****

**ABDUL MUTEEN MASOOD**

**SP20-BSE-085**

**SECTION:A**

***MERGE SORT***

#include <stdio.h>

void merge(int arr[], int ub, int mid, int lb);

void mergeSort(int arr[], int ub, int lb);

void printArray(int arr[], int size);

int main() {

int arr[10];

printf("Enter elements in Array:\n");

for(int i = 0; i < 10; i++){

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

}

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

printf("Array before sorting\n");

printArray(arr, size);

mergeSort(arr, 0, size - 1);

printf("\n");

printf("Array after sorting: \n");

printArray(arr, size);

}

void merge(int arr[], int ub, int mid, int lb) {

int arrCopySize1 = mid - ub + 1;

int arrCopySize2 = lb - mid;

int L[arrCopySize1], M[arrCopySize2];

for (int i = 0; i < arrCopySize1; i++){

L[i] = arr[ub + i];

}

for (int j = 0; j < arrCopySize2; j++){

M[j] = arr[mid + 1 + j];

}

int i, j, k;

i = 0;

j = 0;

k = ub;

while (i < arrCopySize1 && j < arrCopySize2) {

if (L[i] <= M[j]) {

arr[k] = L[i];

i++;

} else {

arr[k] = M[j];

j++;

}

k++;

}

while (i < arrCopySize1) {

arr[k] = L[i];

i++;

k++;

}

while (j < arrCopySize2) {

arr[k] = M[j];

j++;

k++;

}

}

void mergeSort(int arr[], int ub, int lb) {

if (ub < lb) {

int m = ub + (lb - ub) / 2;

mergeSort(arr, ub, m);

mergeSort(arr, m + 1, lb);

merge(arr, ub, m, lb);

}

}

void printArray(int arr[], int size) {

for (int i = 0; i < size; i++)

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

printf("\n");

}