**Data Structures using C**

**Lab-1**

**Program1: Write a program to search a key element with in the given array of elements using linear search process.**

**Solution:**

//File Name: LinearSearchDemo3.c

//==================================

#include<stdio.h>

void main() {

int a[20], i, n, key, flag = 0, pos;

printf("Enter value of n : ");

scanf("%d", &n);

// Write code to read array elements

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

printf("Enter element for a[%d] : ",i);

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

}

printf("Enter key element : ");

scanf("%d", &key);

// Write code for linear search process

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

if(a[i]==key) {

pos=i;

flag=1;

break;

} else {

flag=0;

}

}

if ( flag==1 ) { //Write the condition part

printf("The key element %d is found at the position %d\n",key, pos ); //Complete the statement

} else {

printf("The key element %d is not found in the array\n",key ); //Complete the statement

}

}

**Program2: Write a program to read and print the given integer elements of an array (with max size 10).**

**Solution:**

//File Name: ArraysDemo6.c

//==================================

#include<stdio.h>

int main() {

int arr[10], i=0, n;

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

scanf("%d",&n);

printf("Enter array elements : ");

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

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

}

printf("The given integer array elements : ");

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

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

}

}

**Program3: Write a C program to insert an element at a given position in an array using functions.**

**Solution:**

//File Name: InsertEleAtPosInArray.c

//==================================

#include <stdio.h>

#include "InsertEleAtPosInArray1.c"

void main() {

int arr[50], n, pos, c, value;

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

scanf("%d", &n);

printf("Enter %d elements : ", n);

read(arr, n);

printf("The given array : ");

display(arr, n);

printf("Enter the position with in %d : ", n);

scanf("%d", &pos);

if (pos <= n) {

printf("Enter the value to insert : ");

scanf("%d", &value);

insertAtPosition(arr, n, pos, value);

n++;

printf("The resultant array : ");

display(arr, n);

} else {

printf("Enter the correct available position\n");

}

}

//File Name: InsertEleAtPosInArray1.c

//==================================

#include<stdio.h>

void read(int arr[], int n) {

int i=0 ;

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

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

}

}

void display(int arr[], int n) {

int i = 0;

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

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

}

printf("\n");

}

void insertAtPosition(int arr[], int n, int pos, int value){

int i = 0;

pos=pos-1;

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

arr[i+1] = arr[i];

}

arr[pos] = value;

}

**Program4: Write a program to delete an element from the array.**

**Solution:.**

//File Name: deleteElement.c

//==================================

#include<stdio.h>

int main() {

int n, arr[20], i=0, k;

scanf("%d",&n);

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

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

}

scanf("%d",&k);

for(i=k ; i<n-1 ; i++) {

arr[i]=arr[i+1];

}

printf("Array after deletion is: ");

for(i=0 ; i<n-1 ; i++) {

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

}

}

Program 5: Write a program to search the given element from a list of elements with binary search technique using **recursion**.

Solution:

//File Name: Program912.c

//==================================

#include <stdio.h>

#include "Program912a.c"

void main() {

int a[20], n, key, flag;

printf("Enter value of n : ");

scanf("%d", &n);

read(a, n);

bubbleSort(a, n);

printf("After sorting the elements are : ");

display(a, n);

printf("Enter key element : ");

scanf("%d", &key);

flag = binarySearch(a, 0, n - 1, key);

if (flag == -1) {

printf("The given key element %d is not found\n", key);

} else {

printf("The given key element %d is found at position : %d\n", key, flag);

}

}

//File Name: Program912a.c

//==================================

#include<stdio.h>

void read(int arr[], int n) {

printf("Enter %d elements : ",n);

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

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

}

}

void bubbleSort(int a[], int n){

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

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

if(a[i]<a[j]){

int s=a[i];

a[i]=a[j];

a[j]=s;

}

}

}

}

void display(int array[], int n){

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

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

}

printf("\n");

}

int binarySearch(int a[], int y, int r, int x) {

if(r>=1) {

int mid = (r+y)/2;

if(a[mid]==x){

return mid;

} else if(a[mid]>x) {

return binarySearch(a, y, r-1, x);

} else {

return binarySearch(a, y+1, r, x);

}

}

else {

return -1;

}

}

**Program 6: Write a program to search a key element in the given array of elements using binary search.**

**Solution:**

//File Name: BinarySearchDemo3.c

//==================================

#include<stdio.h>

void main() {

int a[20], i, j, n, key, flag = 0, low, high, mid, temp;

printf("Enter value of n : ");

scanf("%d", &n);

// Write the code to read an array of elements

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

printf("Enter element for a[%d] : ",i);

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

}

printf("Enter key element : ");

scanf("%d", &key);

// Write the code to sort the elements using any sorting technique

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

j=i;

while(a[j]<a[j-1] && j>0) {

temp = a[j];

a[j]=a[j-1];

a[j-1]=temp;

j--;

}

}

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

// Write the code to display the elements

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

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

printf("\n");

}

low = 0; // Complete the statement

high = n-1; // Complete the statement

temp=-1;

// Write the code to search an element using binary search process

while(low<=high) {

int mid = low + (high-low)/2;

if(a[mid]==key) {

temp=mid;

flag=1;

break;

} else if(a[mid]<key) {

low=mid+1;

} else if(a[mid]>key) {

high=mid-1;

} else {

flag = 0;

}

}

if (temp>=0 ) { // Write the condition part

printf("The key element %d is found at the position %d\n",key, temp); // Complete the statement

} else {

printf("The Key element %d is not found in the array\n",key); // Complete the statement

}

}

**Program 7: Write a program to search the given element from a list of elements with linear search technique using recursion.**

**Solution:**

//File Name: Program911.c

//==================================

#include <stdio.h>

#include "Program911a.c"

void main() {

int a[20], n, pos, key;

printf("Enter n value : ");

scanf("%d", &n);

read(a, n);

printf("Enter a key element : ");

scanf("%d", &key);

pos = linearSearch(a, 0, n - 1, key);

if (pos == -1) {

printf("The key element %d is not found\n", key);

} else {

printf("The key element %d is found at position : %d\n", key, pos);

}

}

//File Name: Program911a.c

//==================================

#include<stdio.h>

int linearSearch(int a[], int s, int e, int key);

void read(int a[], int n) {

int i = 0;

printf("Enter %d elements : ",n);

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

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

}

}

int linearSearch(int a[], int s, int e, int key) {

if (s>e) {

return -1;

}

else if (a[s]==key) {

return s;

}

s++;

return linearSearch(a, s, e, key);

}

**Lab-2**

**Program 1: Write a C program to Implement a sparse matrix using arrays.**

**Solution:**

//File Name: sparsematrix\_arrays.c

//==================================

#include<stdio.h>

void main()

{

int i,j,row,column,a[50][50];

printf("Enter the row & column sizes of the sparse matrix : ");

scanf("%d%d",&row,&column);

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

{

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

{

printf("Enter the value of sparse\_matrix[%d][%d] : ",i,j);

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

}

}

printf("Original matrix scanned is:\n");

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

{

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

{

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

}

printf("\n");

}

printf("Sparse matrix array (Triplet) representation is:\n");

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

{

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

{

if(a[i][j]!=0)

{

printf("%d ",i);

}

}

}

printf("\n");

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

{

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

{

if(a[i][j]!=0)

{

printf("%d ",j);

}

}

}

printf("\n");

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

{

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

{

if(a[i][j]!=0)

{

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

}

}

}

printf("\n");

}

**Program 2: SUM OF NON-EDGES :**

**A set of numbers forming a matrix N\*N is passes as input. The program has to print the sum of numbers which are not along the edges.**

**Solution:**

//File Name: SumOfNonEdges.c

//==================================

#include<stdio.h>

void main()

{

int a[100][100],n,sum=0;

scanf("%d",&n);

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

{

for(int j=0;j<n;j++)

{

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

if(i!=0&&i!=n-1)

{

if(j!=0&&j!=n-1)

{

sum+=a[i][j];

}

}

}

}

printf("%d\n",sum);

}

**Program 3: Write a C program to Implement sparse matrix multiplication using arrays.**

**Solution:**

//File Name: sparsematrix\_multiplication.c

//==================================

#include<stdio.h>

void main()

{

int i,j,k,r1,r2,c1,c2,a1[10][10],a2[10][10],c[10][10];

printf("Enter no of rows : ");

scanf("%d",&r1);

printf("Enter no of coloumns : ");

scanf("%d",&c1);

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

{

scanf("%d",&a1[i][j]);

}

}

printf("Enter no of rows : ");

scanf("%d",&r2);

printf("Enter no of columns : ");

scanf("%d",&c2);

for(i=0;i<r2;i++)

{

for(j=0;j<c2;j++)

{

scanf("%d",&a2[i][j]);

}

}

if(c1==r2)

{

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

{

c[i][j]=0;

for(k=0;k<c1;k++)

{

c[i][j]+=a1[i][k]\*a2[k][j];

}

}

}

printf("Result of matrix multiplication : \n");

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

{

if(c[i][j]!=0)

{

printf("%d ",i);

}

}

}

printf("\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

{

if(c[i][j]!=0)

{

printf("%d ",j);

}

}

}

printf("\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

{

if(c[i][j]!=0)

{

printf("%d ",c[i][j]);

}

}

}

printf("\n");

}

else

{

printf("Not possible.");

}

}

**Program 4: Write a program to represent and display sparse matrix.**

**Solution:**

//File Name: sparseMatrix.c

//==================================

#include<stdio.h>

void main()

{

int r,c,i,j,a[20][20],count=0;

printf("Enter the size of matrix (rows, columns): ");

scanf("%d%d",&r,&c);

printf("Enter elements of matrix\n");

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

{

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

{

printf("[%d][%d]: ",i,j);

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

if(a[i][j]!=0)

count++;

}

}

printf("Sparse form - list of 3 triples\n");

printf("%d\t%d\t%d\n",r,c,count);

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

{

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

{

if(a[i][j]!=0)

printf("%d\t%d\t%d\n",i,j,a[i][j]);

}

}

}

**Program 5: Write a program to implement sparse matrix addition using arrays.**

**Solution:**

//File Name: addMatrix.c

//==================================

#include<stdio.h>

#include<stdlib.h>

#define max 20

void printsparse(int b[max][3]);

void readsparse(int b[max][3]);

void addsparse(int b1[max][3],int b2[max][3],int b3[max][3]);

void main()

{

int b1[max][3],b2[max][3],b3[max][3];

readsparse(b1);

readsparse(b2);

addsparse(b1,b2,b3);

printsparse(b3);

}

void readsparse(int b[max][3])

{

int t,m,n;

printf("Enter no. of rows and columns:\n");

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

printf("No. of non-zero triples:");

scanf("%d",&t);

b[0][0]=m;

b[0][1]=n;

b[0][2]=t;

for(int i=1;i<=t;i++)

{

printf("Enter the triples(row,column,value):");

scanf("%d%d%d",&b[i][0],&b[i][1],&b[i][2]);

}

}

void addsparse(int b1[max][3],int b2[max][3],int b3[max][3])

{

int i,j,t1,t2,k;

if(b1[0][0]!=b2[0][0] || b1[0][1]!=b2[0][1])

{

printf("\nYou have entered invalid matrix!! size must be equal");

exit(0);

}

t1=b1[0][2];

t2=b2[0][2];

i=j=k=0;

b3[0][0]=b1[0][0];

b3[0][1]=b1[0][1];

while(i<=t1 && j<=t2)

{

if(b1[i][0]<b2[j][0])

{

b3[k][0]=b1[i][0];

b3[k][1]=b1[i][1];

b3[k][2]=b1[i][2];

k++;

i++;

}

else if(b2[j][0]<b1[i][0])

{

b3[k][0]=b2[j][0];

b3[k][1]=b2[j][1];

b3[k][2]=b2[j][2];

k++;

j++;

}

else if(b1[i][1]<b2[j][1])

{

b3[k][0]=b1[i][0];

b3[k][1]=b1[i][1];

b3[k][2]=b1[i][2];

k++;

i++;

}

else if(b2[j][1]<b1[i][1])

{

b3[k][0]=b2[j][0];

b3[k][1]=b2[j][1];

b3[k][2]=b2[j][2];

k++;

j++;

}

else

{

b3[k][0]=b1[i][0];

b3[k][1]=b1[i][1];

b3[k][2]=b1[i][2]+b2[j][2];

k++;

i++;

j++;

}

}

while(i<=t1)

{

b3[k][0]=b1[i][0];

b3[k][1]=b1[i][1];

b3[k][2]=b1[i][2];

i++;

k++;

}

while(j<=t2)

{

b3[k][0]=b2[j][0];

b3[k][1]=b1[j][1];

b3[k][2]=b1[j][2];

j++;

k++;

}

b3[0][2]=k-1;

}

void printsparse(int b[max][3])

{

int i,t;

t=b[0][2];

printf("row\tcolumn\tvalue\n");

for(i=1;i<=t;i++)

{

printf("%d\t%d\t%d\n",b[i][0],b[i][1],b[i][2]);

}

}

**Program 6: Counter Spiral:**

**Given a square matrix, you have to write a program to print it in a counter-clockwise spiral form.**

**Solution:**

//File Name: CounterSpiral.c

//==================================

#include<stdio.h>

void main()

{

int n;

scanf("%d",&n);

int a[n][n];

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

{

for(int j=0;j<n;j++)

{

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

}

}

int top=0,bottom=n-1,left=0,right=n-1;

int dir=1;

int count=0;

while(top<=bottom&&left<=right)

{

if(dir==1)

{

for(int i=top;i<=bottom;i++)

{

count++;

if(count!=n\*n)

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

else

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

}

left++;

dir=2;

}

else if(dir==2)

{

for(int i=left;i<=right;i++)

{

count++;

if(count!=n\*n)

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

else

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

}

bottom--;

dir=3;

}

else if(dir==3)

{

for(int i=bottom;i>=top;i--)

{

count++;

if(count!=n\*n)

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

else

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

}

right--;

dir=4;

}

else if(dir==4)

{

for(int i=right;i>=left;i--)

{

count++;

if(count!=n\*n)

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

else

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

}

top++;

dir=1;

}

}

printf("\n");

}