Ques :1- Write a Program to search the value ?

Ans 1:-

#include <stdio.h>

#include<conio.h>

void main()

{

clrscr();

int a[50],search,i,n;

printf("Enter number of elements in array\n");

scanf("%d",&n);

printf("Enter %d integer(s)\n", n);

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

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

printf("Enter a number to search\n");

scanf("%d",&search);

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

{

if (a[i] == search)

{

printf("%d is present at location %d.\n", search, i+1);

break;

}

}

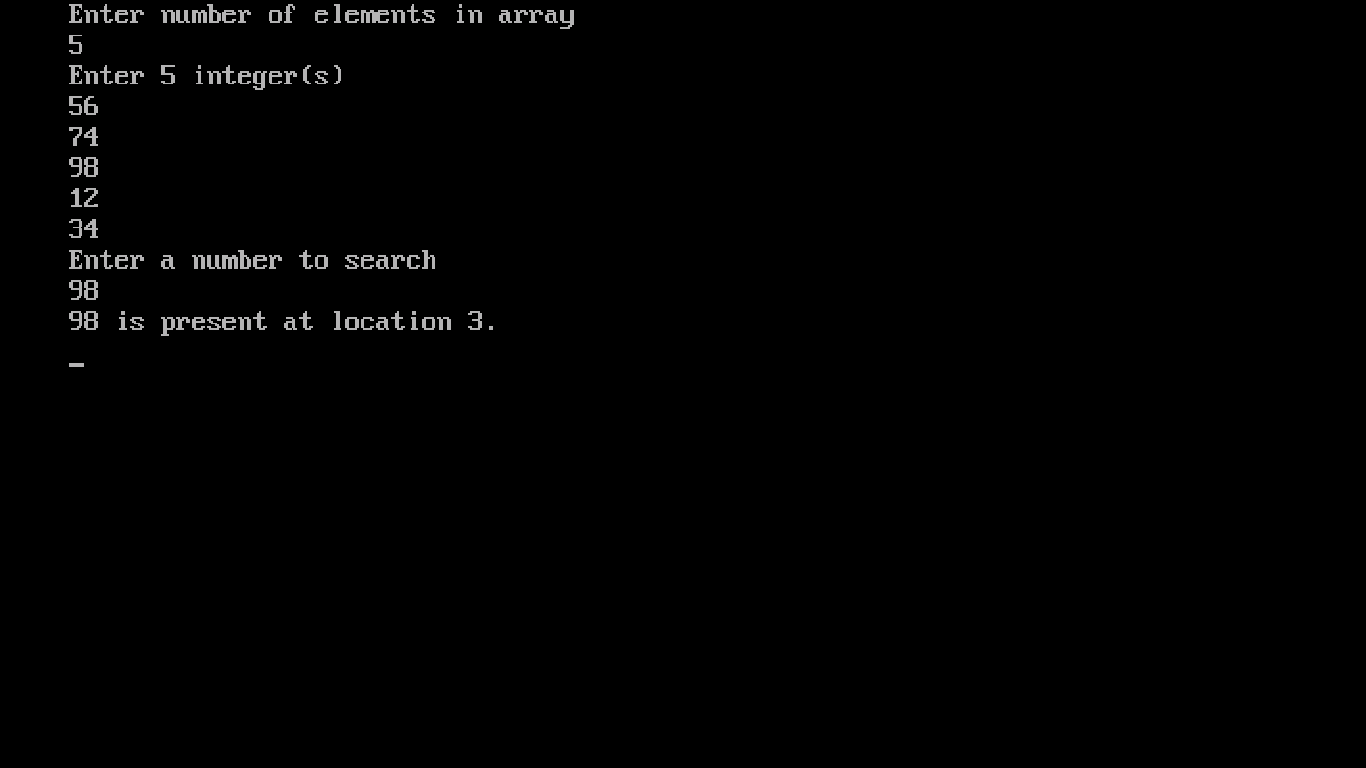
if (i == n)

printf("%d isn't present in the array.\n", search);

getch();

}

**OUTPUT IS:-**

****

Ques :2- Write a Program for BUBBLE SORT ?

Ans 2:-

#include <stdio.h>

#include <conio.h>

void main()

{

clrscr();

int array[100], n, i, j, swap;

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

scanf("%d", &n);

printf("Enter %d integers\n", n);

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

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

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

{

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

{

if (array[j]>array[j+1])

{

swap = array[j];

array[j] = array[j+1];

array[j+1] = swap;

}

}

}

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

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

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

getch();

}

**OUTPUT IS:-**



Ques :3- Write a Program for SELECTION SORT ?

Ans 3:-

#include <stdio.h>

#include <conio.h>

void main()

{

clrscr();

int array[100], n, i, j, p, t;

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

scanf("%d", &n);

printf("Enter %d integers\n", n);

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

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

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

{

p=i;

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

{

if (array[p]>array[j])

p=j;

}

if (p!=i)

{

t = array[i];

array[i] = array[p];

array[p] = t;

}

}

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

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

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

getch();

}

**OUTPUT IS:-**



Ques :4- Write a Program FOR INSERTION SORT ?

Ans 4:-

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

int n,array[100],i,j,k,temp=0;

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

scanf("%d", &n);

printf("Enter %d integers\n", n);

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

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

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

{

k = array[i];

for (j=i-1;j>=0;j--)

{

if (array[j]>k)

{

array[j+1]=array[j];

temp=1;

}

else

break;

}

if (temp)

array[j+1] = k;

}

printf("Sorted list is:\n");

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

{

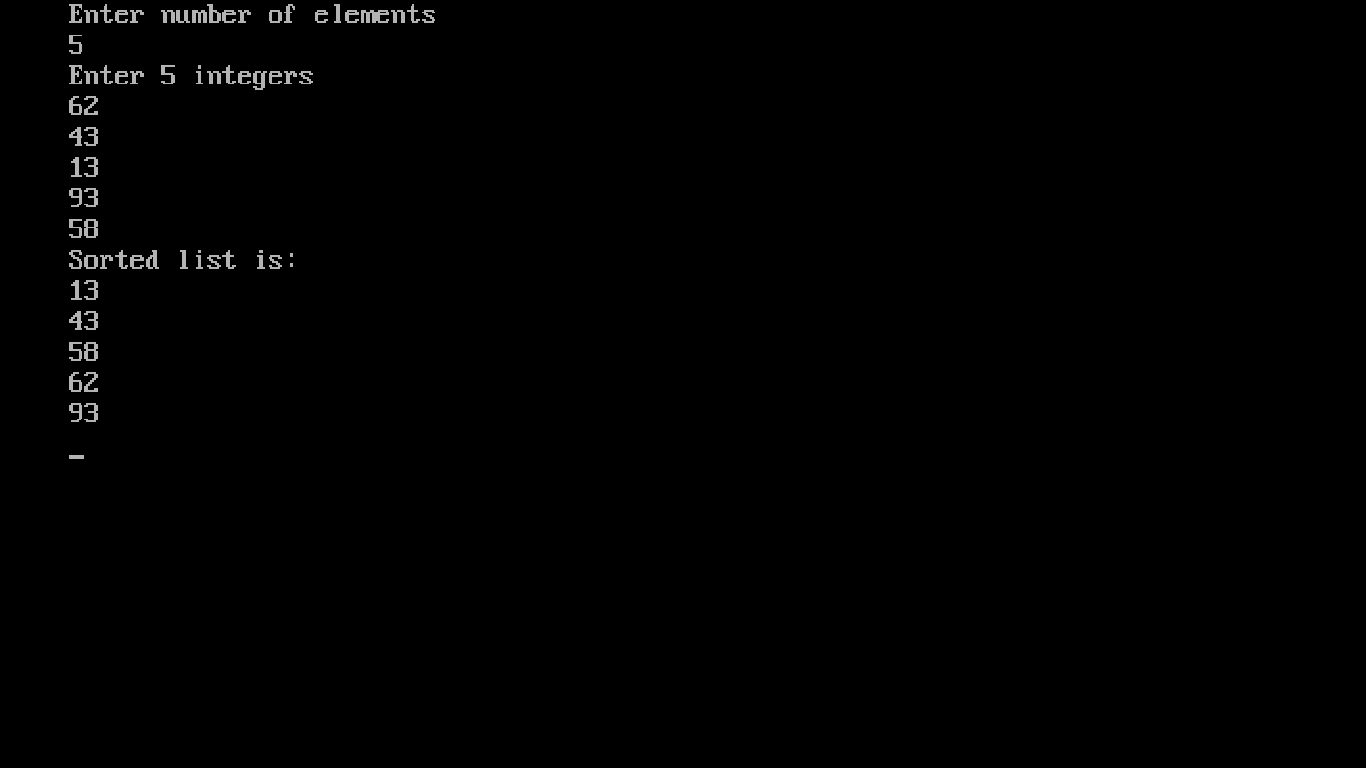
printf("%d\n", array[i]);

}

getch();

}

**OUTPUT IS:-**



Ques :5- Write a Program FOR BINARY SEARCH ?

Ans 5:-

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

int n,i, first, last, middle, search, array[1000];

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

scanf("%d",&n);

printf("Enter %d integers\n", n);

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

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

printf("Enter value to find\n");

scanf("%d",&search);

first = 0;

last = n - 1;

middle = (first+last)/2;

while (first <= last)

{

if (array[middle] < search)

first = middle + 1;

else if (array[middle] == search)

{

printf("%d found at location %d.\n", search, middle+1);

break;

}

else

last = middle - 1;

middle = (first + last)/2;

}

if (first > last)

printf("Not found! %d isn't present in the list.\n", search);

getch();

}

**OUTPUT IS:-**



Ques :6- Write a program for QUICK SORT ?

Ans 6:-

#include<stdio.h>

#include<conio.h>

void quicksort(int number[25],int first,int last)

{

int i, j, pivot, temp;

if(first<last)

{

pivot=first;

i=first;

j=last;

while(i<j)

{

while(number[i]<=number[pivot] && i<last)

i++;

while(number[j]>number[pivot])

j--;

if(i<j)

{

temp=number[i];

number[i]=number[j];

number[j]=temp;

}

}

temp=number[pivot];

number[pivot]=number[j];

number[j]=temp;

quicksort(number,first,j-1);

quicksort(number,j+1,last);

}

}

void main()

{

clrscr();

int i, count, number[25];

printf("How many elements are u going to enter?: ");

scanf("%d",&count);

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

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

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

quicksort(number,0,count-1);

printf("Order of Sorted elements: ");

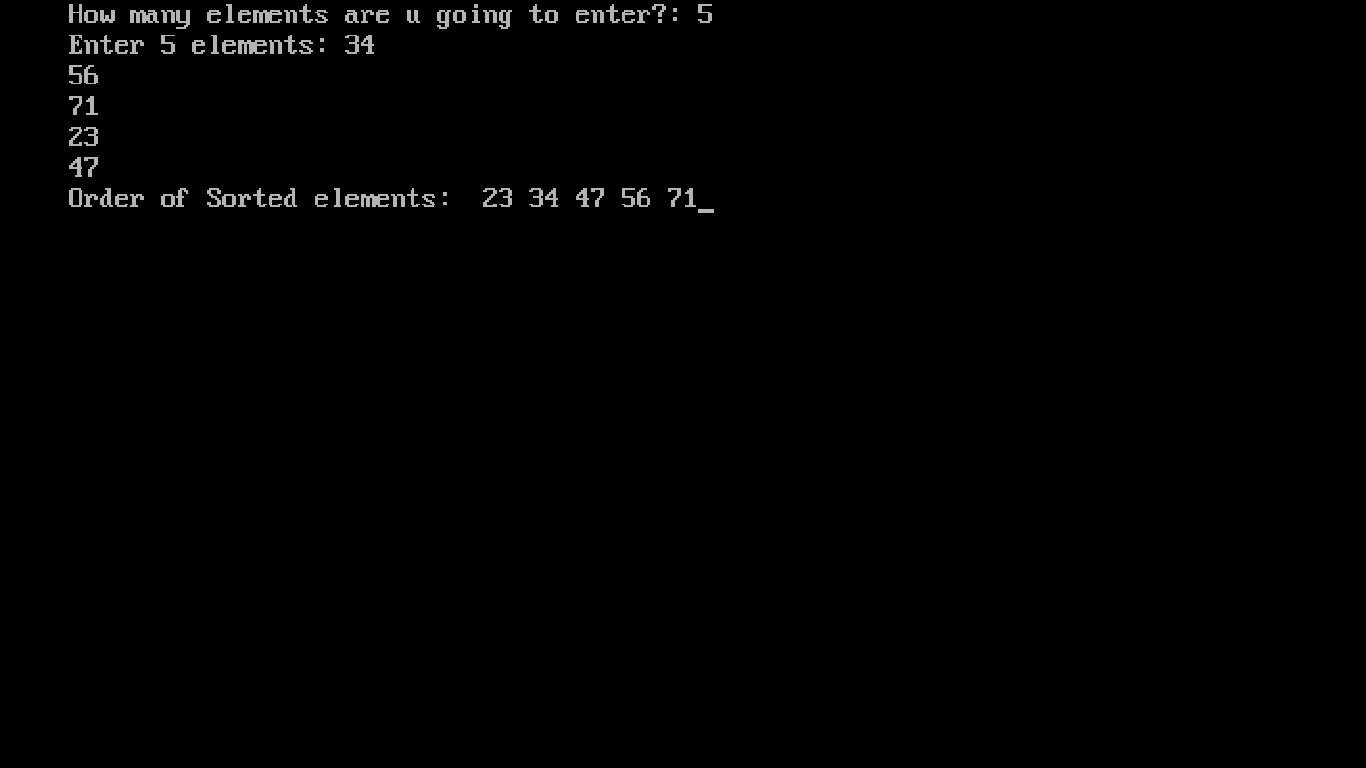
for(i=0;i<count;i++)

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

getch();

}

**OUTPUT IS:-**



Ques :7- Write a program for TOWER OF HANNOI ?

Ans 7:-

#include<stdio.h>

#include<conio.h>

void TOH(int, char, char, char);

int main ()

{

clrscr();

int n;

printf("Enter number of disks required: \n");

scanf ("%d", &n);

TOH (n, 'A', 'B', 'C');

getch();

return 0;

}

void TOH (int n, char src, char spare, char dest)

{

if (n==1)

printf("Move from %c to %c \n", src, dest);

else

{

TOH(n-1, src, dest, spare) ;

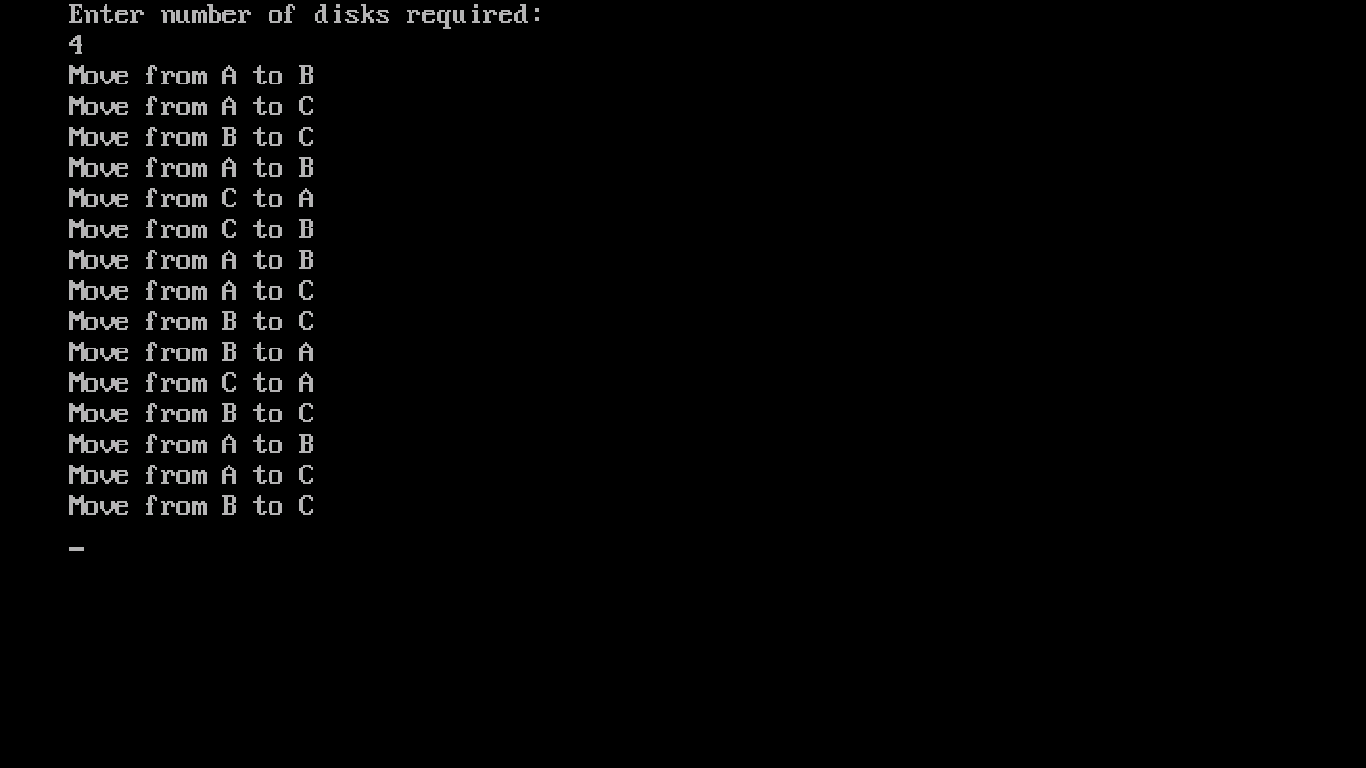
TOH(1, src, spare, dest);

TOH(n-1, spare, src, dest);

}

}

**OUTPUT IS:-**



Ques :8- Write a program for COUNTING SORT?

Ans 8:-

#include <stdio.h>

#include <conio.h>

void counting\_sort(int A[], int k, int n)

{

int i, j;

int B[15], C[100];

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

C[i] = 0;

for (j = 1; j <= n; j++)

C[A[j]] = C[A[j]] + 1;

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

C[i] = C[i] + C[i-1];

for (j = n; j >= 1; j--)

{

B[C[A[j]]] = A[j];

C[A[j]] = C[A[j]] - 1;

}

printf("The Sorted array is : ");

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

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

}

void main()

{

clrscr();

int n, k = 0, A[15], i;

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

scanf("%d", &n);

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

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

{

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

if (A[i] > k) {

k = A[i];

}

}

counting\_sort(A, k, n);

printf("\n");

getch();

}

**OUTPUT IS:-**



Ques :9- Write a program for HEAP SORT?

Ans 9:-

#include<stdio.h>

#include<conio.h>

int temp;

void heapify(int arr[], int size, int i)

{

int largest = i;

int left = 2\*i + 1;

int right = 2\*i + 2;

if (left < size && arr[left] >arr[largest])

largest = left;

if (right < size && arr[right] > arr[largest])

largest = right;

if (largest != i)

{

temp = arr[i];

arr[i]= arr[largest];

arr[largest] = temp;

heapify(arr, size, largest);

}

}

void heapSort(int arr[], int size)

{

int i;

for (i = size / 2 - 1; i >= 0; i--)

heapify(arr, size, i);

for (i=size-1; i>=0; i--)

{

temp = arr[0];

arr[0]= arr[i];

arr[i] = temp;

heapify(arr, i, 0);

}

}

void main()

{

clrscr();

int arr[] = {1, 10, 2, 3, 4, 1, 2, 100,23, 2};

int i;

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

heapSort(arr, size);

printf("The Heapify sorted elements\n");

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

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

}

**OUTPUT IS:-**



Ques :10- Write a program for RADIX SORT?

Ans 10:-

#include <stdio.h>

#include <conio.h>

int print(int \*a, int n)

{

int i;

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

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

}

void radix\_sort(int \*a, int n)

{

int i, b[10], m = 0, exp = 1;

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

{

if (a[i] > m)

m = a[i];

}

while (m / exp > 0)

{

int box[10] = { 0 };

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

box[a[i] / exp % 10]++;

for (i = 1; i < 10; i++)

box[i] += box[i - 1];

for (i = n - 1; i >= 0; i--)

b[--box[a[i] / exp % 10]] = a[i];

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

a[i] = b[i];

exp \*= 10;

}

}

void main()

{

int arr[10];

int i, num;

clrscr();

printf("Enter Number of Elements:- ");

scanf("%d", &num);

printf("Enter %d Integers:- ", num);

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

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

radix\_sort(&arr[0], num);

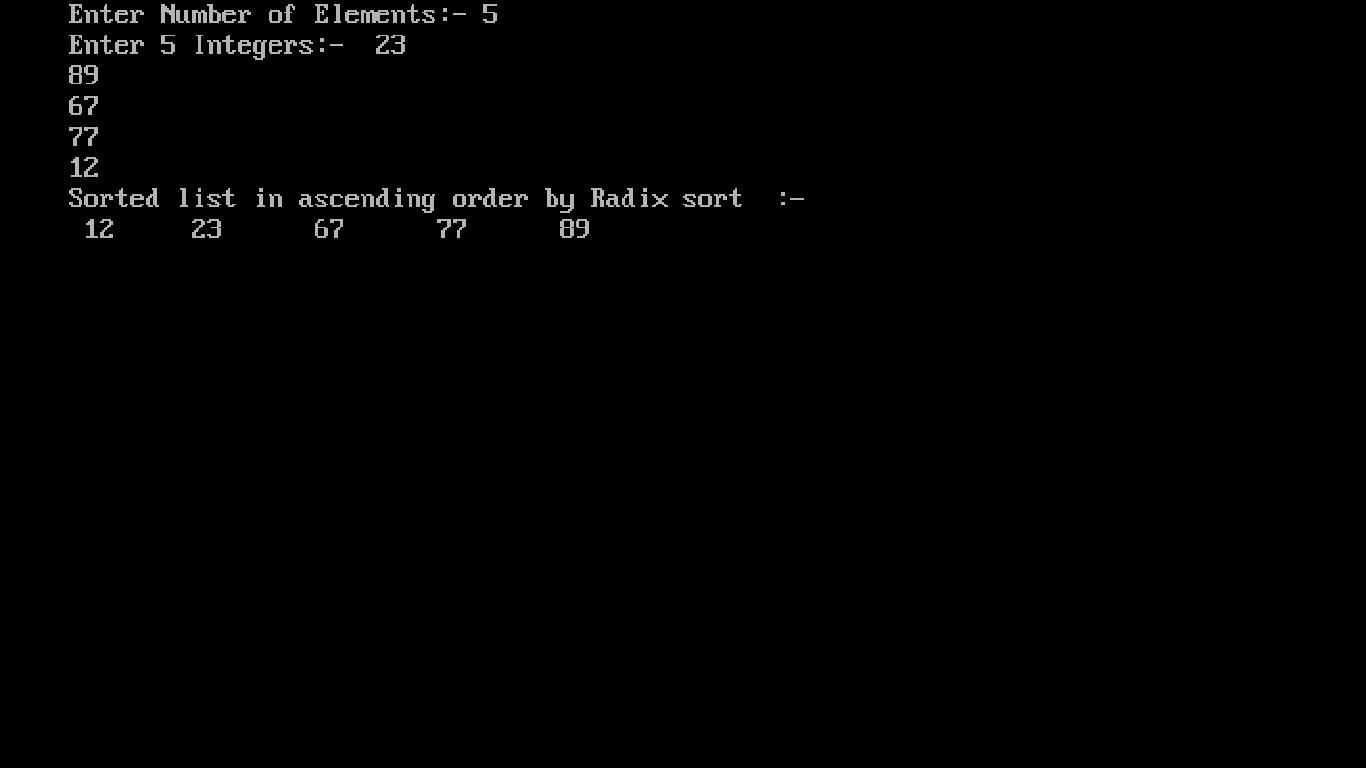
printf("Sorted list in ascending order by Radix sort :- \n ");

print(&arr[0], num);

getch();

}

**OUTPUT IS:-**



Ques :11- Write a program for SHELL SORT?

Ans 11:-

#include <stdio.h>

#include <conio.h>

void shellsort(int arr[], int num)

{

int i, j, k, tmp;

for (i = num / 2; i > 0; i = i / 2)

{

for (j = i; j < num; j++)

{

for(k = j - i; k >= 0; k = k - i)

{

if (arr[k+i] >= arr[k])

break;

else

{

tmp = arr[k];

arr[k] = arr[k+i];

arr[k+i] = tmp;

}

}

}

}

}

void main()

{

clrscr();

int arr[50];

int n, num;

printf(" \n Enter total no. of elements \n :- ");

scanf("%d", &num);

printf(" \n Enter %d numbers \n :- \n ", num);

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

{

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

}

shellsort(arr, num);

printf(" \n Sorted array is \n :- \n ");

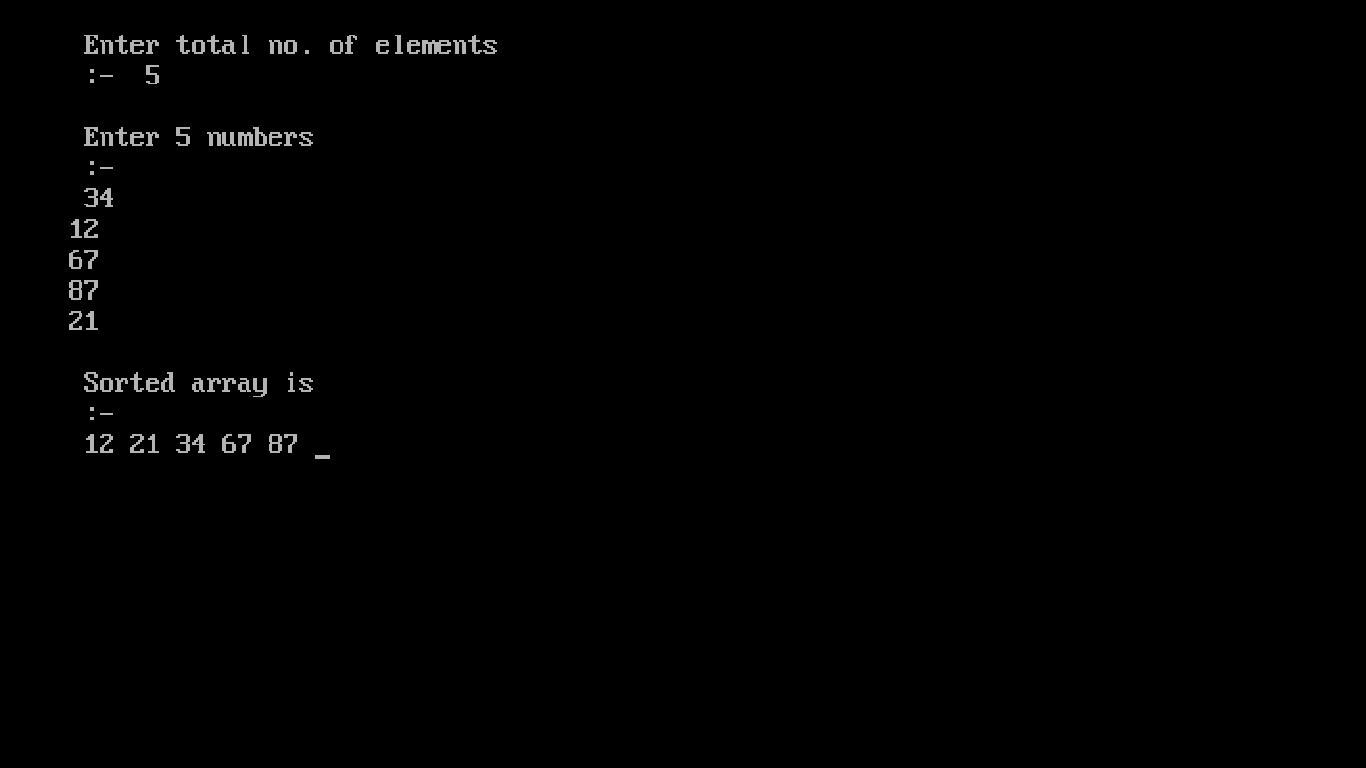
for (n = 0; n < num; n++)

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

getch();

}

**OUTPUT IS:-**



Ques :12- Write a program for fractional Knapsack problem using Greedy Approach?

Ans 12 :-

# include<stdio.h>

# include<conio.h>

void knapsack(int n, float weight[], float profit[], float capacity)

{

float x[20], tp = 0;

int i, j, u;

u = capacity;

for (i = 0; i < n; i++) x[i] = 0.0;

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

{

if (weight[i] > u)

break;

else

{

x[i] = 1.0;

tp = tp + profit[i]; u = u - weight[i];

}

}

if (i < n)

x[i] = u / weight[i];

tp = tp + (x[i] \* profit[i]);

printf("\nThe result vector is:- ");

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

printf("%f\t", x[i]);

printf("\nMaximum profit is:- %f", tp);

}

void main()

{

clrscr();

float weight[20], profit[20], capacity; int num, i, j;

float ratio[20], temp;

printf("Enter the no. of objects \n :- ");

scanf("%d", &num);

printf("Enter the wts and profits of each object \n :- ");

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

{

scanf("%f %f", &weight[i], &profit[i]);

}

printf("Enter the capacityacity of knapsack \n :- ");

scanf("%f", &capacity);

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

{

ratio[i] = profit[i] / weight[i];

}

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

{

for (j = i + 1; j < num; j++)

{

if (ratio[i] < ratio[j])

{

temp = ratio[j];

ratio[j] = ratio[i];

ratio[i] = temp;

temp = weight[j];

weight[j] = weight[i];

weight[i] = temp;

temp = profit[j];

profit[j] = profit[i];

profit[i] = temp;

}

}

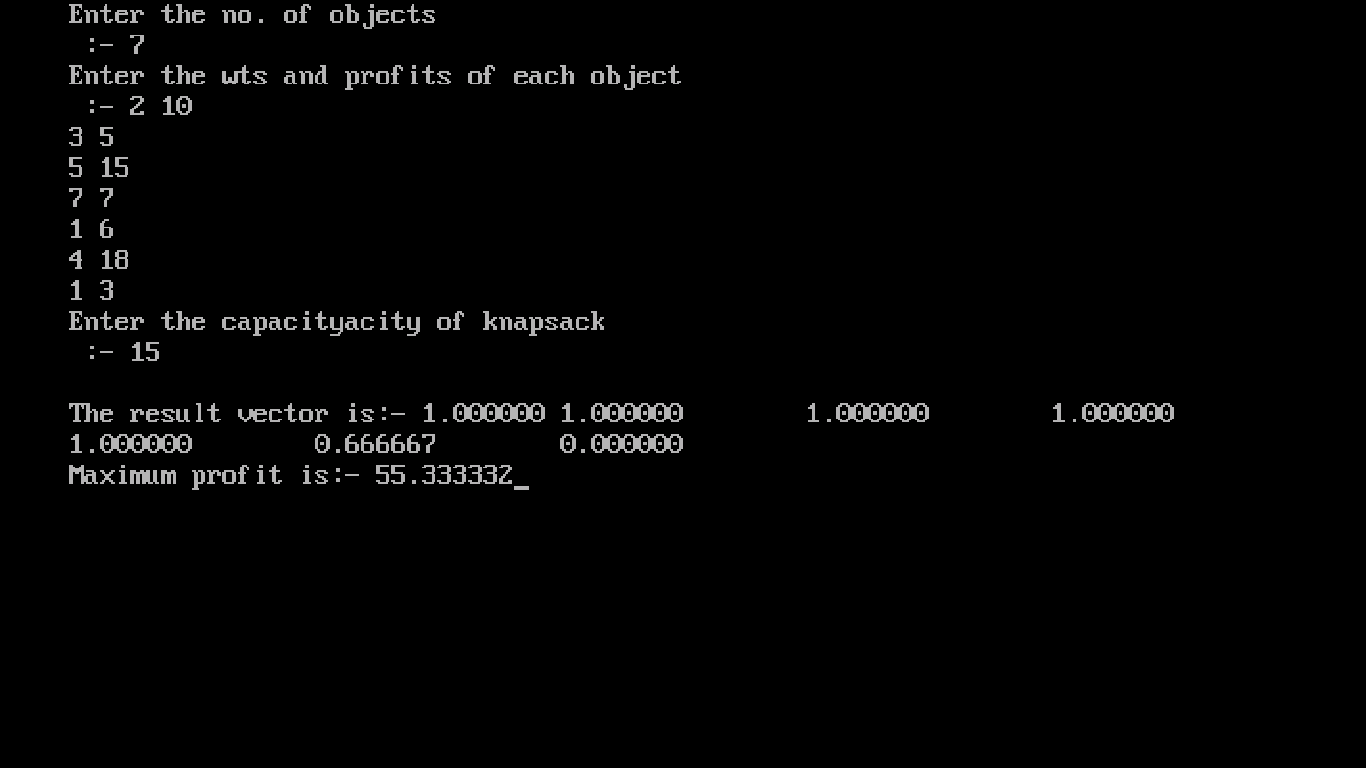
}

knapsack(num, weight, profit, capacity);

getch();

}

**OUTPUT IS:-**



Ques :13- Write a program for TREE SORT?

Ans 13:-

#include<stdio.h>

#include<conio.h>

#include<alloc.h>

struct node{

int info;

struct node \*lp;

struct node \*rp;

};

void inorder(int arr[], struct node\* root)

{

if(root!=NULL)

{

static int i = 0;

inorder(arr,root->lp);

arr[i++]=root->info;

inorder(arr,root->rp);

}

}

void main()

{

clrscr();

int arr[10],n;

printf("Enter the size of array:- \n");

scanf("%d",&n);

printf("\nEnter %d array elements:- \n",n);

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

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

struct node \*head = (struct node \*)malloc(sizeof(struct node));

struct node \*ptr = (struct node \*)malloc(sizeof(struct node));

ptr->info = arr[0];

ptr->lp = NULL;

ptr->rp = NULL;

head = ptr;

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

{

ptr = head;

struct node \*next = (struct node \*)malloc(sizeof(struct node));

next->info = arr[i];

next->lp = NULL;

next->rp = NULL;

int flag;

do

{

flag = 0;

if((next->info)<(ptr->info))

{

if(ptr->lp==NULL)

{

ptr->lp = next;

flag = 1;

}

else

{

ptr = ptr->lp;

}

}

else

{

if(ptr->rp==NULL)

{

ptr->rp = next;

flag = 1;

}

else

{

ptr = ptr->rp;

}

}

}

while(flag==0);

}

inorder(arr,head);

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

printf("\nSorted array:\n");

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

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

getch();

}

**OUTPUT IS:-**



Ques :14- Write a program for strassen’s matrix multiplication?

Ans 14:-

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

int a[2][2], b[2][2], c[2][2], i, j;

int m1, m2, m3, m4 , m5, m6, m7;

printf("Enter the 4 elements of first matrix: ");

for(i = 0;i < 2; i++)

for(j = 0;j < 2; j++)

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

printf("Enter the 4 elements of second matrix: ");

for(i = 0; i < 2; i++)

for(j = 0;j < 2; j++)

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

printf("\nThe first matrix is\n");

for(i = 0; i < 2; i++)

{

printf("\n");

for(j = 0; j < 2; j++)

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

}

printf("\nThe second matrix is\n");

for(i = 0;i < 2; i++)

{

printf("\n");

for(j = 0;j < 2; j++)

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

}

m1= (a[0][0] + a[1][1]) \* (b[0][0] + b[1][1]);

m2= (a[1][0] + a[1][1]) \* b[0][0];

m3= a[0][0] \* (b[0][1] - b[1][1]);

m4= a[1][1] \* (b[1][0] - b[0][0]);

m5= (a[0][0] + a[0][1]) \* b[1][1];

m6= (a[1][0] - a[0][0]) \* (b[0][0]+b[0][1]);

m7= (a[0][1] - a[1][1]) \* (b[1][0]+b[1][1]); c[0][0] = m1 + m4- m5 + m7;

c[0][1] = m3 + m5;

c[1][0] = m2 + m4;

c[1][1] = m1 - m2 + m3 + m6;

printf("\nAfter multiplication using Strassen's algorithm \n");

for(i = 0; i < 2 ; i++)

{

printf("\n");

for(j = 0;j < 2; j++)

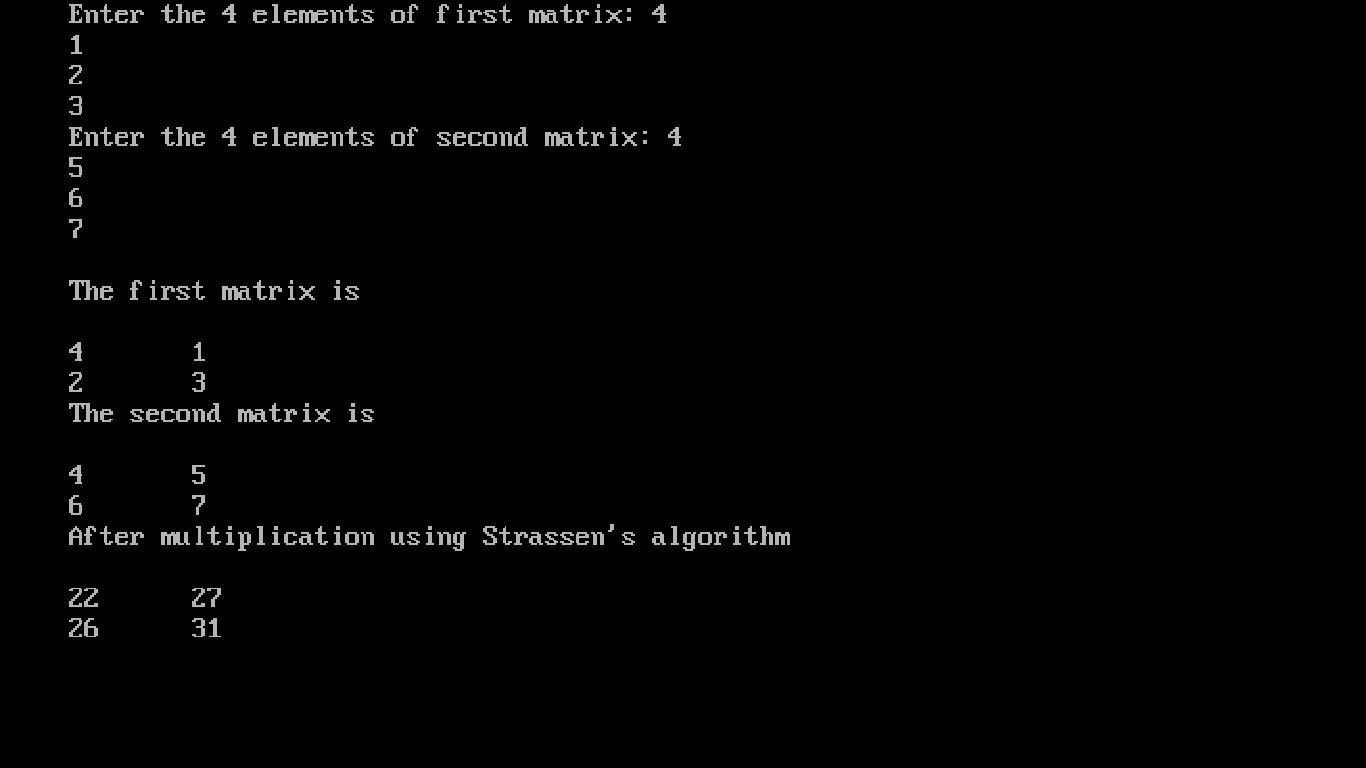
printf("%d\t", c[i][j]);

}

getch();

}

**OUTPUT IS:-**



Ques :15- Write a program for LONGEST COMMON SUBSEQUENCE?

Ans 15:-

#include<stdio.h>

#include<string.h>

int i,j,m,n,c[20][20];

char x[20],y[20],b[20][20];

void print(int i,int j)

{

if(i==0 || j==0)

return;

if(b[i][j]=='c')

{

print(i-1,j-1);

printf("%c",x[i-1]);

}

else

if(b[i][j]=='u')

print(i-1,j);

else

}

print(i,j-1);

void lcs()

{

m=strlen(x);

n=strlen(y);

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

c[i][0]=0;

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

c[0][i]=0;

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

for(j=1;j<=n;j++)

{

if(x[i-1]==y[j-1])

{

c[i][j]=c[i-1][j-1]+1;

b[i][j]='c';

}

else if(c[i-1][j]>=c[i][j-1])

{

}

int main()

{

}

else

{

}

}

c[i][j]=c[i-1][j]; b[i][j]='u';

c[i][j]=c[i][j-1]; b[i][j]='l';

printf("Enter 1st sequence:");

scanf("%s",x);

printf("Enter 2nd sequence:");

scanf("%s",y);

printf("\nThe Longest Common Subsequence is ");

lcs();

print(m,n);

return 0;

}

**OUTPUT IS:-**

Ques :16- Write a program for MERGE SORT?

Ans 16:-

#include<stdio.h>

#include<conio.h>

void mergesort(int a[],int i,int j);

void merge(int a[],int i1,int j1,int i2,int j2);

int main()

{

int a[30],n,i;

printf("Enter no of elements:");

scanf("%d",&n);

printf("Enter array elements:");

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

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

mergesort(a,0,n-1);

printf("\nSorted array is :");

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

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

return 0;

}

void mergesort(int a[],int i,int j)

{

int mid;

if(i<j)

{

mid=(i+j)/2;

mergesort(a,i,mid);

mergesort(a,mid+1,j);

merge(a,i,mid,mid+1,j);

}

}

void merge(int a[],int i1,int j1,int i2,int j2)

{

int temp[50];

int i,j,k;

i=i1;

j=i2;

k=0;

while(i<=j1 && j<=j2)

{

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

temp[k++]=a[i++];

else

}

temp[k++]=a[j++];

while(i<=j1)

temp[k++]=a[i++];

while(j<=j2)

temp[k++]=a[j++];

for(i=i1,j=0;i<=j2;i++,j++)

a[i]=temp[j];

}

**OUTPUT IS:-**

Ques :17 - Write a program for CHAIN MATRIX MULTIPLICATION?

Ans 17:-

#include <stdio.h>

#include <conio.h>

#include <limits.h>

#define INFY 999999999

long int m[20][20];

int s[20][20];

int p[20],i,j,n;

void print\_optimal(int i,int j)

{

if (i == j)

printf(" A%d ",i);

else

{

printf("( ");

print\_optimal(i, s[i][j]);

print\_optimal(s[i][j] + 1, j);

printf(" )");

}

}

void matmultiply(void)

{

long int q;

int k;

for(i=n;i>0;i--)

{

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

{

if(i==j) m[i][j]=0;

else

{

for(k=i;k<j;k++)

{

q=m[i][k]+m[k+1][j]+p[i-1]\*p[k]\*p[j];

if(q<m[i][j])

{

m[i][j]=q;

s[i][j]=k;

}

}

}

}

}

}

int MatrixChainOrder(int p[], int i, int j)

{

if(i == j)

return 0;

int k;

int min = INT\_MAX;

int count;

for (k = i; k <j; k++)

{

count = MatrixChainOrder(p, i, k) + MatrixChainOrder(p, k+1, j) + p[i-1]\*p[k]\*p[j];

if (count < min) min = count;

}

return min;

}

void main()

{

clrscr();

int k;

printf("Enter the no. of elements: ");

scanf("%d",&n);

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

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

{

m[i][i]=0;

m[i][j]=INFY;

s[i][j]=0;

}

printf("\nEnter the dimensions: \n");

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

{

printf("P%d: ",k);

scanf("%d",&p[k]);

}

matmultiply();

printf("\nCost Matrix M:\n");

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

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

printf("m[%d][%d]: %ld\n",i,j,m[i][j]);

i=1,j=n;

printf("\nMultiplication Sequence : ");

print\_optimal(i,j);

printf("\nMinimum number of multiplications is : %d ", MatrixChainOrder(p, 1, n));

}

**OUTPUT IS:-**



Ques :18- Write a program for FLOYD WARSHELL ALGO?

Ans 18:-

#include<stdio.h>

#include<conio.h>

int i, j, k,n,dist[10][10];

void floydWarshell ()

{

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

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

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

if (dist[i][k] + dist[k][j] < dist[i][j])

dist[i][j] = dist[i][k] + dist[k][j];

}

int main()

{

clrscr();

int i,j;

printf("enter no of vertices :");

scanf("%d",&n);

printf("\n");

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

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

{

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

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

}

floydWarshell();

printf (" \n\n shortest distances between every pair of vertices \n");

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

{

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

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

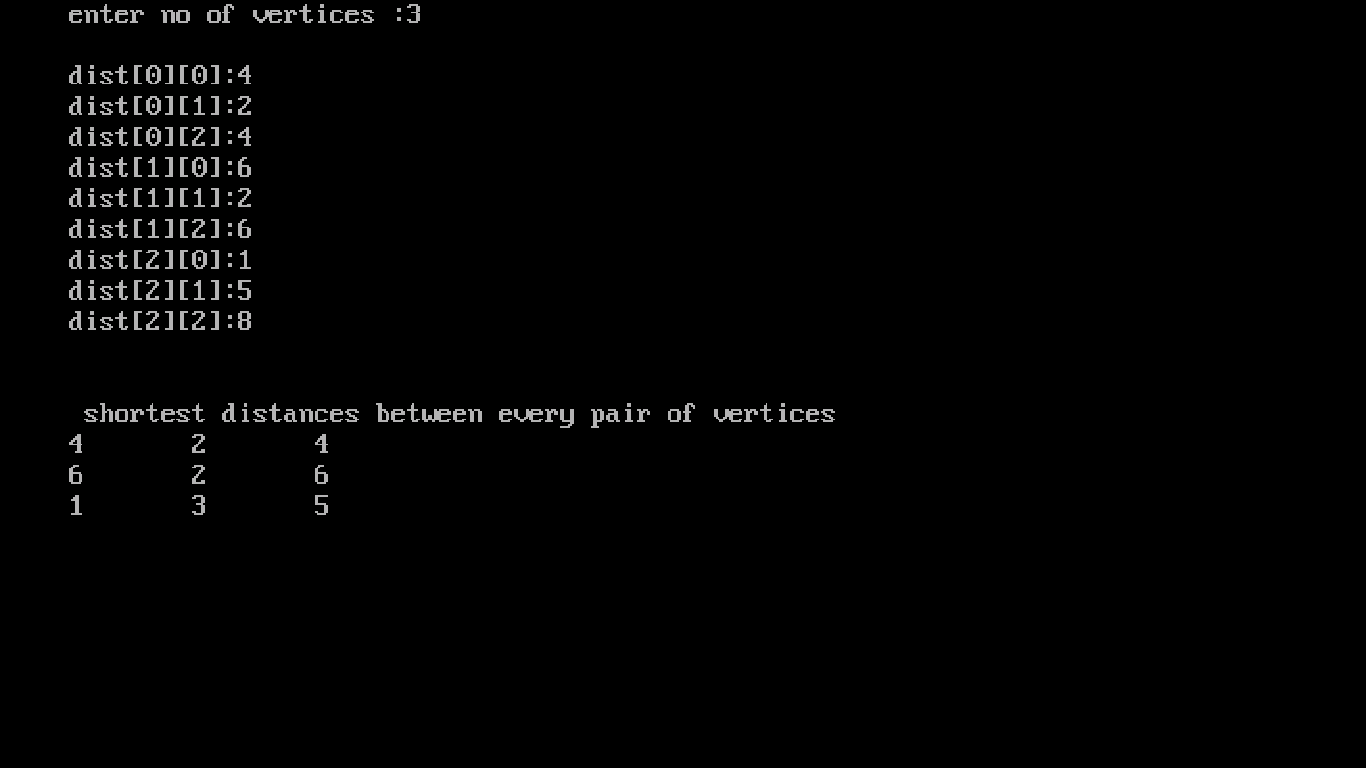
printf("\n");

}

return 0;

}

**OUTPUT IS:-**



Ques :19 - Write a program for GRAPH TRAVERSAL BY BFS/DFS?

Ans 19:-

DFS:-

#include<stdio.h>

#include<conio.h>

int size = 20;

int stack[20],top=-1;

int empty()

{

if(top == -1)

return 0;

return 1;

}

void pop()

{

if(top==-1)

{

printf("Stack is empty\n");

return;

}

stack[top] = NULL;

top--;

}

void push(int x)

{

if(top == size-1)

{

printf("Stack full");

return;

}

top++;

stack[top] = x;

}

void main(){

clrscr();

int adj[10][10],n,status[10],result[10];

printf("Enter number of nodes \n :- ");

scanf("%d",&n);

printf("Enter adjacency matrix \n :- \n");

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

{

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

{

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

}

}

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

status[i] = 1;

i=0;

push(0);

do{

result[i] = stack[top];

pop();

status[result[i]] = 3;

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

{

if((adj[result[i]][j] == 1) && (status[j] == 1))

{

push(j);

status[j] = 2;

}

}

i++;

}while(empty());

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

{

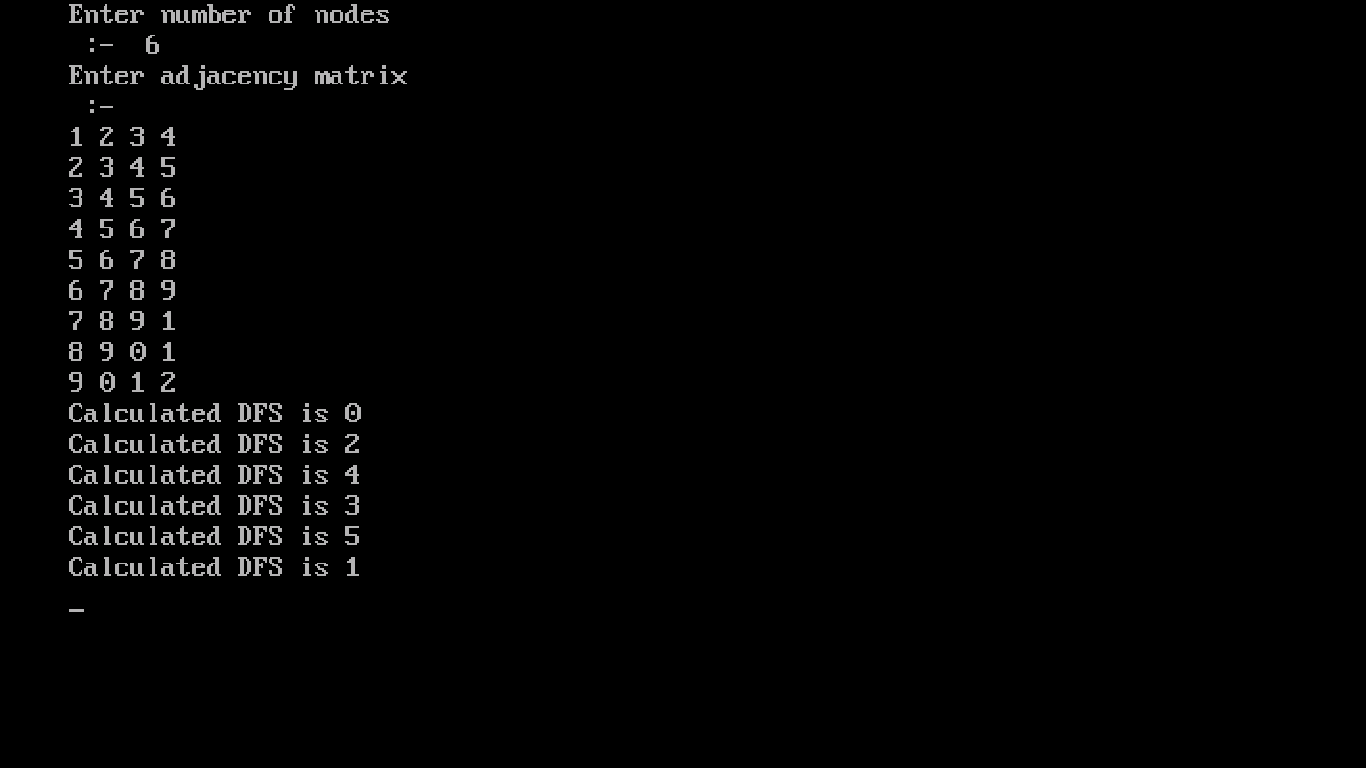
printf("Calculated DFS is %d\n",result[i]);

}

getch();

}

**OUTPUT IS:-**



BFS:-

#include<stdio.h>

#include<conio.h>

int size = 20;

int que[20],front=0,rear=-1;

int empty()

{

if(front>rear)

return 0;

return 1;

}

void pop()

{

if(front>rear)

{

printf("Queue is empty\n");

return;

}

que[front] = NULL;

front++;

}

void push(int x)

{

if(rear==size-1)

{

printf("Queue full");

return;

}

rear++;

que[rear] = x;

}

void main(){

clrscr();

int adj[10][10],n,status[10],result[10];

printf("Enter number of nodes\n:-");

scanf("%d",&n);

printf("\nEnter adjacency matrix:-\n");

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

{

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

{

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

}

}

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

status[i] = 1;

i=0;

push(0);

do{

result[i] = que[front];

pop();

status[result[i]] = 3;

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

{

if((adj[result[i]][j] == 1) && (status[j] == 1))

{

push(j);

status[j] = 2;

}

}

i++;

}while(empty());

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

{

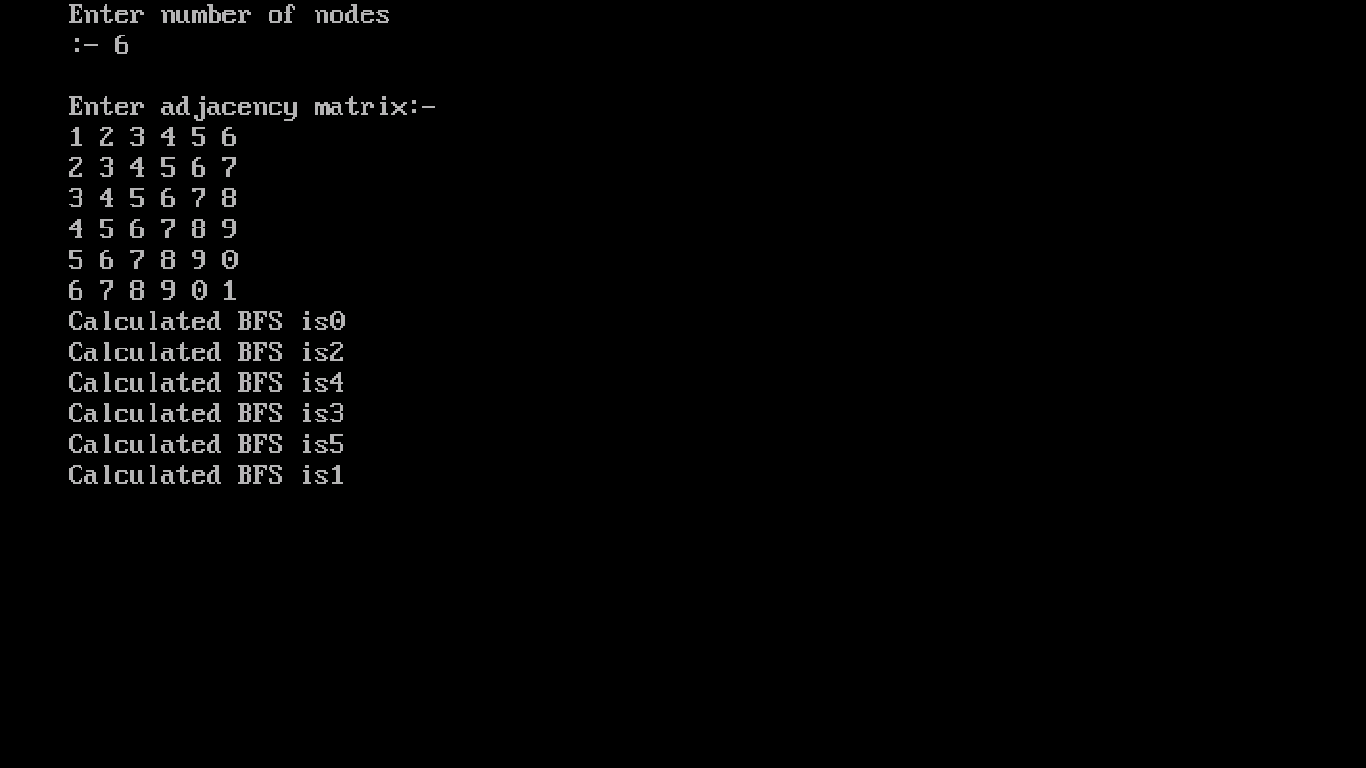
printf("Calculated BFS is%d\n",result[i]);

}

getch();

}

**OUTPUT IS:-**



Ques :20 - Write a program for N-QUEENS problem using backtracking?

Ans 20:-

#include<stdio.h>

#include<stdlib.h>

int t[8] = {-1};

int sol = 1; void printsol()

{

int i,j;

char crossboard[8][8]; for(i=0;i<8;i++)

{

for(j=0;j<8;j++)

{

crossboard[i][j]='\_';

}

}

for(i=0;i<8;i++)

{

crossboard[i][t[i]]='q';

}

for(i=0;i<8;i++)

{

for(j=0;j<8;j++)

{

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

}

printf("\n");

}

}

int empty(int i)

{

int j=0;

while((t[i]!=t[j])&&(abs(t[i]-t[j])!=(i-j))&&j<8)j++; return i==j?1:0;

}

void queens(int i)

{

for(t[i] = 0;t[i]<8;t[i]++)

{

if(empty(i))

{

if(i==7){ printsol();

printf("\n solution %d\n",sol++);

}

else queens(i+1);

}

}

}

int main()

{

queens(0);

printf("\n Total Number of Solutions is %d",sol); return 0;

}

**OUTPUT IS:-**



Ques :21 - Write a programs for MINIMUM SPANNING TREE using Prim's and Kruskal'S algorithm?

Ans 21:-

PRIM’s:-

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<string.h>

#define infinity 99999

#define MAX 20

int G[MAX][MAX],spanning[MAX][MAX],n;

int prims();

int main()

{

clrscr();

int i,j,Min\_cost;

printf("Enter no. of vertices:");

scanf("%d",&n);

printf("\nEnter the adjacency matrix:\n");

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

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

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

Min\_cost=prims();

printf("\nspanning tree matrix:\n");

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

{

printf("\n");

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

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

}

printf("\n\nMinimum cost of spanning tree=%d",Min\_cost);

return 0;

}

int prims()

{

int cost[MAX][MAX];

int u,v,min\_distance,distance[MAX],from[MAX];

int visited[MAX],no\_of\_edges,i,min\_cost,j;

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

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

{

if(G[i][j]==0)

cost[i][j]=infinity;

else

cost[i][j]=G[i][j];

spanning[i][j]=0;

}

distance[0]=0;

visited[0]=1;

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

{

distance[i]=cost[0][i];

from[i]=0;

visited[i]=0;

}

min\_cost=0;

no\_of\_edges=n-1;

while(no\_of\_edges>0)

{

min\_distance=infinity;

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

if(visited[i]==0&&distance[i]<min\_distance)

{

v=i;

min\_distance=distance[i];

}

u=from[v];

spanning[u][v]=distance[v];

spanning[v][u]=distance[v];

no\_of\_edges--;

visited[v]=1;

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

if(visited[i]==0&&cost[i][v]<distance[i])

{

distance[i]=cost[i][v];

from[i]=v;

}

min\_cost=min\_cost+cost[u][v];

}

return(min\_cost);

}

**OUTPUT IS:-**

