Write a C program to divide the integer array into two halves using function

recursion. Call the user-defined “divide” function recursively, with the left half

of the split array being passed as an argument for that function. Let the

recursive function to get executed until the array size becomes one. Count the

number of iterations to reach the base condition. Explain the working

procedure of recursive function with stack structure

#include <stdio.h>

void divide(int arr[], int start, int end, int\* count) {

if (start == end) {

return;

}

int mid = (start + end) / 2;

divide(arr, start, mid, count);

divide(arr, mid + 1, end, count);

(\*count)++;

}

int main() {

int arr[] = {1, 2, 3, 4, 5, 6, 7, 8};

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

int count = 0;

divide(arr, 0, n - 1, &count);

printf("Number of iterations: %d\n", count);

return 0;

}

