PRACTICAL NO.:1(A)

AIM: Write a program to store the elements in 1-D array and perform the operations like searching, sorting, reversing the elements.

PROGRAM CODE:

#include<stdio.h> #include<conio.h> intsize,val; [void disp(int size); void sort(int size); void reverse(int size); void search(intval,int size); intarr[20]; intmain()](https://e-next.in)

[{ inti,ch;](https://e-next.in)

[printf("Enter the size of array : "); scanf("%d",&size); for(i=0;i<size;i++)](https://e-next.in)

[{](https://e-next.in)

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

}

do

{

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

printf("1.Display\n"); printf("2.Sorting\n"); printf("3.Reverse\n"); printf("Enter your Choice : "); scanf("%d",&ch); switch(ch)

[{](https://e-next.in)

[case 1:disp(size); break; case 2:sort(size); break; case 3:reverse(size); break;](https://e-next.in)

[case 4:printf("Enter value to be search : "); scanf("%d",&val); search(val,size); break;](https://e-next.in)

[}](https://e-next.in)

}

while(ch!=4); getch (); return 0;

}

void search(intval,int size)

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

{

if(arr[i]==val)

{

[printf("Value is found at %d position.",i); break;](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[if(i==size)](https://e-next.in)

[{](https://e-next.in)

[printf("Value is not found.");](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[void disp(int size)](https://e-next.in)

[{ inti;](https://e-next.in)

printf("Given Array :\n"); for(i=0;i<size;i++)

{

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

} } void sort(size)

{ inti,j;

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

{

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

[{](https://e-next.in)

[if(arr[j]>arr[j+1])](https://e-next.in)

[{](https://e-next.in)

[int temp; temp=arr[j]; arr[j]=arr[j+1]; arr[j+1]=temp;](https://e-next.in)

[} }](https://e-next.in)

[}](https://e-next.in)

[printf("Sorted Array : \n"); for(i=0;i<size;i++)](https://e-next.in)

{

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

}

}

void reverse(size)

{

inti,j,temp; j=size-1; i=0; while(i<j)

{

temp=arr[i]; [arr[i]=arr[j]; arr[j]=temp; i++;](https://e-next.in)

[j--;](https://e-next.in)

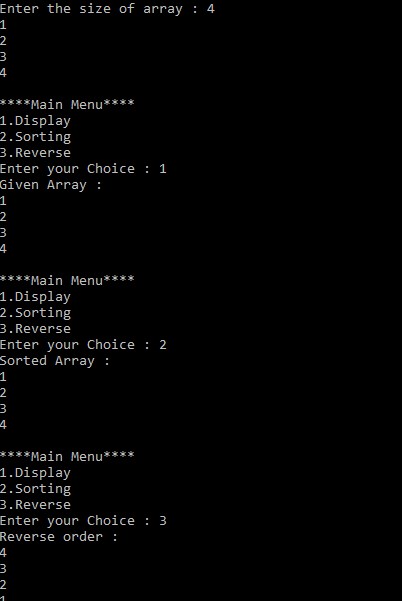
[}](https://e-next.in)

[printf("Reverse order : \n"); for(i=0;i<size;i++)](https://e-next.in)

[{](https://e-next.in)

[printf("%d \n",arr[i]);](https://e-next.in)

[} }](https://e-next.in) OUTPUT :



PRACTICAL NO.:1(B)

AIM : Read the two arrays from the user and merge them and display the elements in sorted order.

PROGRAM CODE:

#include<stdio.h> #include<conio.h> [intmain()](https://e-next.in)

[{](https://e-next.in)

[int arr1[20],arr2[20],arr3[40]; inti,j,k,size1,size2,temp;](https://e-next.in)

[printf("Enter the array size of array 1 :"); scanf("%d",&size1);](https://e-next.in)

[printf("Enter the element in arra 1 :\n"); for(i=0;i<size1;i++)](https://e-next.in)

[{](https://e-next.in)

[scanf("%d",&arr1[i]);](https://e-next.in)

[}](https://e-next.in)

[printf("Enter the size of array 2 : ");](https://e-next.in) scanf("%d",&size2);

printf("Enter the element in array2 : \n"); for(j=0;j<size2;j++)

{

scanf("%d",&arr2[j]);

}

for(i=0;i<size1;i++)

{

for(j=0;j<size1-i-1;j++)

{

if(arr1[j]>arr1[j+1])

[{](https://e-next.in)

[temp=arr1[j]; arr1[j]=arr1[j+1]; arr1[j+1]=temp;](https://e-next.in)

[} }](https://e-next.in)

[}](https://e-next.in)

[for(i=0;i<size2;i++)](https://e-next.in)

[{](https://e-next.in)

[for(j=0;j<size2-i-1;j++)](https://e-next.in)

[{](https://e-next.in)

[if(arr2[j]>arr2[j+1])](https://e-next.in)

{

temp=arr2[j]; arr2[j]=arr2[j+1]; arr2[j+1]=temp;

} } }

/\* printf("Sorting array 1\n"); for(i=0;i<size1;i++)

{

printf("%d \n",arr1[i]);

}

[printf("Sorting array 2\n"); for(i=0;i<size2;i++)](https://e-next.in)

[{](https://e-next.in)

[printf("%d \n",arr2[i]); }\*/ i=0; j=0; k=0; while(i<size1 && j<size2)](https://e-next.in)

[{](https://e-next.in)

[if(arr1[i]<arr2[j])](https://e-next.in)

[{](https://e-next.in)

arr3[k]=arr1[i]; i++; k++;

}

else

{

arr3[k]=arr2[j]; j++; k++;

}

}

while(i<size1)

[{](https://e-next.in)

[arr3[k]=arr1[i]; i++; k++;](https://e-next.in)

[}](https://e-next.in)

[while(j<size2)](https://e-next.in)

[{](https://e-next.in)

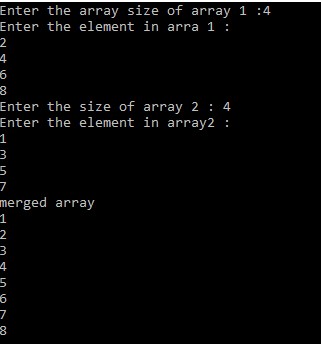
[arr3[k]=arr2[j]; j++; k++;](https://e-next.in)

[}](https://e-next.in)

[printf("merged array \n");](https://e-next.in) for(k=0;k<size1+size2;k++) printf("%d \n",arr3[k]); getch(); return 0;

}

OUTPUT :



PRACTICAL NO.:1(C)

AIM : write a program to perform the Matrix addition, Multiplication, Transpose operation.

PROGRAM CODE :

#include<stdio.h> #include<conio.h>

[int arr1[20][10],arr2[20][10],arr3[20][10]; int r1,c1,r2,c2;](https://e-next.in)

[void addition(int r1,int r2,int c1,int c2); void multiplication(int r1,int r2,int c1,int c2); void transpose1(int r1,int c1); void transpose2(int r2,int c2); void main()](https://e-next.in)

[{](https://e-next.in)

[inti,j,k,choice;](https://e-next.in)

[printf("Enter the matrix 1 size : \n"); scanf("%d %d",&r1,&c1);](https://e-next.in)

[printf("Enter the element in matrix 1 : \n");](https://e-next.in) for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++) scanf("%d",&arr1[i][j]);

}

printf("Matrix 1 : \n"); for(i=0;i<r1;i++)

{

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

{

printf("%d \t",arr1[i][j]);

[}](https://e-next.in)

[printf("\n");](https://e-next.in)

[}](https://e-next.in)

[printf("Enter the matrix 2 size :"); scanf("%d %d",&r2,&c2); printf("Enter the element in matrix 2 : \n"); for(i=0;i<r2;i++)](https://e-next.in)

[{](https://e-next.in)

[for(j=0;j<c2;j++) scanf("%d",&arr2[i][j]);](https://e-next.in)

[}](https://e-next.in)

[printf("Matrix 2 : \n");](https://e-next.in) for(i=0;i<r2;i++)

{

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

{

printf("%d \t",arr2[i][j]);

}

printf("\n");

}

do

{

printf("\*\*\*\*\*\*\* Main Menu\*\*\*\*\*\*\* \n"); printf("1.Addition\n"); [printf("2.Multiplication\n"); printf("3.transpose 1\n"); printf("4.transpose 2\n"); printf("Enter your chioce : "); scanf("%d",&choice); switch(choice)](https://e-next.in)

[{](https://e-next.in)

[case 1 : addition(r1,r2,c1,c2); break;](https://e-next.in)

[case 2: multiplication(r1,r2,c1,c2); break; case 3: transpose1(r1,c1);](https://e-next.in) break; case 4 : transpose2(r2,c2); break;

}

}while(choice>0); getch();

}

void addition(int r1,int r2,int c1,int c2)

{ inti,j;

if(r1==r2 && c1==c2)

{

[printf("Addition of matrix :\n"); for(i=0;i<r1;i++) for(j=0;j<r2;j++) arr3[i][j]=arr1[i][j]+arr2[i][j]; for(i=0;i<r1;i++)](https://e-next.in)

[{](https://e-next.in)

[for(j=0;j<c1;j++)](https://e-next.in)

[{](https://e-next.in)

[printf("%d \t",arr3[i][j]);](https://e-next.in)

[}](https://e-next.in)

[printf("\n");](https://e-next.in)

[}](https://e-next.in)

}

else

{

printf("ORDER INCORRECT. \n");

} } void multiplication(int r1,int r2,int c1,int c2)

{ inti,j,k;

if(c1==r2)

{

printf("Multiplication of matrix :\n"); [for(i=0;i<r1;i++)](https://e-next.in)

[{](https://e-next.in)

[for(j=0;j<c2;j++)](https://e-next.in)

[{](https://e-next.in)

[arr3[i][j]=0; int k; for(k=0;k<r1;k++)](https://e-next.in)

[{](https://e-next.in)

[arr3[i][j]+=arr1[i][k]\*arr2[k][j];](https://e-next.in)

[}](https://e-next.in)

[printf("%d\t",arr3[i][j]);](https://e-next.in)

[} printf("\n");](https://e-next.in)

}

}

else

{

printf("ORDER INCORRECT.\n");

}

}

void transpose1(int r1,int c1)

{ inti,j;

printf("Transpose matrix 1 \n"); for(i=0;i<r1;i++)

[{](https://e-next.in)

[for(j=0;j<c1;j++)](https://e-next.in)

[{ arr3[j][i]=arr1[i][j];](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[for(i=0;i<c1;i++)](https://e-next.in)

[{](https://e-next.in)

[for(j=0;j<r1;j++)](https://e-next.in)

[{](https://e-next.in)

[printf("%d\t",arr3[i][j]);](https://e-next.in)

[}](https://e-next.in)

printf("\n");

}

}

void transpose2(int r2,int c2)

{ inti,j; printf("Transpose matrix 2 \n"); for(i=0;i<r2;i++)

{

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

{ arr3[j][i]=arr1[i][j];

[}](https://e-next.in)

[}](https://e-next.in)

[for(i=0;i<c2;i++)](https://e-next.in)

[{](https://e-next.in)

[for(j=0;j<r2;j++)](https://e-next.in)

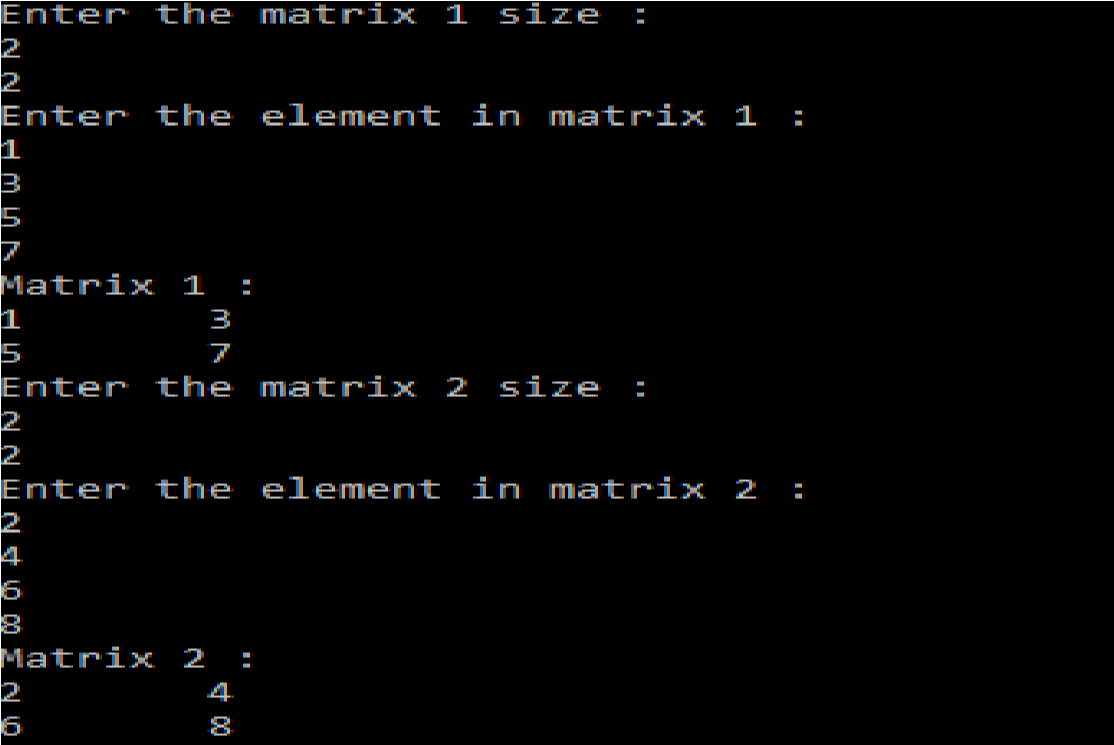
[{](https://e-next.in)

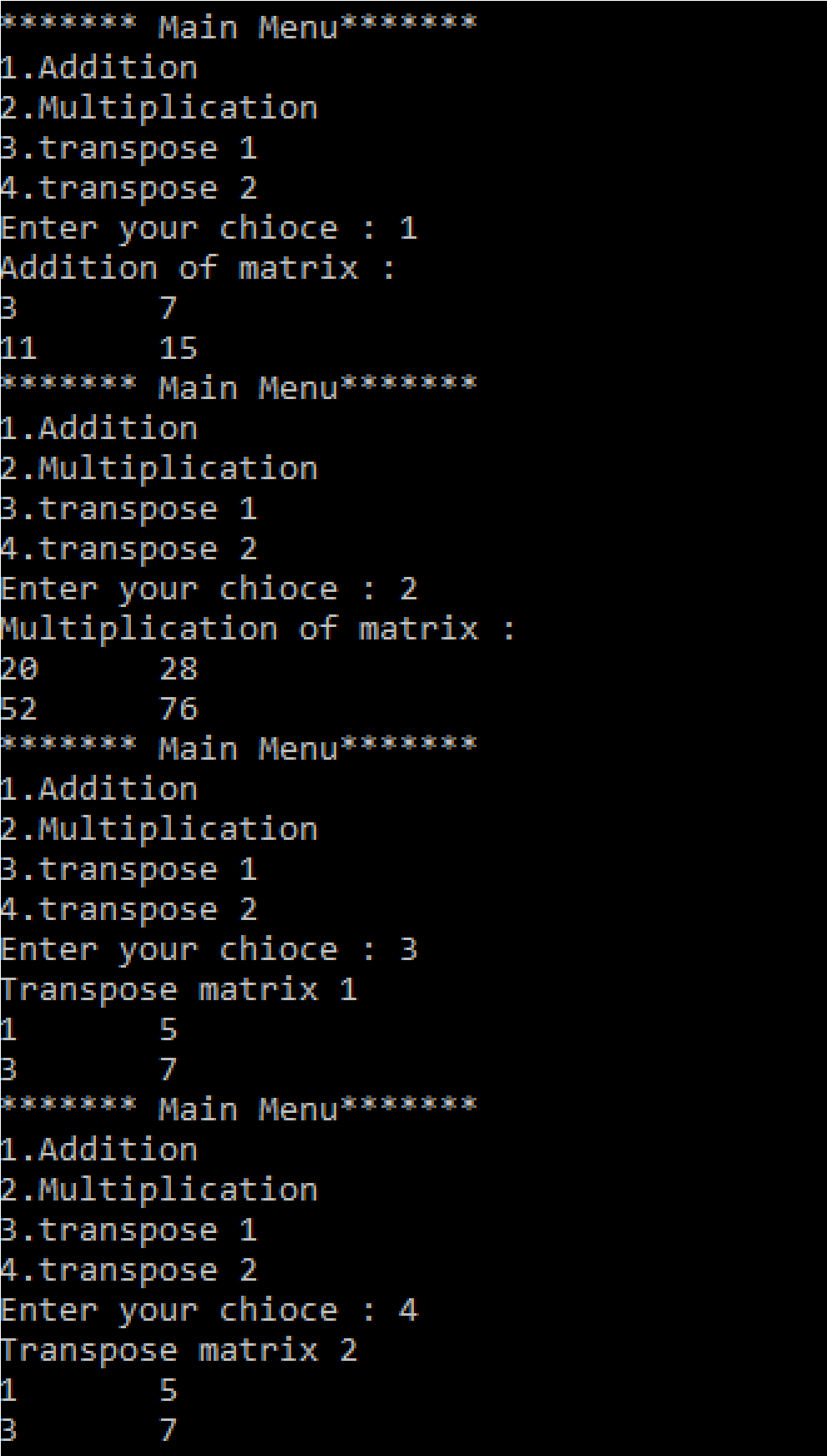
[printf("%d\t",arr3[i][j]);](https://e-next.in)

[}](https://e-next.in)

[printf("\n");](https://e-next.in)

[} }](https://e-next.in) OUTPUT :





PRACTICAL NO.:2(A)

AIM : Write a program to create a single linked list and display the node elements in reserve order.

Pogram code:

#include<stdio.h>

[#include<conio.h>](https://e-next.in)

[#include<malloc.h> #include<stdlib.h> struct node](https://e-next.in)

[{](https://e-next.in)

[int info; struct node \*next;](https://e-next.in)

[};](https://e-next.in)

[struct node \*start=NULL;](https://e-next.in)

[struct node \*create(struct node \*start); struct node \*dispaly(struct node \*start); void reverse(struct node \*start);](https://e-next.in) int main()

{ clrscr(); start=create(start); start=dispaly(start); printf("\n"); printf("Reverse \t"); reverse(start); getch (); return 0;

}

[struct node \*create(struct node \*start)](https://e-next.in)

[{](https://e-next.in)

[struct node \*new\_node=NULL,\*temp=NULL;](https://e-next.in)

[int val;](https://e-next.in)

[printf("Enter -1 value to exit list.\n"); printf("Enter the value : \n"); scanf("%d",&val); while(val!=-1)](https://e-next.in)

[{](https://e-next.in)

[new\_node=(struct node\*)malloc(sizeof(struct node)); new\_node->info=val; if(start==NULL)](https://e-next.in)

{

start=new\_node; new\_node->next=NULL;

}

else

{

temp=start; while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=new\_node; [new\_node->next=NULL;](https://e-next.in)

[}](https://e-next.in)

[printf("Enter the value : \n"); scanf("%d",&val);](https://e-next.in)

[}](https://e-next.in)

[printf("List is successfully created.\n"); return start;](https://e-next.in)

[}](https://e-next.in)

[struct node \*dispaly(struct node \*start)](https://e-next.in)

[{](https://e-next.in)

[struct node \*temp=NULL; temp=start;](https://e-next.in)

printf("List is :\n");

while(temp!=NULL)

{

printf("%d \t",temp->info); temp=temp->next;

}

return start;

}

void reverse(struct node \*start)

{

struct node \*prev=NULL; struct node \*current=start; [struct node \*next\_node; while(current!=NULL)](https://e-next.in)

[{](https://e-next.in)

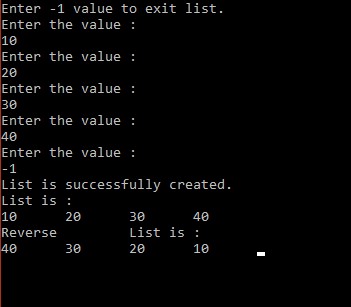
[next\_node=current->next; current->next=prev; prev=current; current=next\_node;](https://e-next.in)

[}](https://e-next.in)

[start=prev; start=dispaly(start);](https://e-next.in)

[}](https://e-next.in)

OUTPUT :



PRACTICAL NO.:2(B)

AIM : Write a program to search the elements in the linked list and display the same.

PROGRAM CODE:

#include<stdio.h>

#include<conio.h>

[#include<malloc.h> #include<stdlib.h> struct node](https://e-next.in)

[{](https://e-next.in)

[int info; struct node \*next;](https://e-next.in)

[};](https://e-next.in)

[struct node \*start=NULL;](https://e-next.in)

[struct node \*create(struct node \*start); struct node \*dispaly(struct node \*start); struct node \*search(struct node \*start); int main()](https://e-next.in)

{

start=create(start); start=dispaly(start); printf("\n"); start=search(start); getch (); return 0;

}

struct node \*create(struct node \*start)

{

struct node \*new\_node=NULL,\*temp=NULL;

[int val;](https://e-next.in)

[printf("Enter -1 value to exit list.\n"); printf("Enter the value : \n"); scanf("%d",&val); while(val!=-1)](https://e-next.in)

[{](https://e-next.in)

[new\_node=(struct node\*)malloc(sizeof(struct node)); new\_node->info=val; if(start==NULL)](https://e-next.in)

[{](https://e-next.in)

[start=new\_node; new\_node->next=NULL;](https://e-next.in)

}

else

{

temp=start; while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=new\_node; new\_node->next=NULL;

}

printf("Enter the value : \n"); [scanf("%d",&val);](https://e-next.in)

[}](https://e-next.in)

[printf("List is successfully created.\n"); return start;](https://e-next.in)

[}](https://e-next.in)

[struct node \*dispaly(struct node \*start)](https://e-next.in)

[{](https://e-next.in)

[struct node \*temp=NULL; temp=start;](https://e-next.in)

[printf("List is :\n");](https://e-next.in)

[while(temp!=NULL)](https://e-next.in)

[{](https://e-next.in)

printf("%d \t",temp->info); temp=temp->next;

}

return start;

}

struct node \*search(struct node \*start) {

int val,count; struct node \*temp;

printf("\nwhich value are you looking for?\n"); scanf("%d",&val); count=1; [temp=start;](https://e-next.in)

[while(temp->info!=val && temp->next!=NULL)](https://e-next.in)

[{](https://e-next.in)

[temp=temp->next; count++;](https://e-next.in)

[}](https://e-next.in)

[//temp=temp->next; if(temp->next==NULL && temp->info!=val)](https://e-next.in)

[{](https://e-next.in)

[printf("value not found");](https://e-next.in)

[}](https://e-next.in)

[else if(temp->next==NULL && temp->info==val)](https://e-next.in)

{

printf("value found at %d node",count);

}

else

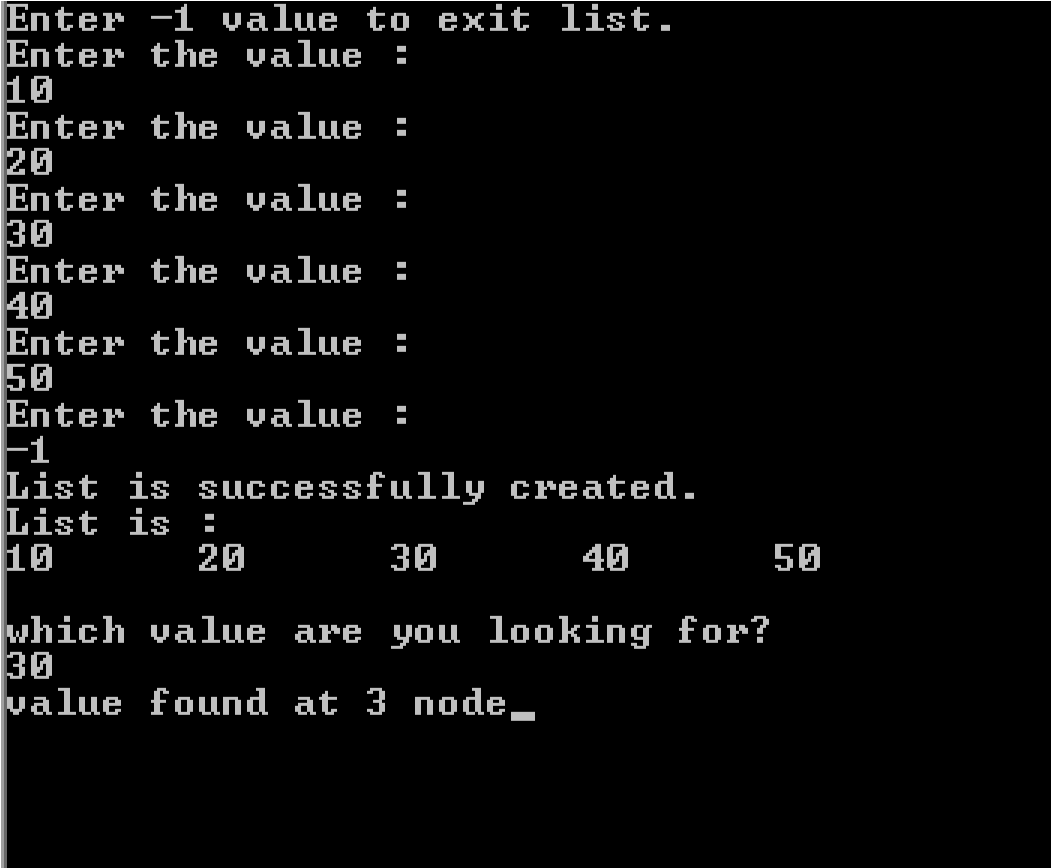
{

printf("value found at %d node",count);

return start;

}

OUTPUT :



PRACTICAL NO:-2(C)

AIM:Write a program to create double linked list and sort the elements in the linked list.

PROGRAM CODE:

#include<stdio.h>

#include<conio.h>

#include<malloc.h> [#include<stdlib.h> struct node {int data; struct node \*next; struct node \*prev;](https://e-next.in)

[};](https://e-next.in)

[struct node \*start=NULL; struct node \*create(struct node \*start); struct node \*display(struct node \*start); struct node \*sort(struct node \*start); int main()](https://e-next.in)

[{](https://e-next.in)

start=create(start); start=display(start); printf("\n"); printf("sort \t"); start=sort(start); struct node \*create(struct node \*start)

{

struct node \*new\_node=NULL,\*temp=NULL,prev; int val;

printf("Enter the data or enter -1 to exit:"); [scanf("%d",&val); while(val!=-1)](https://e-next.in)

[{](https://e-next.in)

[new\_node=(struct node\*)malloc(sizeof(struct node)); new\_node->data=val; if(start==NULL)](https://e-next.in)

[{](https://e-next.in)

[start=new\_node; new\_node->next=NULL; new\_node->prev=NULL;](https://e-next.in)

[}](https://e-next.in)

[else](https://e-next.in)

{

temp=start; while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=new\_node; new\_node->prev=NULL; new\_node->next=NULL;

}

printf("enter the data or enter -1 to exit:"); scanf("%d",&val);

[}](https://e-next.in)

[printf("Linked list successfully created.\n"); return start;](https://e-next.in)

[}](https://e-next.in)

[struct node \*display(struct node \*start)](https://e-next.in)

[{](https://e-next.in)

[struct node \*temp=NULL; temp=start; printf("\nThe Linked list is:"); while(temp->next!=NULL)](https://e-next.in)

[{](https://e-next.in)

[printf("\t %d \t",temp->data);](https://e-next.in) temp=temp->next;

}

if(temp->next==NULL) printf("%d \n",temp->data); printf("\n"); return start;

struct node \*sort(struct node\*start)

{

struct node \*temp1=start; struct node \*temp2,\*temp;

int x;

[while (temp1->next!=NULL)](https://e-next.in)

[{](https://e-next.in)

[temp2=start; while(temp2->next!=NULL)](https://e-next.in)

[{](https://e-next.in)

[temp=temp2->next; if(temp2->data>temp->data)](https://e-next.in)

[{](https://e-next.in)

[x=temp->data; temp->data=temp2->data; temp2->data=x;](https://e-next.in)

[}](https://e-next.in)

temp2=temp2->next;

}

temp1=temp1->next;

}

temp=start; printf("The Linked List is:"); while(temp->next!=NULL)

{

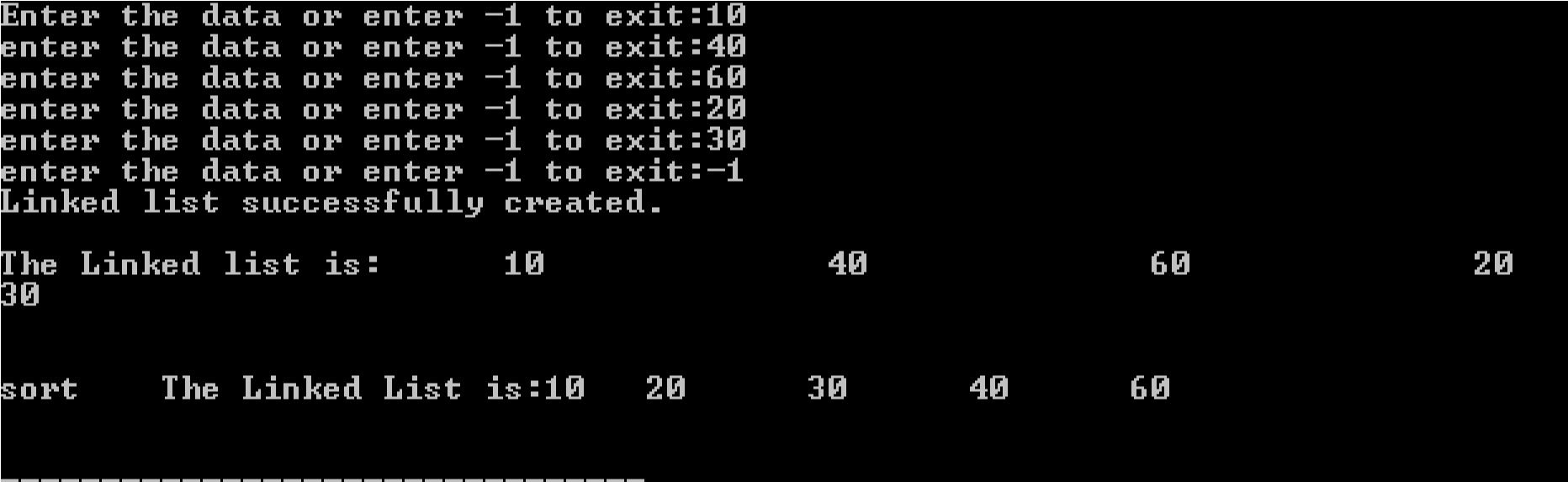
printf("%d \t",temp->data); temp=temp->next;

}

if(temp->next==NULL) [printf("%d \n",temp->data); printf("\n"); return start;](https://e-next.in)

[}](https://e-next.in)

[OUTPUT :](https://e-next.in)



PRACTICAL NO:-3(A)

AIM : Write a program to implement the concept of stack with Push, Pop, Display and exit operation.

PROGRAM CODE:

#include<stdio.h>

#include<conio.h> [#define MAX 30 int stack[MAX]; int top =-1; //Stack is empty.](https://e-next.in)

[void push(); int pop(); int peek(); void display(); int main()](https://e-next.in)

[{](https://e-next.in)

[int choice; do](https://e-next.in)

[{](https://e-next.in)

printf("\n \*\*\*\* Main Menu \*\*\*\* \n"); printf("1.Push\n"); printf("2.Pop \n"); printf("3.Peek \n"); printf("4.Display \n");

printf("Enter your choice :"); scanf("%d",&choice); printf("\n"); switch(choice)

{

case 1: push(); [break; case 2 : pop(); break; case 3 : peek(); break; case 4 : display(); break; case 5 : break;](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[while(choice!=5); return 0;](https://e-next.in)

}

void push()

{ int val; if(top == MAX -1)

{

printf("Stack is full.");

} else

{

printf("Enter the value to be pushed : "); scanf("%d",&val); stack[++top]=val; [printf("Successfully pushed.\n");](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[int pop()](https://e-next.in)

[{](https://e-next.in)

[if(top == -1)](https://e-next.in)

[{](https://e-next.in)

[printf("Stack is already empty.");](https://e-next.in)

[}](https://e-next.in)

[else](https://e-next.in)

[{](https://e-next.in)

[int val = stack[top];](https://e-next.in) top--;

printf("The value is popped : %d",val);

}

}

int peek()

{

if(top == -1)

{

printf("Stack is empty.");

}

else

{

[int topmost = stack[top];](https://e-next.in)

[printf("The topmost element of stack : %d ",topmost);](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[void display()](https://e-next.in)

[{](https://e-next.in)

[if(top == -1)](https://e-next.in)

[{](https://e-next.in)

[printf("Stack is empty.");](https://e-next.in)

[}](https://e-next.in)

[else](https://e-next.in)

[{](https://e-next.in) int i; printf("Stack is : "); for(i=top;i>=0;i--)

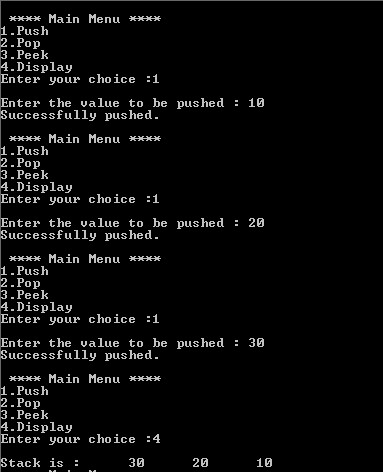
{

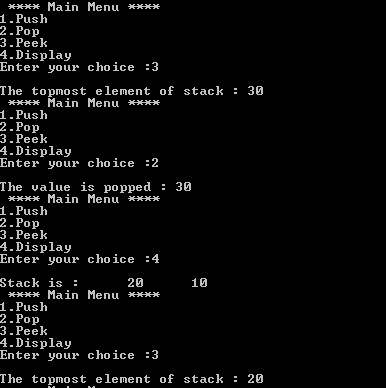
printf("\t%d",stack[i]);

}

} }

OUTPUT :





PRACTICAL NO.:3(B) AIM : Write a program to implement Tower of Hanio problem .

PROGRAM CODE:

[#include<stdio.h> #include<conio.h>](https://e-next.in)

[int move(int n,char source,char temp,char destination); int main()](https://e-next.in)

[{ int n;](https://e-next.in)

[printf("enter number of Disk"); scanf("%d",&n); move(n,'A','B','C'); getch(); return 0;](https://e-next.in)

[}](https://e-next.in)

int move(int n,char source,char temp,char destination)

{

if(n == 1)

{

printf("\n Move from %c to %c",source,destination);

}

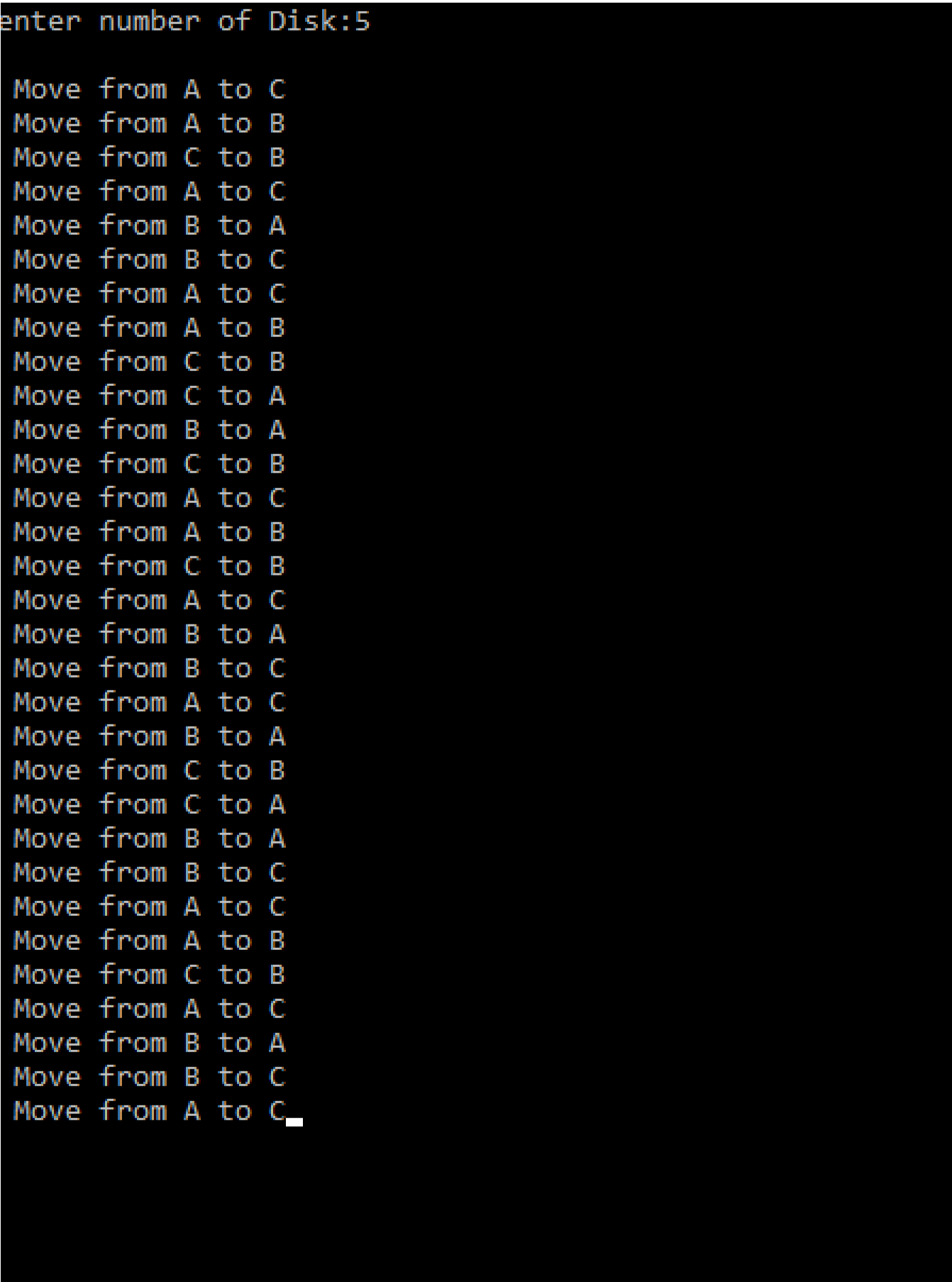
else

{

move(n-1,source,destination,temp); move(1,source,temp,destination); move(n-1,temp,source,destination);

} [}](https://e-next.in)

OUTPUT :



PRACTICAL NO:-4(A)

AIM : Write a program to implement the concept of Queue with Insert, Delete, Display and exit operation.

PROGRAM CODE:

#include<stdio.h>

[#include<conio.h> #define max 30 int rear=-1; int front=-1; void insert(); int deleteq(); void display(); int q[max]; void insert()](https://e-next.in)

[{](https://e-next.in)   [int val;](https://e-next.in)

[printf("Enter value to be inserted :");](https://e-next.in) scanf("%d",&val); if(rear==max-1)

{

printf("Queue is full.");

}

else if(front==-1)

{

front=rear=0; q[rear]=val;

printf("Value inserted successfully.");

}

[else](https://e-next.in)

[{](https://e-next.in)

[q[++rear]=val; printf("Value inserted successfully.");](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[int deleteq()](https://e-next.in)

[{](https://e-next.in)

[if(front==-1)](https://e-next.in)

[{](https://e-next.in)

[printf("Queue is already empty."); return -1;](https://e-next.in)

}

else if(front==rear) //Only one item is present.

{ int val ; val=q[front]; front=rear=-1;

printf("Value to be deletede : %d",val);

return val;

}

else

{ int val ; val=q[front]; [front++;](https://e-next.in)

[printf("Value to be deletede : %d",val); return val;](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[void display()](https://e-next.in)

[{](https://e-next.in)

[if(front==-1)](https://e-next.in)

[{](https://e-next.in)

[printf("Queue is empty.");](https://e-next.in)

[}](https://e-next.in)

[else](https://e-next.in)

{ int i;

printf("Queue is :"); for(i=front;i<=rear;i++)

{

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

} }

}

int main()

{

int choice; [do](https://e-next.in)

[{](https://e-next.in)

[printf("\n \*\*\*\* Main Menu \*\*\*\* \n"); printf("1.Insert\n"); printf("2.Delete\n"); printf("3.Display \n"); printf("Enter your choice :"); scanf("%d",&choice); printf("\n"); switch(choice)](https://e-next.in)

[{](https://e-next.in)

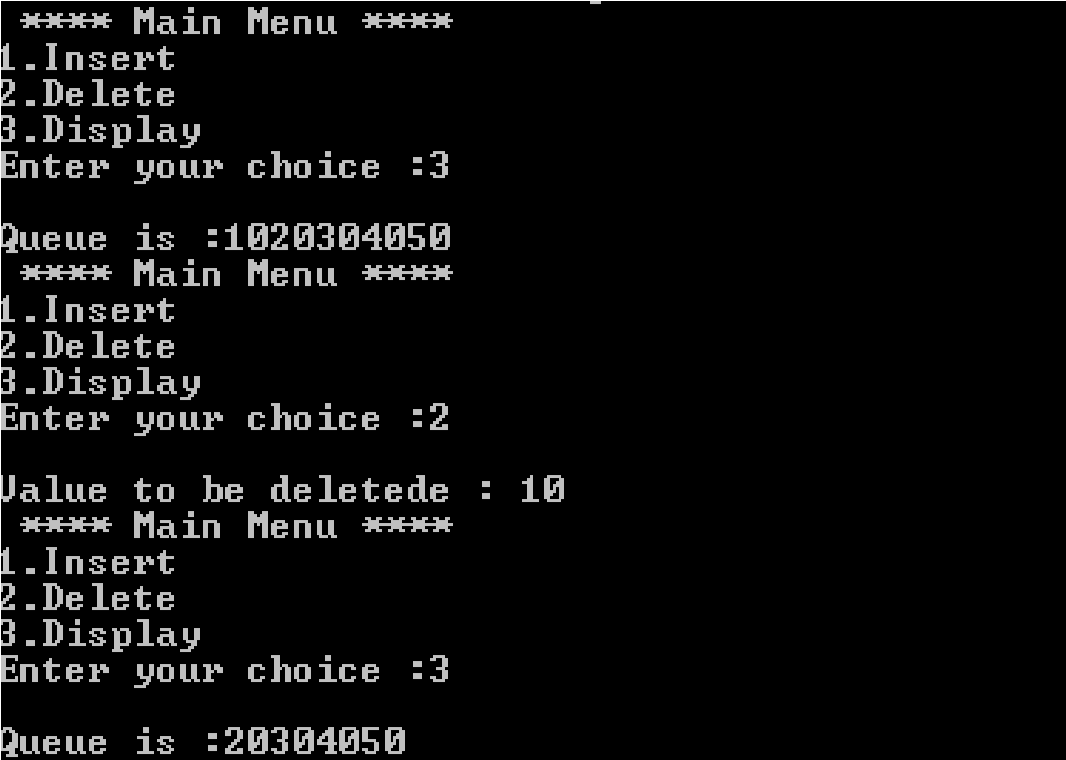
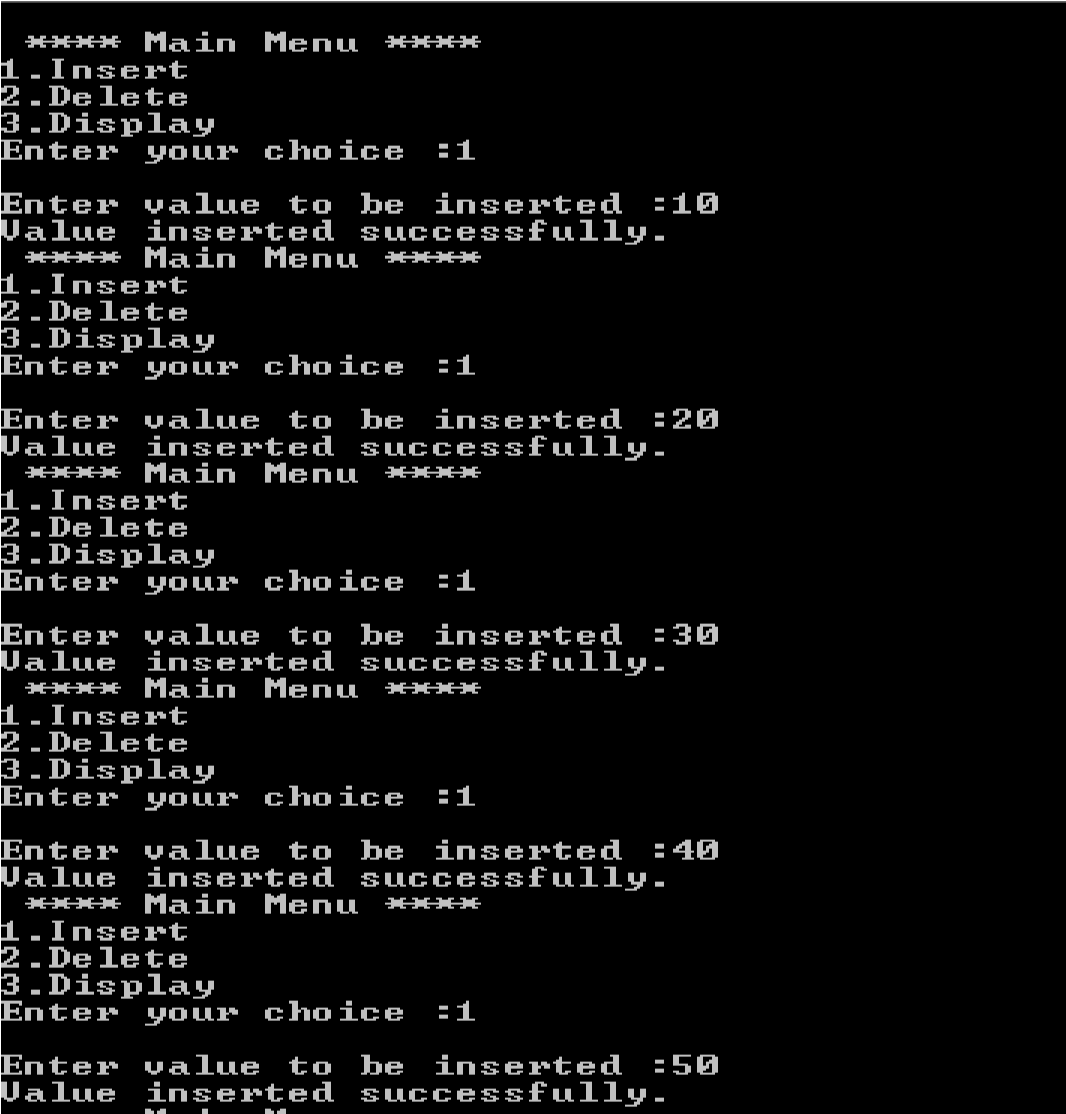
[case 1: insert();](https://e-next.in) break; case 2 : deleteq(); break; case 3 : display(); break; case 4 : break;

} }

while(choice!=4); return 0;

}

OUTPUT :



PRACTICAL NO:-4(B)

AIM : Write a program to implement the concept of circular Queue.

PROGRAM CODE:

#include<stdio.h>

[#include<conio.h> #define max 5 int front = -1; int rear = -1; void insetr(); void display(); intdeleteq(); int q[max]; void insert()](https://e-next.in)

[{](https://e-next.in)

[intval;](https://e-next.in)

[printf("Enter value :");](https://e-next.in) scanf("%d",&val); if((rear+1)%max==front)

{

printf("Queue is full.");

}

else if(rear==-1)

{

rear=front=0; q[rear]=val; printf("Inserted successfully.");

}

Else

[{](https://e-next.in)

[rear=(rear + 1)%max; q[rear]=val; printf("Inserted successfully.");](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[intdeleteq()](https://e-next.in)

[{](https://e-next.in)

[intval; if(front== -1)](https://e-next.in)

[{](https://e-next.in)

[printf("Queue is empty.");](https://e-next.in) return -1;

}

else if(front == rear)

{

intval=q[front]; front=rear= -1; printf("Deleted value : %d",val); return val;

}

else

{

val=q[front]; [front=(front+1)%max; printf("Deleted value : %d",val); return val;](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[void display()](https://e-next.in)

[{ inti; if(front ==-1)](https://e-next.in)

[{](https://e-next.in)

[printf("Queue is empty.");](https://e-next.in)

[}](https://e-next.in)

else

{

printf("Queue is :"); for(i=front;i!=rear;i=(i+1)%max)

{

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

} printf("%d",q[i]);

}

}

intmain()

{ int choice;

[clrscr(); do](https://e-next.in)

[{](https://e-next.in)

[printf("\n \*\*\*\* Main Menu \*\*\*\* \n"); printf("1.Insert\n"); printf("2.Delete\n"); printf("3.Display \n"); printf("Enter your choice :"); scanf("%d",&choice); printf("\n"); switch(choice)](https://e-next.in)

[{](https://e-next.in)

case 1: insert(); break; case 2 :deleteq(); break; case 3 : display(); break;

case 4 : break;

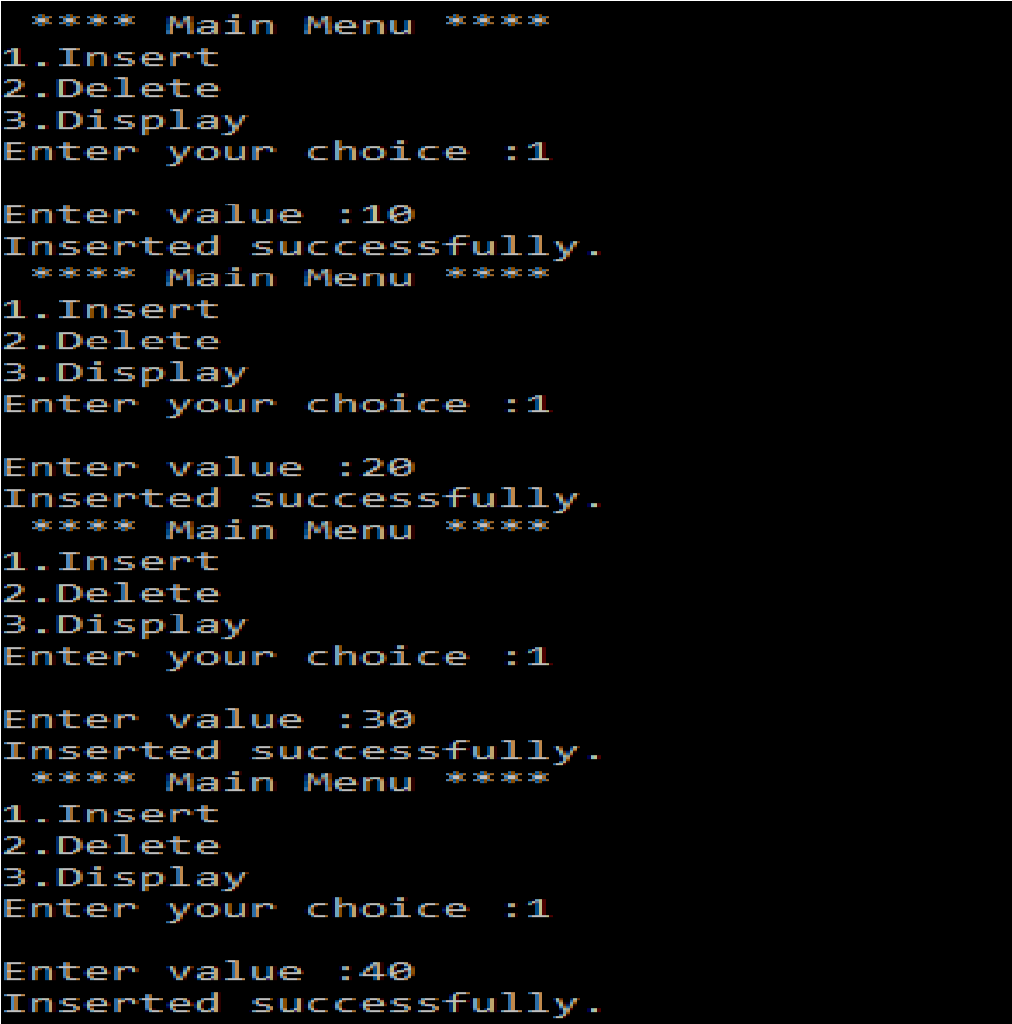
}

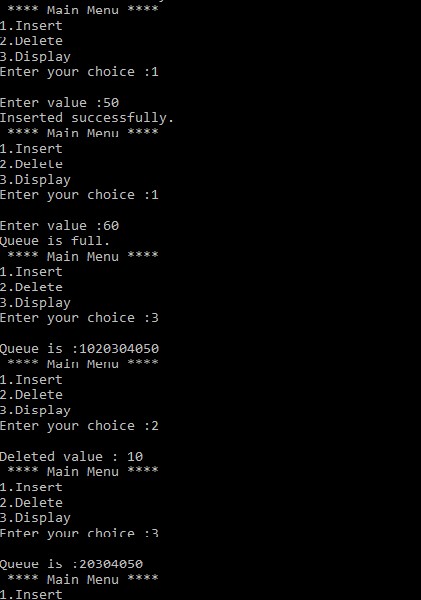
}

while(choice!=4); return 0;

}

[OUTPUT :](https://e-next.in)





PRACTICAL NO:-4(C) AIM : Write a program to implement the concept of Deque.

PROGRAM CODE:

#include<stdio.h>

[#include<conio.h> #define max 5 int front = -1; int rear = -1; intinsert\_rear(); intinsert\_front(); void display(); intdeleteq\_rear(); intdeleteq\_front(); int q[max]; intmain()](https://e-next.in)

[{](https://e-next.in)

int choice; clrscr(); do

{

printf("\n \*\*\*\* Main Menu \*\*\*\* \n"); printf("1.Insert From Rear\n"); printf("2.Insert From Front\n"); printf("3.Delete From Front \n"); printf("4.Delete From Rear \n"); printf("5.Display\n"); printf("Enter your choice :"); scanf("%d",&choice); [printf("\n"); switch(choice)](https://e-next.in)

[{](https://e-next.in)

[case 1: insert\_rear(); break; case 2 :insert\_front(); break; case 3 :deleteq\_front(); break; case 4 :deleteq\_rear(); break; case 5 : display();](https://e-next.in) break; case 6 : break;

}

}

while(choice!=6); return 0;

}

intinsert\_rear()

{ intval;

printf("Enter value :"); scanf("%d",&val); if((rear+1)%max==front)

[{](https://e-next.in)

[printf("Queue is full."); return 0;](https://e-next.in)

[}](https://e-next.in)

[else if(rear==-1)](https://e-next.in)

[{](https://e-next.in)

[rear=front=0; q[rear]=val; printf("Inserted successfully."); return val;](https://e-next.in)

[}](https://e-next.in)

[else](https://e-next.in)

{

rear=(rear + 1)%max; q[rear]=val; printf("Inserted successfully."); return val;

}

}

intinsert\_front()

{ intval;

printf("Enter value :"); scanf("%d",&val); if((rear+1)%max==front)

[{](https://e-next.in)

[printf("Queue is full."); return 0;](https://e-next.in)

[}](https://e-next.in)

[Else if(front==-1)](https://e-next.in)

[{](https://e-next.in)

[rear=front=0; q[front]=val; printf("Inserted successfully."); return val;](https://e-next.in)

[}](https://e-next.in)

[else](https://e-next.in)

{

front=(front-1+max)%max; q[front]=val; printf("Inserted successfully."); return val;

}}

intdeleteq\_front()

{ intval; if(front== -1)

{

printf("Queue is empty.");

//return -1;

[}](https://e-next.in)

[else if(front == rear)](https://e-next.in)

[{](https://e-next.in)

[intval=q[front]; front=rear= -1; printf("Deleted value : %d",val); return val;](https://e-next.in)

[}](https://e-next.in)

[else](https://e-next.in)

[{](https://e-next.in)

[val=q[front]; front=(front+1)%max;](https://e-next.in) printf("Deleted value : %d",val); return val;

}}

intdeleteq\_rear()

{ intval;

if(rear== -1)

{

printf("Queue is empty."); return -1;

}

else if(front == rear)

{

[intval=q[rear]; front=rear= -1; printf("Deleted value : %d",val); return val;](https://e-next.in)

[}](https://e-next.in)

[else](https://e-next.in)

[{](https://e-next.in)

[val=q[rear]; rear=(rear-1+max)%max; printf("Deleted value : %d",val); return val;](https://e-next.in)

[}}](https://e-next.in)

void display()

{

inti; if(front ==-1)

{

printf("Queue is empty.");

}

else

{

printf("Queue is :"); for(i=front;i!=rear;i=(i+1)%max)

{

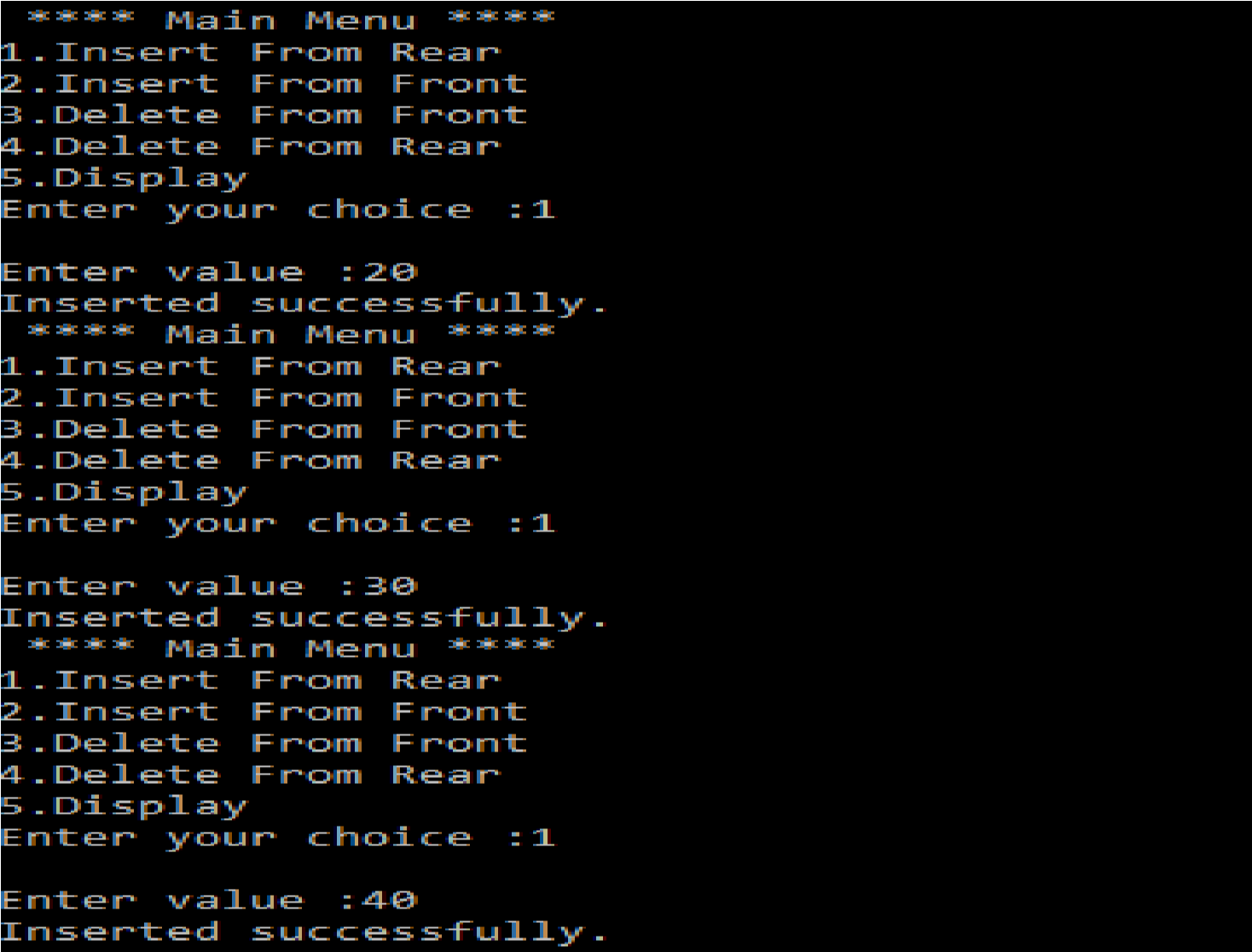
[printf(" %d ",q[i]);](https://e-next.in)

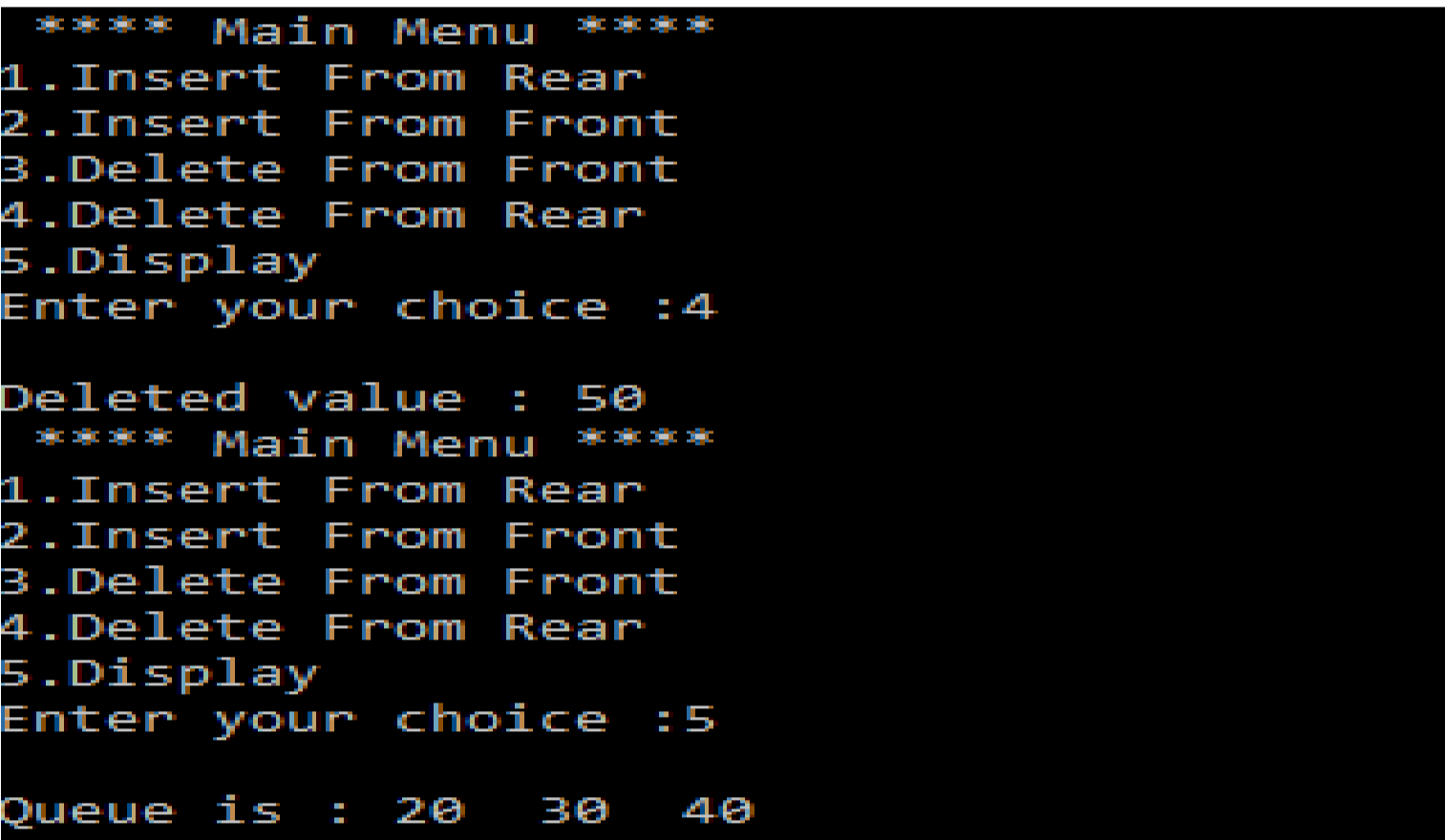
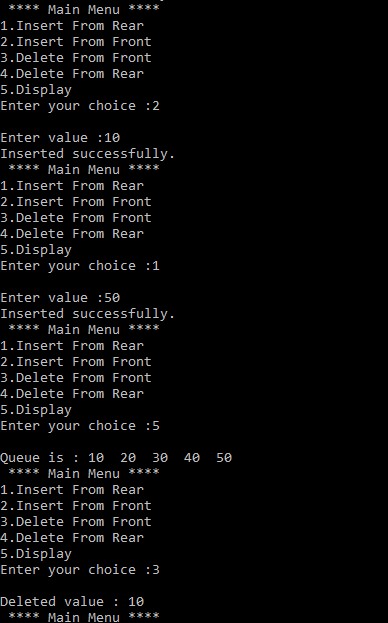
[}](https://e-next.in)

[printf(" %d ",q[i]);](https://e-next.in)

[}}](https://e-next.in)

[OUTPUT :](https://e-next.in)





PRACTICAL NO:-5(A) AIM : Write a program to implement bubble sort.

PROGRAM CODE:

#include<stdio.h> #include<conio.h>

[int size,val; void disp(int size); int sort(int size); int arr[20]; int main()](https://e-next.in)

[{ int i,ch;](https://e-next.in)

[printf("Enter the size of array : "); scanf("%d",&size); for(i=0;i<size;i++)](https://e-next.in)

[{](https://e-next.in)

[scanf("%d",&arr[i]);](https://e-next.in)

}

do

{

printf("\n\*\*\*\*Main Menu\*\*\*\*\n"); printf("1.Display\n"); printf("2.Sorting\n");

printf("Enter your Choice : "); scanf("%d",&ch); switch(ch)

{

case 1:disp(size); break; [case 2:sort(size); break;](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[while(ch!=2); getch (); return 0;](https://e-next.in)

[}](https://e-next.in)

[void disp(int size)](https://e-next.in)

[{ int i;](https://e-next.in)

[printf("Given Array :\n");](https://e-next.in) for(i=0;i<size;i++)

{

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

}

}

int sort(size)

{ int i,j;

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

{

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

{

[if(arr[j]>arr[j+1])](https://e-next.in)

[{](https://e-next.in)

[int temp; temp=arr[j]; arr[j]=arr[j+1]; arr[j+1]=temp;](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

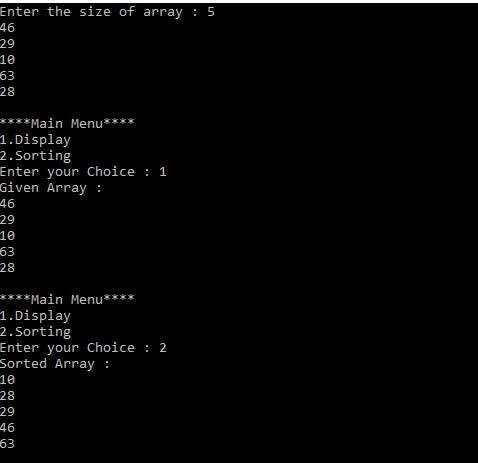
[}](https://e-next.in)

[printf("Sorted Array : \n"); for(i=0;i<size;i++)](https://e-next.in)

[{](https://e-next.in)

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

} } OUTPUT :



PRACTICAL NO:-5(B) AIM : Write a program to implement selection sort.

PROGRAM CODE:

#include<stdio.h>

#include<conio.h> #include<malloc.h> [int selection\_sort(int n); int A[20]; int selection\_sort(int n)](https://e-next.in)

[{](https://e-next.in)

[int imin,i,j,temp; for(i=0;i<n;i++)](https://e-next.in)

[{](https://e-next.in)

[imin=i; for(j=i+1;j<n;j++)](https://e-next.in)

[{](https://e-next.in)

[if(A[imin]>A[j])](https://e-next.in)

[{](https://e-next.in)

imin=j;

}

}

temp = A[i];

A[i] = A[imin];

A[imin] = temp;

}

printf("Successfully sorted using Selection sort :");

}

int main()

{ int n,i; [printf("Enter the size :"); scanf("%d",&n); printf("Enter the element :\n"); for(i=0;i<n;i++)](https://e-next.in)

[{](https://e-next.in)

[scanf("%d",&A[i]); printf("\n");](https://e-next.in)

[}](https://e-next.in)

[selection\_sort(n); for(i=0;i<n;i++)](https://e-next.in)

[{](https://e-next.in)

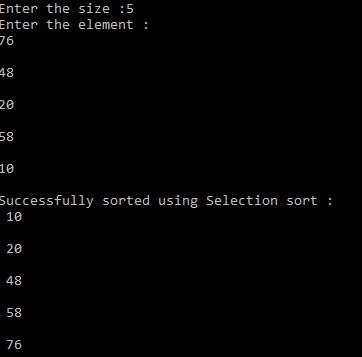
printf("\n %d\n",A[i]);

}

return 0;

}

OUTPUT :



PRACTICAL NO:-5(C) AIM : Write a program to implement insertionsort.

PROGRAM CODE:

#include<stdio.h> #include<conio.h> int A[10]; [void insertion\_sort(int n)](https://e-next.in)

[{](https://e-next.in)

[int val,vacant,i; for(i=1;i<n;i++)](https://e-next.in)

[{](https://e-next.in)

[val=A[i]; vacant=i;](https://e-next.in)

[while(A[vacant-1]>val && vacant!=0)](https://e-next.in)

[{](https://e-next.in)

[A[vacant]=A[vacant-1]; vacant=vacant - 1;](https://e-next.in)

[}](https://e-next.in)

A[vacant]=val;

}

printf("Successfully sorted using Insertion Sort Algorithm : \n");

}

void main()

{

int n,i;

printf("Enter the size of array : "); scanf("%d",&n); printf("Enter the elements :\n"); for(i=0;i<n;i++)

{

[scanf("%d",&A[i]); printf("\n");](https://e-next.in)

[}](https://e-next.in)

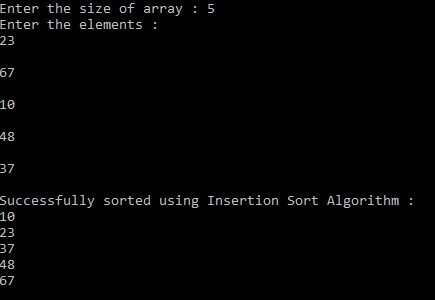
[insertion\_sort(n); for(i=0;i<n;i++)](https://e-next.in)

[{](https://e-next.in)

[printf("%d \n",A[i]);](https://e-next.in)

[}}](https://e-next.in)

[OUTPUT :](https://e-next.in)



PRACTICAL NO.:6(A) AIM : Write a program to implement merge sort.

PROGRAM CODE:

#include<stdio.h>

#include<conio.h> #define max 10

[int a[11] = { 10, 14, 19, 26, 27, 31, 33, 35, 42, 44, 0 }; int b[10];](https://e-next.in)

[void merging(int low, int mid, int high) {](https://e-next.in)

[int l1, l2, i;](https://e-next.in)

[for(l1 = low, l2 = mid + 1, i = low; l1 <= mid && l2 <= high; i++) { if(a[l1] <= a[l2]) b[i] = a[l1++]; else b[i] = a[l2++];](https://e-next.in)

[}](https://e-next.in)

while(l1 <= mid) b[i++] = a[l1++]; while(l2 <= high) b[i++] = a[l2++]; for(i = low; i <= high; i++) a[i] = b[i];

}

void sort(int low, int high) { int mid; if(low < high) {  [mid = (low + high) / 2; sort(low, mid); sort(mid+1, high); merging(low, mid, high);](https://e-next.in)

[} else { return;](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

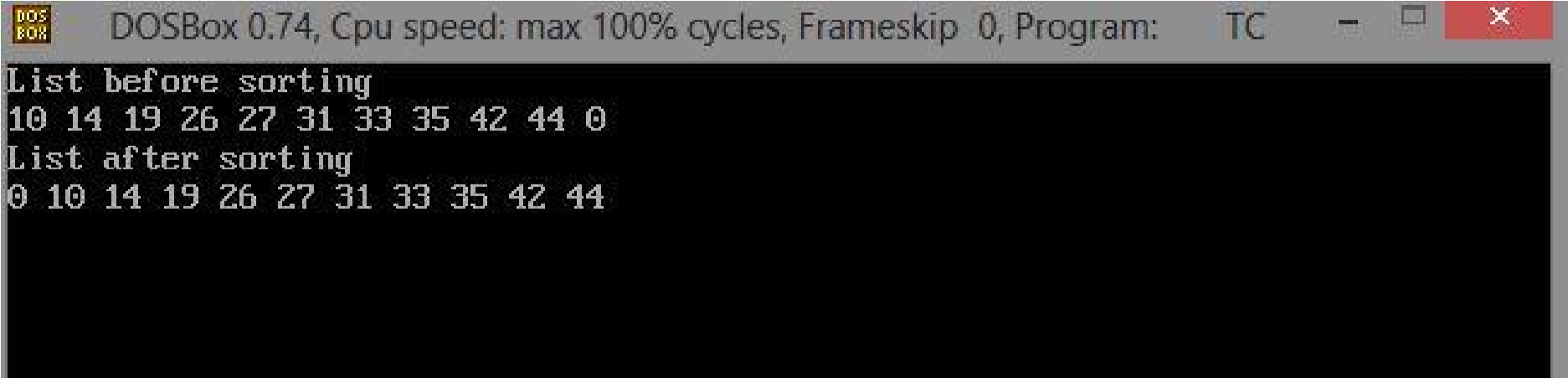
[int main() {](https://e-next.in)

[int i; clrscr(); printf("List before sorting\n");](https://e-next.in)

for(i = 0; i <= max; i++) printf("%d ", a[i]); sort(0, max); printf("\nList after sorting\n"); for(i = 0; i <= max; i++) printf("%d ", a[i]); getch();

}

OUTPUT :



PRACTICAL NO.:6(B)

AIM: Write a program to search the element using seqential search.

PROGRAM CODE:

#include<stdio.h> #include<conio.h> intsize,val; [void disp(int size); void search(intval,int size); intarr[20]; intmain()](https://e-next.in)

[{ inti,ch;](https://e-next.in)

[printf("Enter the size of array : "); scanf("%d",&size); for(i=0;i<size;i++)](https://e-next.in)

[{](https://e-next.in)

[scanf("%d",&arr[i]);](https://e-next.in)

[}](https://e-next.in)

do

{

printf("\n\*\*\*\*Main Menu\*\*\*\*\n"); printf("1.Display\n"); printf("2.Search\n"); printf("Enter your Choice : "); scanf("%d",&ch); switch(ch)

{

case 1:disp(size); break;

case 2:printf("Enter value to be search : "); [scanf("%d",&val); search(val,size); break;](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[while(ch!=2); getch (); return 0;](https://e-next.in)

[}](https://e-next.in)

[void search(intval,int size)](https://e-next.in)

[{ inti;](https://e-next.in) for(i=0;i<size;i++)

{

if(arr[i]==val)

{

printf("Value is found at %d position.",i); break;

}

}

if(i==size)

{

printf("Value is not found.");

}

[}](https://e-next.in)

[void disp(int size)](https://e-next.in)

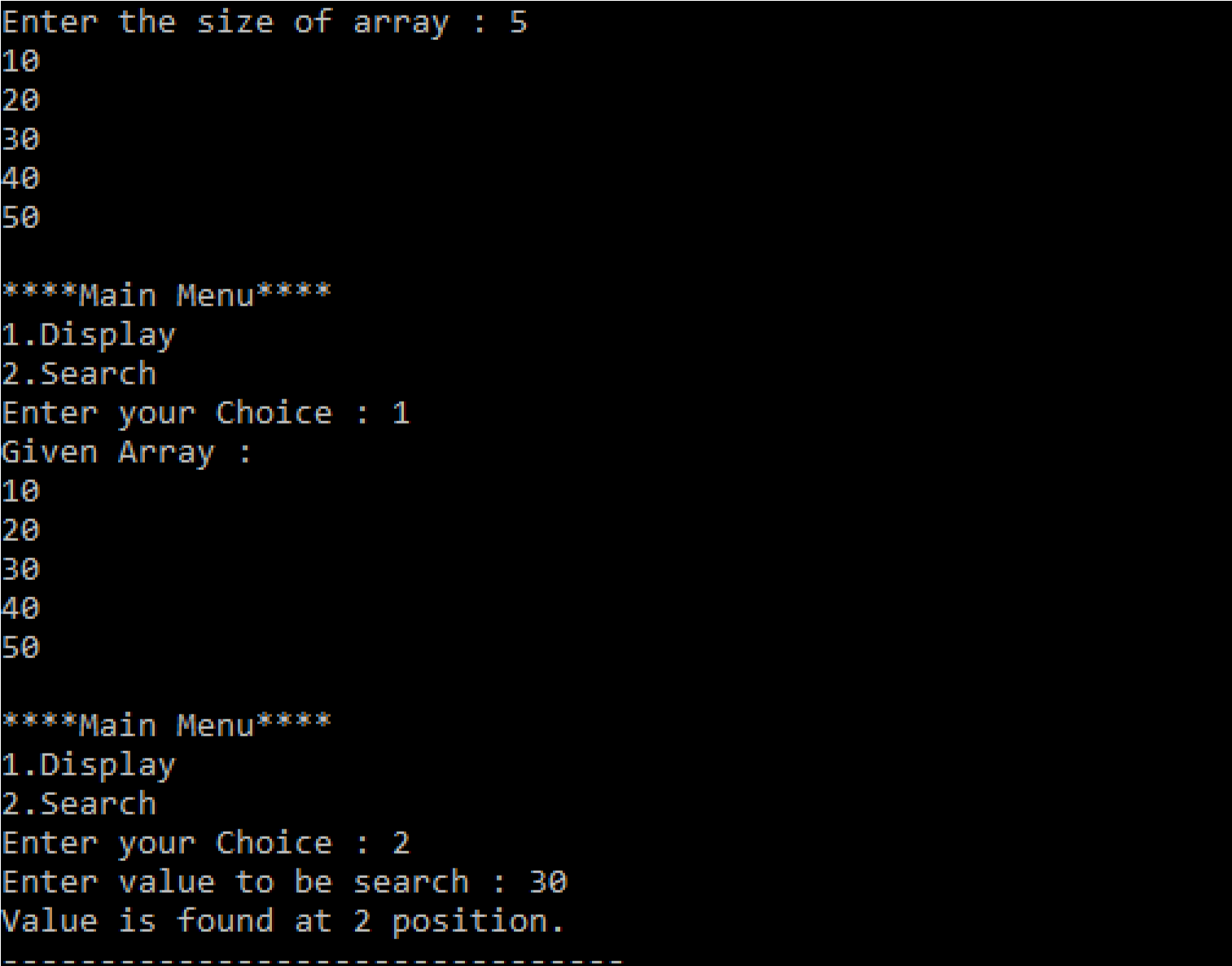
[{ inti;](https://e-next.in)

[printf("Given Array :\n"); for(i=0;i<size;i++)](https://e-next.in)

[{](https://e-next.in)

[printf("%d\n",arr[i]);](https://e-next.in)

[} }](https://e-next.in) OUTPUT :



PRACTICAL NO.:6(C)

AIM : Write a program to search the element using binary search.

PROGRAM CODE:

#include <stdio.h> intmain()

[{](https://e-next.in)

[int c, first, last, middle, n, search, array[100]; 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]); printf("Enter value to find\n"); scanf("%d", &search);](https://e-next.in)

[first = 0; last = n - 1; middle = (first+last)/2;](https://e-next.in) 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;

}

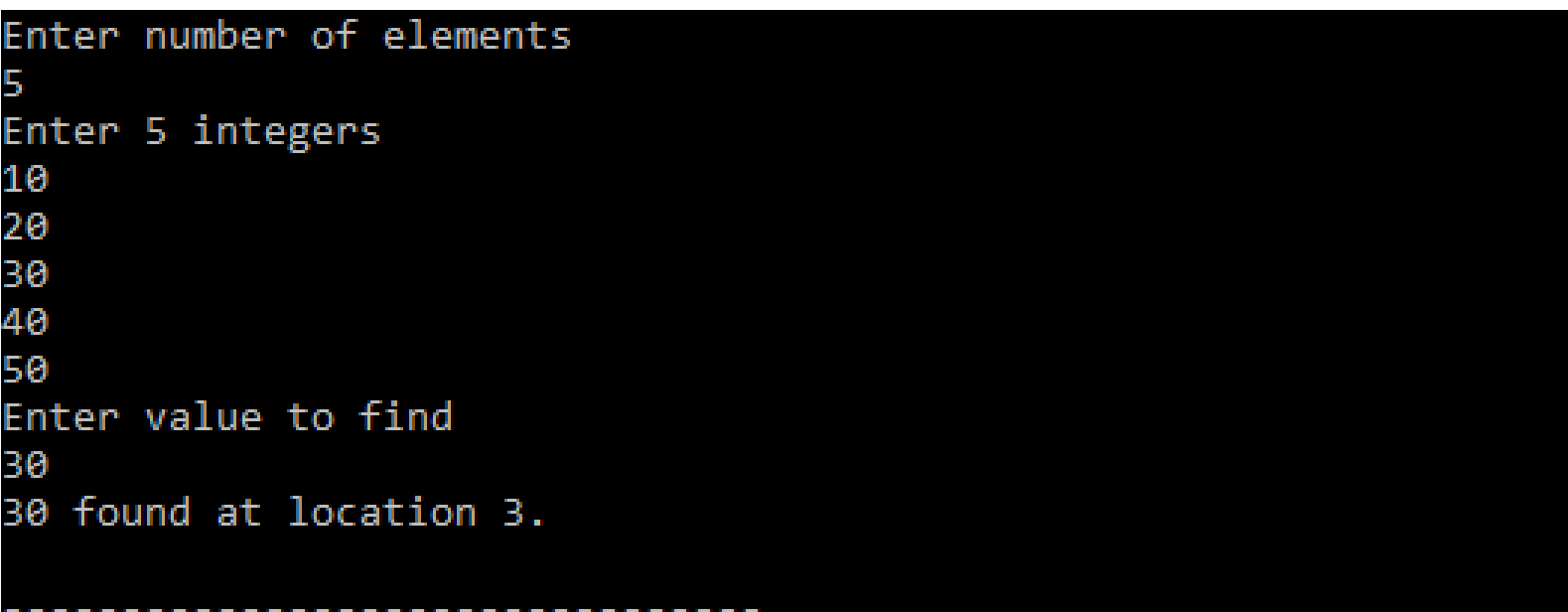
else if(array[middle]>search) last = middle - 1; middle = (first + last)/2;

[} if (first > last)](https://e-next.in)

[printf("Not found! %d is not present in the list.\n", search); return 0;](https://e-next.in)

[}](https://e-next.in)

[OUTPUT :](https://e-next.in)



PRACTICAL NO.:7

(Implement the following data structure techiniques)Aim 7(A): Write a program to create the tree and display the element.

Aim 7(B): Write a program to construct the binary tree.

Aim 7(C): Write a program for inorder,postorder and preorder traversal of tree.

[PROGRAM CODE:](https://e-next.in)

[#include<stdio.h> #include<malloc.h> struct node](https://e-next.in)

[{](https://e-next.in)

[int data; struct node \*left; struct node \*right;](https://e-next.in)

[};](https://e-next.in)

[struct node \*root=NULL;](https://e-next.in) struct node \*create(struct node\*); struct node \*display(struct node\*); void preorder(struct node \*temp); void postorder(struct node \*temp); void inorder(struct node \*temp); intmain()

{

intchoice,val,count,min,max; do

{

printf("\*\*\*\* Main Menu \*\*\*\n"); printf("1. create a binary search\n"); [printf("2. Display the tree \n"); printf("3. EXIT \n"); printf("Enter your choice:"); scanf("%d",&choice); printf("\n\n"); switch(choice)](https://e-next.in)

[{](https://e-next.in)

[case 1:root=create(root); break; case 2:root=display(root); break; case 3:break;](https://e-next.in)

}

}while(choice!=3); return 0;

}

struct node \*create(struct node \*root)

{

struct node \*newnode=NULL,\*temp=NULL,\*parent=NULL; intval;

printf("Enter the data or enter -1 to exit:"); scanf("%d",&val); while(val!=-1)

{

[newnode=(struct node\*)malloc(sizeof(struct node)); newnode->data=val; if(root==NULL)](https://e-next.in)

[{](https://e-next.in)

[root=newnode; newnode->left=NULL; newnode->right=NULL;](https://e-next.in)

[}](https://e-next.in)

[else](https://e-next.in)

[{](https://e-next.in)

[temp=root; while(temp!=NULL)](https://e-next.in)

{

parent=temp; if(val<temp->data)

{

temp=temp->left;

}

else

{

temp=temp->right;

}

}

if(val<parent->data)

[{](https://e-next.in)

[parent->left=newnode; newnode->left=NULL; newnode->right=NULL;](https://e-next.in)

[}](https://e-next.in)

[else](https://e-next.in)

[{](https://e-next.in)

[parent->right=newnode; newnode->left=NULL; newnode->right=NULL;](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

printf("Enter the data or enter -1 to exit:"); scanf("%d",&val);

}

printf("Succesfully created \n"); return root;

}

struct node \*display(struct node \*root)

{

int choice1; printf("\*\*\* Display Menu\*\*\*\n"); printf("1.pre-order\n"); printf("2.In-order\n"); [printf("3.post-order\n"); printf("4. EXIT\n"); printf("Enter your choice :"); scanf("%d",&choice1); switch(choice1)](https://e-next.in)

[{](https://e-next.in)

[case 1:printf("\tThe Pre-order Traveral is:"); preorder(root); break;](https://e-next.in)

[case 2:printf("\tThe in order traversal is:"); inorder(root); break;](https://e-next.in)

case 3:printf("\tThe post-order traversal is:"); postorder(root); break; case 4:break;

}

printf("\n");

return root;

}

void preorder(struct node \*temp)

{

if(temp!=NULL)

{

[printf("%d",temp->data); preorder(temp->left); preorder(temp->right);](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[void postorder(struct node \*temp)](https://e-next.in)

[{](https://e-next.in)

[if(temp!=NULL)](https://e-next.in)

[{](https://e-next.in)

[postorder(temp->left); postorder(temp->right); printf("%d",temp->data);](https://e-next.in)

}

}

void inorder(struct node \*temp)

{

if(temp!=NULL)

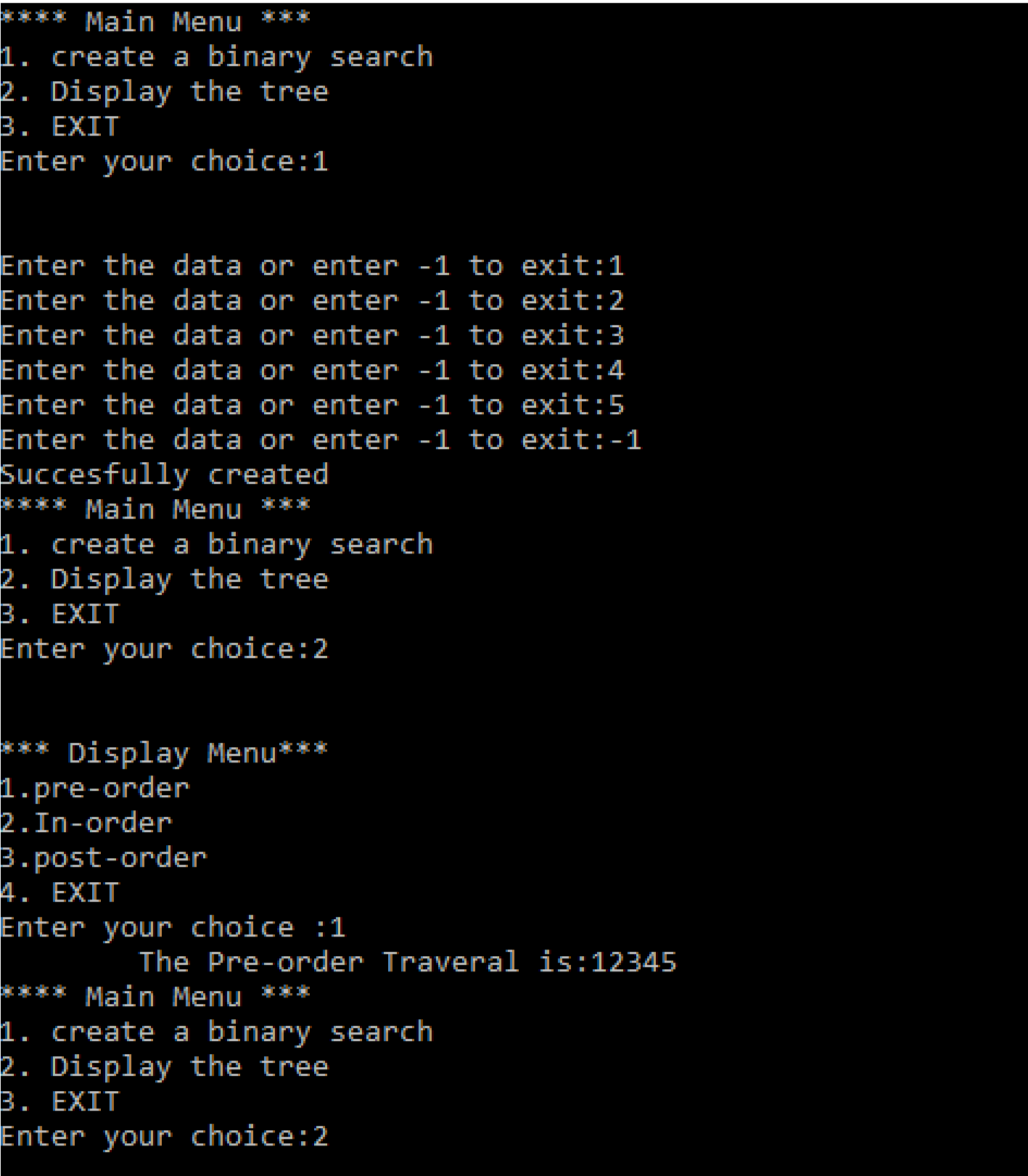
{

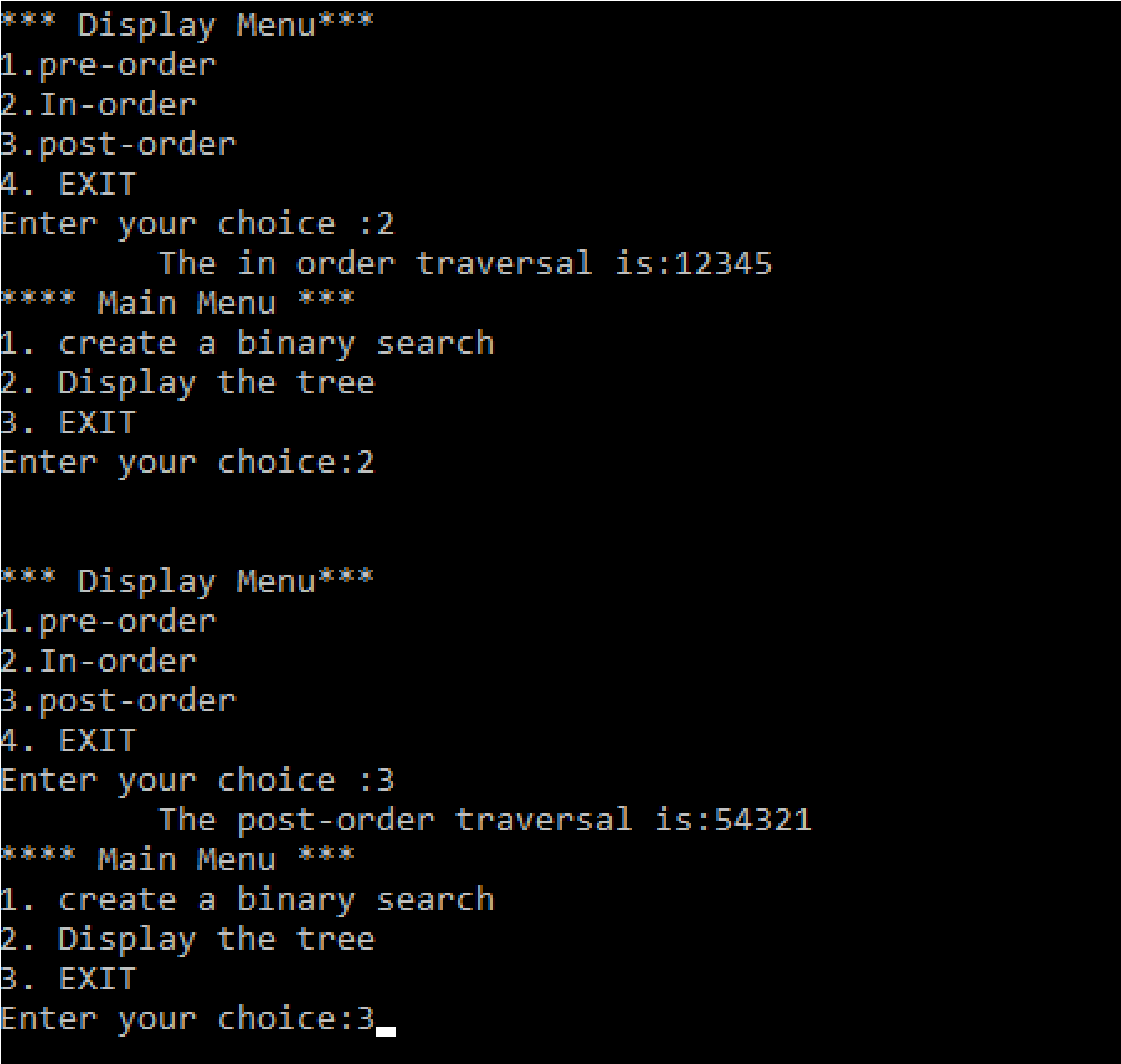
inorder(temp->left); printf("%d",temp->data); inorder(temp->right);

}

}

[OUTPUT :](https://e-next.in)





PRACTICAL NO.:8(A)

AIM : Write a program to insert the element into maximum heap.

PROGRAM CODE:

[#include<stdio.h>](https://e-next.in)

[#include<string.h> #define SIZE 30 int a[SIZE],n;](https://e-next.in)

[void maxheapify(int a[],int i,int n1); void buildheap(int a[],int n1); void heap\_sort(int a[]); void swap(int i,int j); int length(int a[]); int main()](https://e-next.in)

[{](https://e-next.in) int i,j;

printf("Enter the number of element:"); scanf("%d",&n); for(i=0;i<=n;i++)

{

printf("Enter a value:");

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

}

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

{

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

[}](https://e-next.in)

[printf("\n");](https://e-next.in)

[}](https://e-next.in)

[void buildheap(int a[],int n1)](https://e-next.in)

[{ int i,j;](https://e-next.in)

[for(i=(n1/2)-1;i>=0;i--)](https://e-next.in)

[{](https://e-next.in)

[maxheapify(a,i,n1);](https://e-next.in)

[}](https://e-next.in)

[for(j=0;j<n1;j++)](https://e-next.in)

{

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

}

printf("\n\n");

}

void maxheapify(int a[],int i,int n1) { int max,l,r; max=i; l=2\*i+1; r=2\*i+2; if(l<n1&&r<n1)

[{](https://e-next.in)

[if(a[l]>a[max])](https://e-next.in)

[{](https://e-next.in)

[max=l;](https://e-next.in)

[}](https://e-next.in)

[if(a[r]>a[max])](https://e-next.in)

[{](https://e-next.in)

[max=r;](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[else if(l<n1&&r>=n1)](https://e-next.in)

[{](https://e-next.in)

if(a[l]>a[max])

{

max=l;

}

}

else if(l>=n1&&r<n1) { if(a[r]>a[max])

{

max=r;

}

}

[if(i!=max)](https://e-next.in)

[{](https://e-next.in)

[swap(i,max); maxheapify(a,max,n1);](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[void swap(int i,int j)](https://e-next.in)

[{](https://e-next.in)

[int temp=a[i]; a[i]=a[j]; a[j]=temp;](https://e-next.in)

[}](https://e-next.in)

int length(int a[])

{

int i=0; while(a[i]!='\0')

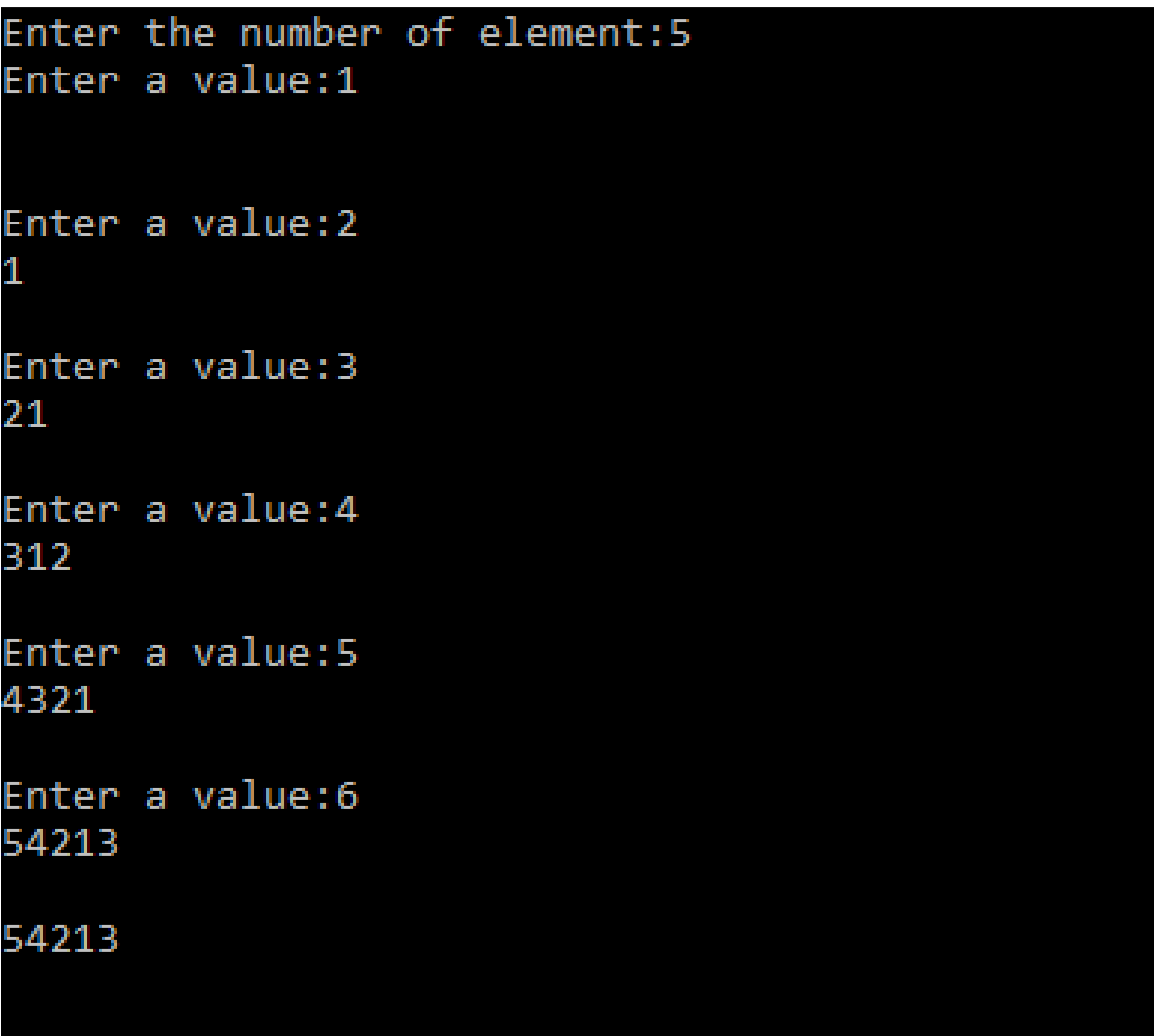
{

i++;

} return i;

}

OUTPUT :



PRACTICAL NO.:8(B)

AIM : Write a program to insert the element into minimum heap.

PROGRAM CODE:

#include<stdio.h>

#include<string.h> [#define SIZE 30 int a[SIZE],n;](https://e-next.in)

[void maxheapify(int a[],inti,int n1); void buildheap(int a[],int n1); void heap\_sort(int a[]); void swap(inti,int j); intlength(int a[]); intmain()](https://e-next.in)

[{ inti,j;](https://e-next.in)

[printf("Enter the number of element:"); scanf("%d",&n);](https://e-next.in) for(i=0;i<=n;i++)

{

printf("Enter a value:"); scanf("%d",&a[i]); buildheap(a,i);

}

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

{

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

}

printf("\n");

}

[void buildheap(int a[],int n1)](https://e-next.in)

[{ inti,j;](https://e-next.in)

[for(i=(n1/2)-1;i>=0;i--)](https://e-next.in)

[{](https://e-next.in)

[maxheapify(a,i,n1);](https://e-next.in)

[}](https://e-next.in)

[for(j=0;j<n1;j++)](https://e-next.in)

[{](https://e-next.in)

[printf("%d",a[j]);](https://e-next.in)

[}](https://e-next.in)

[printf("\n");](https://e-next.in)

}

void maxheapify(int a[],inti,int n1)

{

intmin,l,r; min=i; l=2\*i+1; r=2\*i+2; if(l<n1&&r<n1)

{

if(a[l]<a[min])

{

min=l;

[}](https://e-next.in)

[if(a[r]<a[min])](https://e-next.in)

[{](https://e-next.in)

[min=r;](https://e-next.in)

[}](https://e-next.in)

[}](https://e-next.in)

[else if(l<n1&&r>=n1)](https://e-next.in)

[{](https://e-next.in)

[if(a[l]<a[min])](https://e-next.in)

[{](https://e-next.in)

[min=l;](https://e-next.in)

[}](https://e-next.in)

}

else if(l>=n1&&r<n1)

{

if(a[r]<a[min])

{

min=r;

}

}

if(i!=min)

{

swap(i,min); maxheapify(a,min,n1);

[}](https://e-next.in)

[}](https://e-next.in)

[void swap(inti,int j)](https://e-next.in)

[{](https://e-next.in)

[int temp=a[i]; a[i]=a[j]; a[j]=temp;](https://e-next.in)

[}](https://e-next.in)

[intlength(int a[])](https://e-next.in)

[{](https://e-next.in)

[inti=0; while(a[i]!='\0')](https://e-next.in)

{

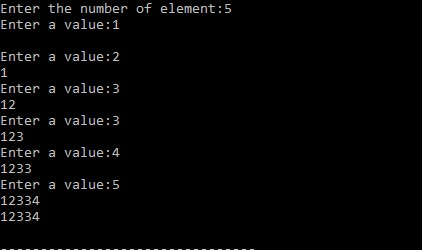
i++;

}

return i;

}

OUTPUT :



v[11:44 pm, 21/10/2024] Harshil:9.1

practical no 9. q.1)Write a program to implement the collision technique.

q.2)Write a program to implement the concept the linear probing.

program to implement hash search using linear probin\*/

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

int hashSearch(int a[],int x,int n)

{

int index,start;

index=x%n;

if(a[index]==x)

return index;

else

if(a[index]==-1)

return -1;

else

{

start=index;

do

{

index=(index+1)%n;

if(a[index]==x)

return index;

else

if(a[index]==-1)

break;

}while(index!=start);

return -1;

}

}

void main()

{

int hsh[10],i,x,index,k;

clrscr();

cout<<"hashing using linear probing\n";

cout<<"hash table creation\n\n";

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

hsh[i]=-1;

for(i=1;i<=7;i++)

{

cout<<"\n ENter a number";

cin>>x;

index=x%10;

while(hsh[index]!=-1)

index=(index+1)%10;

hsh[index]=x;

}

clrscr();

cout<<"\n hash table\n";

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

cout<<setw(4)<<i;

cout<<"\n";

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

cout<<setw(4)<<hsh[i];

do

{

cout<<"\n Elments tobe searched to stop enter 1\n";

cin>>x;

if(x>=0)

{

k=hashSearch(hsh,x,10);

if(k>=0)

cout<<x<<"is prsent at hash["<<k<<"]\n";

else

cout<<x<<"is not present\n";

}

}while(x>=0);

getch();

}

[11:45 pm, 21/10/2024] Harshil: 10.1

practical 10.1 Write a program to generate the adjacency matrix\*/

#include<iostream.h>

#include<conio.h>

void main()

{

clrscr();

int ad[10][10],i,j,m,n,a,b;

cout<<"\n Enter no. of nodes";

cin>>n;

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

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

ad[i][j]=0;

cout<<"\n Enter the number of edges:";

cin>>m;

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

{

cout<<"\n enter edge between :";

cin>>a>>b;

ad[a][b]=1;

}

cout<<"\n adjanceny matrix:\n";

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

{

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

cout<<ad[i][j]<<" ";

cout<<"\n";

}

getch();

}

[11:47 pm, 21/10/2024] Harshil: **10.2**

/\* practica 10.2 write a program for shortest path diagram\*/

#include<iostream.h>

#include<iomanip.h>

#include<conio.h>

#include<stdlib.h>

#define MAX 10

#define TEMP 0

#define PERM 1

#define infinity 9999

class node

{

public:

int predecessor;

int dist;

int status;

};

int adj[MAX][MAX];

int n;

void create\_graph()

{

int i,maxedge,origin,destin,wt;

cout<<"\n ENtre number of vertices:";

cin>>n;

maxedge=n\*(n-1);

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

{

cout<<"Enter edge "<<i<<"(0 0 to quit):";

cin>>origin>>destin;

if((origin==0)&&(destin==0))

break;

cout<<"Enter weight fro this edge:";

cin>>wt;

if(origin >n || destin>n || origin<=0 ||destin<=0)

{

cout<<"invalid edge\n";

i--;

}

else

adj[origin][destin]=wt;

}

}

void display()

{

int i,j;

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

{

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

cout<<setw(3)<<adj[i][j];

cout<<"\n";

}

}

int findpath(int s,int d,int path[MAX],int \*sdist)

{

node state[MAX];

int i,k,min,count=0,current,newdist,u,v;

\*sdist=0;

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

{

state[i].predecessor=0;

state[i].dist=infinity;

state[i].status=TEMP;

}

state[s].predecessor=0;

state[s].dist=0;

state[s].status=PERM;

current=s;

while(current!=d)

{

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

{

if(adj[current][i]> 0 && state[i].status==TEMP)

{

newdist=state[current].dist=current;

if(newdist<state[i].dist)

{

state[i].predecessor=current;

state[i].dist=newdist;

}

}

}

min=infinity;

current=0;

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

{

if(state[i].status==TEMP && state[i].dist <min)

{

min=state[i].dist;

current=i;

}

}

if(current==0)

return 0;

state[current].status=PERM;

}

while(current!=0)

{

count++;

path[count]=current;

current=state[current].predecessor;

}

for(k=count;k>1;k--)

{

u=path[k];

v=path[k-1];

\*sdist+=adj[u][v];

}

return(count);

}

void main()

{

int i,j;

int source,dest;

int path[MAX];

int shortdist,count;

clrscr();

create\_graph();

cout<<"\nthe adjency matrix is:\n";

display();

while(1)

{

cout<<"Enter source node(0 to quit):";

cin>>source;

cout<<"Enter destination node(0 to quit):";

cin>>dest;

if(source==0 || dest==0)

exit(1);

count=findpath(source,dest,path,&shortdist);

if(shortdist!=0)

{

cout<<"Shortest distance is"<<shortdist<<"\n";

cout<<"shortest path is";

for(i=count;i>1;i--)

cout<<path[i]<<"->";

cout<<path[i];

cout<<"\n";

}

else

cout<<"there is no path from source to destination nodes\n";

}

getch();

}

end