1. **Divide and conquer to find maximum and minimum value in list**

**Code:**

#include <stdio.h>

// Structure to store the result of max and min values

struct Pair {

int min;

int max;

};

// Function to find the maximum and minimum values in a list using Divide and Conquer

struct Pair findMaxMin(int arr[], int low, int high) {

struct Pair result, left, right;

int mid;

// If there is only one element

if (low == high) {

result.min = arr[low];

result.max = arr[low];

return result;

}

// If there are two elements

if (high == low + 1) {

if (arr[low] > arr[high]) {

result.max = arr[low];

result.min = arr[high];

} else {

result.max = arr[high];

result.min = arr[low];

}

return result;

}

// If there are more than two elements, divide the array into two halves

mid = (low + high) / 2;

left = findMaxMin(arr, low, mid);

right = findMaxMin(arr, mid + 1, high);

// Compare minimums of two halves

if (left.min < right.min)

result.min = left.min;

else

result.min = right.min;

// Compare maximums of two halves

if (left.max > right.max)

result.max = left.max;

else

result.max = right.max;

return result;

}

int main() {

int n;

printf("Enter the number of elements in the list: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the list:\n");

for (int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

}

struct Pair result = findMaxMin(arr, 0, n - 1);

printf("Minimum element: %d\n", result.min);

printf("Maximum element: %d\n", result.max);

return 0;

}

**Output:**

Enter the number of elements in the list:

6

Enter the elements of the list:

1

4

6

1

032

2

Minimum element: 1

Maximum element: 32

--------------------------------

Process exited after 21.62 seconds with return value 0

Press any key to continue . . .

