**Experiment 5: (sorting)**

**Q1) Write a program to input elements into two arrays A[5] and B[5]. Input the elements in ascending order and then merge their values into a resultant array C[10] in sorted manner using UDF.**

**#include <stdio.h>**

**void merge(int [], int, int [], int, int []);**

**int main()**

**{**

**int a[5], b[5], m, n, c, sorted[10];**

**printf("Input number of elements in first array\n");**

**scanf("%d", &m);**

**printf("Input %d integers\n", m);**

**for (c = 0; c < m; c++)**

**{**

**scanf("%d", &a[c]);**

**}**

**printf("Input number of elements in second array\n");**

**scanf("%d", &n);**

**printf("Input %d integers\n", n);**

**for (c = 0; c < n; c++)**

**{**

**scanf("%d", &b[c]);**

**}**

**merge(a, m, b, n, sorted);**

**printf("Sorted array:\n");**

**for (c = 0; c < m + n; c++)  
 {**

**printf("%d\n", sorted[c]);**

**}**

**return 0;**

**}**

**void merge(int a[], int m, int b[], int n, int sorted[]) {**

**int i, j, k;**

**j = k = 0;**

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

**{**

**if (j < m && k < n)**

**{**

**if (a[j] < b[k])**

**{**

**sorted[i] = a[j];**

**j++;**

**}**

**else**

**{**

**sorted[i] = b[k];**

**k++;**

**}**

**i++;**

**}**

**else if (j == m)**

**{**

**for (; i < m + n;)**

**{**

**sorted[i] = b[k];**

**k++;**

**i++;**

**}**

**}**

**else**

**{**

**for (; i < m + n;)**

**{**

**sorted[i] = a[j];**

**j++;**

**i++;**

**}**

**}**

**}**

**}**

**Q2) Write a program to implement insertion sort on a given list of array elements.**

**// C program for insertion sort**

**#include <math.h>**

**#include <stdio.h>**

**/\* Function to sort an array using insertion sort\*/**

**void insertionSort(int arr[], int n)**

**{**

**int i, key, j;**

**for (i = 1; i < n; i++) {**

**key = arr[i];**

**j = i - 1;**

**/\* Move elements of arr[0..i-1], that are**

**greater than key, to one position ahead**

**of their current position \*/**

**while (j >= 0 && arr[j] > key) {**

**arr[j + 1] = arr[j];**

**j = j - 1;**

**}**

**arr[j + 1] = key;**

**}**

**}**

**// A utility function to print an array of size n**

**void printArray(int arr[], int n)**

**{**

**int i;**

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

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

**printf("\n");**

**}**

**int main()**

**{**

**int arr[] = { 12, 11, 13, 5, 6 };**

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

**insertionSort(arr, n);**

**printArray(arr, n);**

**return 0;**

**}**

**Q3) Write a C program to implement quick sort to a given list of integers to sort in ascending order.**

**#include<stdio.h>**

**void quicksort(int [10],int,int);**

**int main()**

**{**

**int x[20],size,i;**

**printf("Enter size of the array: ");**

**scanf("%d",&size);**

**printf("Enter %d elements: ",size);**

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

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

**quicksort(x,0,size-1);**

**printf("Sorted elements: ");**

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

**printf(" %d",x[i]);**

**return 0;**

**}**

**void quicksort(int x[10],int first,int last){**

**int pivot,j,temp,i;**

**if(first<last)**

**{**

**pivot=first;**

**i=first;**

**j=last;**

**while(i<j)**

**{**

**while(x[i]<=x[pivot]&&i<last)**

**i++;**

**while(x[j]>x[pivot])**

**j--;**

**if(i<j)**

**{**

**temp=x[i];**

**x[i]=x[j];**

**x[j]=temp;**

**}**

**}**

**temp=x[pivot];**

**x[pivot]=x[j];**

**x[j]=temp;**

**quicksort(x,first,j-1);**

**quicksort(x,j+1,last);**

**}**

}

**Output:**

**Enter size of the array: 5**

**Enter 5 elements: 3 8 0 1 2**

**Sorted elements: 0 1 2 3 8**

**Q4) write a program to implement bubble sort on a given list of array elements.**

**#include <stdio.h>**

**int main()  
{  
  int array[100], n, c, d, swap;**

**printf("Enter number of elements\n");  
  scanf("%d", &n);**

**printf("Enter %d integers\n", n);**

**for (c = 0; c < n; c++)  
    scanf("%d", &array[c]);**

**for (c = 0 ; c < n - 1; c++)  
  {  
    for (d = 0 ; d < n - c - 1; d++)  
    {  
      if (array[d] > array[d+1]) /\* For decreasing order use '<' instead of '>' \*/  
      {  
        swap       = array[d];  
        array[d]   = array[d+1];  
        array[d+1] = swap;  
      }  
    }  
  }**

**printf("Sorted list in ascending order:\n");**

**for (c = 0; c < n; c++)  
     printf("%d\n", array[c]);**

**return 0;  
}**