NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY
(An Institution of National Importance under MHRD, Govt. of India)
KARAIKAL – 609 609

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Roll Number: CS19B1009 **Name** ARUN KUMAR R

Semester: 2nd Semester **Class:** Computer Science and Engineering

Subject Code: CS106 **Subject Name:** Data Structures Laboratory

Exercise Number – 3

Date: 18.12.19

1.MERGE_SORT

AIM:

To sort a given array using merge sort algorithm.

ALGORITHM:

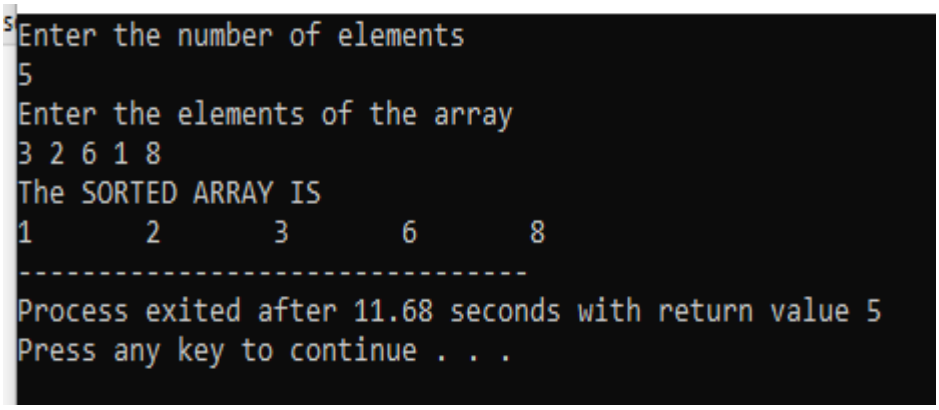
1. Start the program .
2. declare the variable.
3. using divide and conquer method , divide the given array into basic elements.
4. from lower to higher sort the array, i.e merge the array.
5. output the result.
6. end the program.

PROGRAM:

```
#include<stdio.h>
int merge(int a[], int low, int mid, int high)
{
        int b[100], i = low, j = mid+1, k = low;
        while(i <= mid && j <= high)
        {
                if(a[i] < a[j])
                {
                        b[k++] = a[i++];

                }
                else
                {
                        b[k++] = a[j++];

                }
        }
```

```
        while(i<=mid)
        b[k++] =a[i++];
        while(j<=high)
        b[k++] = a[j++];
                for(i = low; i <= high; i++)
                        {
                                a[i] = b[i];
                        }
}

void merge_sort(int a[], int low, int high)
{
        int mid;
        mid = (low+high)/2;
        if(low<high)
        {
                merge_sort(a, low, mid);
                merge_sort(a, mid+1, high);
                merge(a, low, mid, high);
        }
}

void main()
{
         int i, a[100], n;
    printf("Enter the number of elements\n");
    scanf("%d",&n);

    printf("Enter the elements of the array\n");
    for(i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }

    merge_sort(a,0,n-1);

    printf("The SORTED ARRAY IS\n");

    for(i = 0; i < n; i++)
    {
        printf("%d\t", a[i]);
    }
}
```

OUTPUT:



RESULT:

The program was executed successfully.

2.HEAP_SORT

DATE:18.12.19

AIM:

To sort a given array using heap sort algorithm.

ALGORITHM:

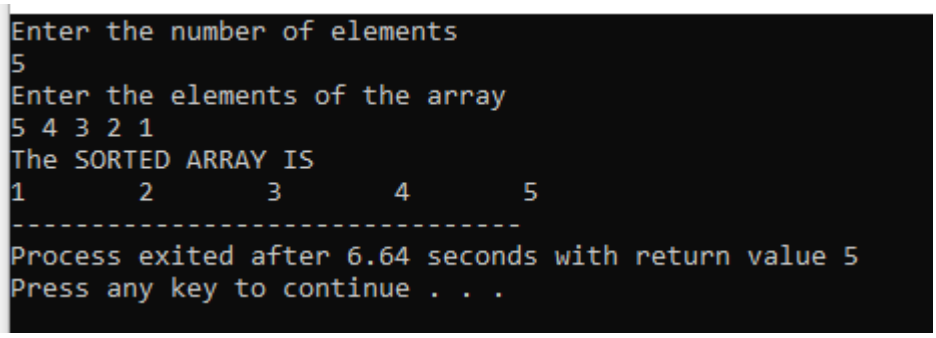
1. Start the program.
2. declare the variable.
3. construct a max heap using heapify algorithm.
4. construct a result array by getting the root elements of the heap.
5. output the result.
6. end the program.

PROGRAM:

```
#include<stdio.h>

void swap(int* a, int* b)
{
    int t = *a;
    *a = *b;
    *b = t;
}


void heapify(int arr[], int n, int i)
{
    int largest = i;
    int l = 2*i + 1;
    int r = 2*i + 2;

    if (l < n && arr[l] > arr[largest])
        largest = l;


    if (r < n && arr[r] > arr[largest])
        largest = r;


    if (largest != i)
    {
        swap(&arr[i], &arr[largest]);
```

```
        heapify(arr, n, largest);
    }
}


void heap_sort(int arr[], int n)
{
   int i;
    for ( i = n / 2 - 1; i >= 0; i--)
        heapify(arr, n, i);


    for ( i=n-1; i>=0; i--)
    {

        swap(&arr[0], &arr[i]);


        heapify(arr, i, 0);
    }
}




void main()
{
    int i, a[100], n;
    printf("Enter the number of elements\n");
    scanf("%d",&n);

    printf("Enter the elements of the array\n");
    for(i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }

   heap_sort(a, n);

    printf("The SORTED ARRAY IS\n");

    for(i = 0; i < n; i++)
    {
        printf("%d\t", a[i]);
    }
}
```

OUTPUT:



RESULT:

The program was executed successfully.