Search Implementation

main.c	Output
<pre>1 #include <stdio.h> 2 3 int binarySearch(int arr[], int n, int target) { 4 int low = 0, high = n - 1; 5 while (low <= high) { 6 int mid = low + (high - low) / 2; 7 8 // Check if target is present at mid 9 if (arr[mid] == target) { 10 return mid; 11 } 12 13 // If target is greater, ignore left half 14 if (arr[mid] < target) { 15 low = mid + 1; 16 } 17 // If target is smaller, ignore right half 18 else { 19 high = mid - 1; 20 } 21 } 22 return -1; // Target not found 23 } 24 25 int main() { 26 int arr[] = {1, 2, 4, 5, 7, 9}; 27 int target = 5; 28 int n = sizeof(arr) / sizeof(arr[0]); 29 int result = binarySearch(arr, n, target); 30 if (result != -1) { 31 printf("Target %d found at index %d\n", target, result); 32 } else { 33 printf("Target %d not found in the array\n", target); 34 } 35 return 0; 36 }</pre>	<pre>/tmp/YDdnQYfJKg.o Target 5 found at index 3 === Code Execution Successful ===</pre>