Binary Search using Recursion:

Algorithm:

1. Divide the sorted array into two halves.
2. Compare the target element with the middle element.
3. If the target element is equal to the middle element, return the index.
4. If the target element is less than the middle element, recurse on the left half.
5. If the target element is greater than the middle element, recurse on the right half.

```c
#include <stdio.h>

// Function to perform binary search using recursion
int binarySearch(int arr[], int target, int low, int high)
{
    // Base case: If low exceeds high, target not found
    if (low > high) {
        return -1;
    }

    // Calculate mid index
    int mid = (low + high) / 2;

    // Target found at mid index
    if (arr[mid] == target) {
        return mid;
    }
    // Target less than mid element, search left half
    else if (arr[mid] > target) {
        return binarySearch(arr, target, low, mid - 1);
    }
    // Target greater than mid element, search right half
    else {
        return binarySearch(arr, target, mid + 1, high);
    }
}

int main() {
    int arr[] = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91};
    int target = 23;
    int n = sizeof(arr) / sizeof(arr[0]);

    int result = binarySearch(arr, target, 0, n - 1);

    if (result != -1) {
        printf("Target found at index %d\n", result);
    } else {
        printf("Target not found\n");
    }

    return 0;
}
```