Name : Yash Liladhar Amarapurkar

Class : TY A Roll No:41

**EXPERIMENT NO: 1**

#include <stdio.h>

// conquer operation

void merge(int a[], int p, int q, int r)

{

int i, j, k;

int n1 = q - p + 1;

int n2 = r - q;

int LeftArray[n1], RightArray[n2];

for ( i = 0; i< n1; i++)

LeftArray[i] = a[p + i]; // left sub array created

for (j = 0; j < n2; j++)

RightArray[j] = a[q + 1 + j]; // right sub array created

i = 0;

j = 0;

// copying and comparing

k = p;

while (i< n1 && j < n2)

{

if(LeftArray[i] <= RightArray[j])

{

a[k] = LeftArray[i];

i++;

}

else

{

a[k] = RightArray[j];

j++;

}

k++;

}

while (i<n1)

{

a[k] = LeftArray[i];

i++;

k++;

}

while (j<n2)

{

a[k] = RightArray[j];

j++;

k++;

}

}

// merge operation

void mergeSort(int a[], int p, int r)

{

if (p < r)

{

int q = (p + r) / 2;

mergeSort(a, p, q);

mergeSort(a, q + 1, r);

merge(a, p, q, r);

}

}

// array printed

void printArray(int a[], int n)

{

int i;

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

printf("%d ", a[i]);

printf("\n");

}

// array creation

int main()

{

int a[] = { 12, 13, 27, 42, 10 };

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

printf("Before sorting array elements are - \n");

printArray(a, n);

mergeSort(a, 0, n - 1);

printf("After sorting array elements are - \n");

printArray(a, n);

return 0;

}

