Aim: write a program in c to perform binary search with recursive approach

Algorithm:

1. Take an array arr, left index l, right index r, and element to be searched x as input.
2. If r >= l, find the mid index mid = l + (r - l) / 2.
3. If arr[mid] is equal to x, return mid as element is found.
4. If arr[mid] > x, recursively call the function with l and mid - 1.
5. If arr[mid] < x, recursively call the function with mid + 1 and r.
6. Return -1 if element is not found.

Source Code:

#include <stdio.h>

*int* binary\_search\_recursive(*int* arr[], *int* l, *int* r, *int* x) {

    if (r >= l) {

*int* mid = l + (r - l) / 2;

        if (arr[mid] == x)

            return mid;

        if (arr[mid] > x)

            return binary\_search\_recursive(arr, l, mid - 1, x);

        return binary\_search\_recursive(arr, mid + 1, r, x);

    }

    return -1;

}

*int* main() {

*int* arr[] = {2, 3, 4, 10, 40};

*int* x = 10;

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

*int* result = binary\_search\_recursive(arr, 0, n - 1, x);

    if (result == -1)

        printf("Element is not present in array");

    else

        printf("Element is present at index %d", result);

    return 0;

}

Output:

Element is present at index 3