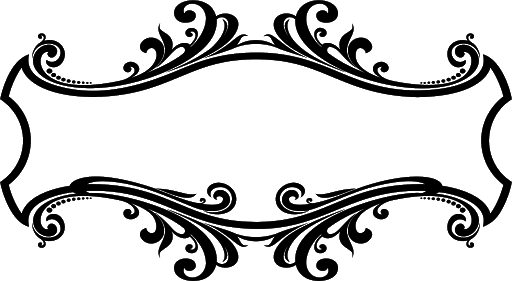


DELHI SKILL AND ENTERPRENEURSHIP UNIVERSITY

SESSION 2021-22

SUBJECT:-

Data Structures



PRACTICAL FILE



COURSE :- BCA-1A

ROLL NO :- 41221139

SUBMITTED BY :- SUBMITTED TO:-

Sachin Rajbhar Mrs. Divya Rana

|  |  |  |  |
| --- | --- | --- | --- |
| **INDEX** | | | |
| S.NO | T O P I C S | Date | Teachers  Sign. |
| 1 | To initialize 1D array at compile time. | 29/04/22 |  |
| 2 | To initialize 1D array at run time (using loops) | 06/05/22 |  |

Program 1

**Aim:** To initialize 1D array at compile time.

**Software used:** Vs Code

**Program:**

#include <iostream>

using namespace std;

#define N 5

int main(){

    int a[N]; //note N is defined as constant preprocessor i.e, 5

    cout<<"Array Declared successfully in compile time";

    return 0;

}

**Output:**



Program 2

**Aim:** To initialize 1D array at run time (using loops)

**Software used:** Vs Code

**Program:**

#include <iostream>

using namespace std;

int main() {

    int a[5];

    cout<<"Enter the 5 elements of array ";

    //read array at compile time by loops

    for(int i=0;i<5;i++)

    {

        cin>>a[i];

    }

    //printing array elements

    for(int i=0;i<5;i++)

    {

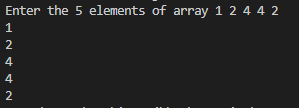
        cout<<a[i]<<"\n";

    }

    return 0;

}

**Output:**



**PRACTICAL-24**

**AIM:** Write a program for Merge Sort in array.

**PROGRAM:**

#include<stdio.h>

void mergesort(int arr[],int,int,int);

void partition(int arr[],int,int);

int main(){

    int arr[50];

    int i,size;

    printf("Enter size of array:");

    scanf("%d",&size);

    printf("Enter the elements\n");

    for(i=0;i<size;i++){

        scanf("%d",&arr[i]);

    }

    partition(arr,0,size-1);

    printf("After merge sort\n");

     for(i=0;i<size;i++){

        printf("%d ",arr[i]);

    }

return 0;

}

void partition(int arr[],int lb ,int ub)

{

int mid;

if(lb < ub){

     mid = (lb + ub)/2;

     partition(arr,lb,mid);

     partition(arr,mid+1,ub);

     mergesort(arr,lb,mid,ub);

}

}

void mergesort(int arr[],int lb,int mid ,int ub)

{

 int i,j,k;

 int b[50];

    i=lb;

    j=mid+1;

    k=lb;

 while ((i <= mid) && (j <= ub))

    {

        if (arr[i] <= arr[j])

        {

            b[k] = arr[i];

            i++;k++;

        }

        else

        {

            b[k] = arr[j];

            j++;

            k++;

        }

    }

    if (i > mid)

    {

          while(j<=ub){

        {

            b[k] = arr[j];

            j++;

            k++;

        }

    }

    }

    else

    {

        while(i<=mid){

             b[k] = arr[i];

             i++;

             k++;

    }

    }

    for (k = lb; k <= ub; k++)

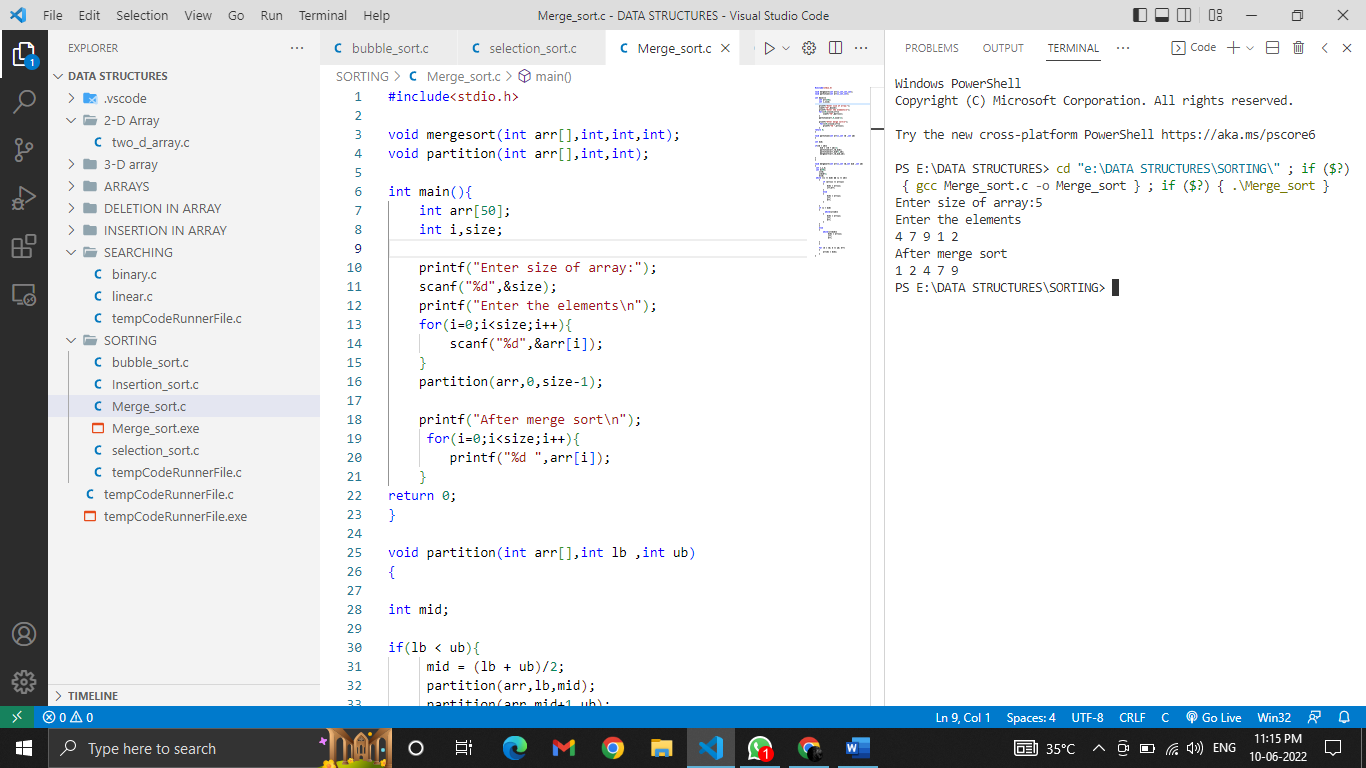
    {

        arr[k] = b[k];

    }

}

**OUTPUT:**



**PRACTICAL-25**

**AIM:** Write a program for Quick Sort in array.

**PROGRAM:**

#include <stdio.h>

void quicksort (int [], int, int);

int main()

{

    int arr[50];

    int size, i;

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

    scanf("%d", &size);

    printf("Enter the elements to be sorted:\n");

    for (i = 0; i < size; i++)

    {

        scanf("%d", &arr[i]);

    }

    quicksort(arr, 0, size - 1);

    printf("After applying quick sort\n");

    for (i = 0; i < size; i++)

    {

        printf("%d ", arr[i]);

    }

    printf("\n");

    return 0;

}

void quicksort(int arr[], int lb, int ub)

{

    int pivot, i, j, temp;

    if (lb < ub)

    {

        pivot = lb;

        i = lb;

        j = ub;

        while (i < j)

        {

            while (arr[i] <= arr[pivot] && i <= ub)

            {

                i++;

            }

            while (arr[j] > arr[pivot] && j >= lb)

            {

                j--;

            }

            if (i < j)

            {

                temp = arr[i];

                arr[i] = arr[j];

                arr[j] = temp;

            }

        }

        temp = arr[j];

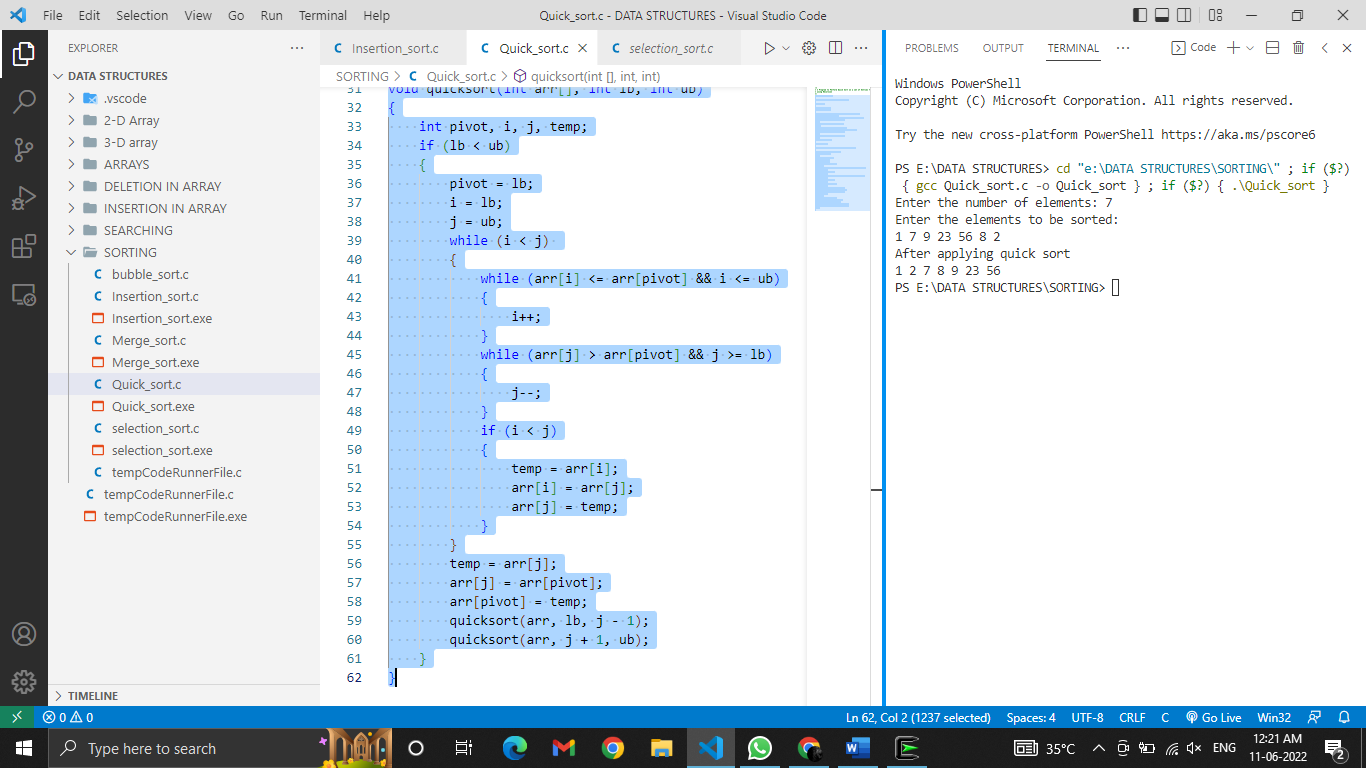
        arr[j] = arr[pivot];

        arr[pivot] = temp;

        quicksort(arr, lb, j - 1);

        quicksort(arr, j + 1, ub);

    }

}

**OUTPUT:**