**PRACTICAL FILE**

**of**

**data structure using ‘c’ LAB**

****

Submitted to: Submitted by:

Mrs. Heena Manjyot Singh

Astt. Prof. 141071

CSE/IT Deptt. 3rd sem. – CSE

**GLOBAL INSTITUTE OF TECHNOLOGY AND MANAGEMENT, GURGAON**

Manjyot Singh

141071

Program No. 1

To search an element in two-dimensional array using Linear Search

#include<stdio.h>

#include<conio.h>

void main()

{

int a[10],I,n,k,data;

printf(“enter the no. of elements\n”);

scanf(“%d”, &n);

printf(“enter elements\n”);

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

{

Scanf(“%d”,&a[i]);

}

printf(“enter element to be searched\n”,);

scanf(“%k”,&k);

/\* linear search conditions\*/

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

{

if(k==a[i])

{

data=1;

break;

}

}

if(data==1)

{

printf(“elements is present in array\n”);

printf(“positon of element is:- %d”,&i);

}

else

printf(“element is not present in array\n”);

getch();

}

Manjyot Singh

141071

Program No. 2

To search an element in two-dimensional array using Binary Search

#include<stdio.h>

#include<conio.h>

void main()

{

int I,first,last,middle,n,search,array[20];

printf(“enter number of elements\n”);

scanf(%d”,&n);

printf(“enter elements %d \n”,n);

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

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(“element found at location %d \n”,search,middle+1);

break;

}

else

{

last=middle-1;

middle=(first+last)/2;

}

if(first>last)

printf(“not found! Element is notpresentin the list\n”,search);

getch();

}

Manjyot Singh

141071

Program no.3

To sort the elements for exchange sorting

#include<stdio.h>

#include<conio.h>

void main()

{

int array[10],n,i,j,temp;

clrscr();

printf("enter no. of elements:");

scanf("%d",&n);

printf("enter the elements:");

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

{

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

}

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

{

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

{

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

{

temp=array[i];

array[i]=array[j];

array[j]=temp;

}

}

}

printf("sorted array:");

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

{

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

}

getch();

}

Manjyot Singh

141071

Program No. 4

To sort the elements of Merge Sort

#include<stdio.h>

#include<conio.h>

void main()

{

int a[10],b[10],c[20],i,j,k,m,n;

clrscr();

printf("enetr the no. of elements of first orderd list:");

scanf("%d",&n);

printf("enter the first list:");

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

{

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

}

printf("enter the no. of second orded list:");

scanf("%d",&m);

printf("enter the elements of second list");

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

{

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

}

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

i=0;

j=0;

k=0;

while(i<n&&j<m)

{

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

{

c[k]=a[i];

k=k+1;

i=i+1;

}

else if(a[i]>b[j])

{

c[k]=b[j];

k=k+1;

j=j+1;

}

else if(a[i]==b[j])

{

c[k]=a[i];

k=k+1;

i=i+1;

c[k]=b[j];

k=k+1;

j=j+1;

}

}

while(i<n)

{

c[k]=a[i];

k=k+1;

i=i+1;

}

while(j<m)

{

c[k]=b[j];

k=k+1;

j=j+1;

}

printf("sorted list:");

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

{

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

}

getch();

}

Manjyot Singh

141071

Program No. 5

To sort the elements by straight selection sort

#include<stdio.h>

#include<conio.h>

void main()

{

int a[10],n,i,j,p, small;

clrscr();

printf("enter no. of elements:");

scanf("%d",&n);

printf("enter the elements:");

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

{

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

}

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

{

small=a[i];

p=i;

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

{

if(a[j]<small)

{

small=a[j];

p=j;

}

}

a[p]=a[i];

a[i]+small;

}

printf("sorted array:");

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

{

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

}

getch();

}

Manjyot Singh

141071

Program No. 6

To sort the elements by Bubble Sort

#include<stdio.h>

#include<conio.h>

void main()

{

int a[10],n,i,j,temp;

clrscr();

printf("enter no. of elements:");

scanf("%d",&n);

printf("enter the elements:");

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

{

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

}

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

{

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

{

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

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

printf("sorted array:");

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

{

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

}

getch();

}

Manjyot Singh

141071

Program No. 7

To insert elements in Linked List

#include <stdio.h>

#include <conio.h>

#include <malloc.h>

struct node

{

int info;

struct node \*next;

};

typedef struct node NODE;

NODE \*start;

void createmptylist(NODE \*\*start)

{

\*start=(NODE \*)NULL;

}

void traversinorder(NODE \*start)

{

while (start != (NODE \*) NULL)

{

printf("%d\n",start->info);

start=start->next;

}

}

void insertatbegin( int item)

{

NODE \*ptr;

ptr=(NODE \*)malloc( sizeof (NODE));

ptr->info=item;

if (start==(NODE \*)NULL)

0ptr->next=(NODE \*)NULL;

else

ptr->next=start;

start=ptr;

}

void insert\_at\_end( int item)

{

NODE \*ptr,\*loc;

ptr=(NODE \*)malloc( sizeof (NODE));

ptr->info=item;

ptr->next=(NODE \*)NULL;

if (start==(NODE\*)NULL)

start=ptr;

else

{

loc=start;

while (loc->next!=(NODE \*)NULL)

loc=loc->next;

loc->next=ptr;

}

}

void insert\_spe(NODE \*start, int item)

{

NODE \*ptr,\*loc;

int temp,k;

for (k=0,loc=start;k<temp;k++)

{

loc=loc->next;

if (loc==NULL)

{

printf("node in the list at less than one\n");

return ;

}

}

ptr=(NODE \*)malloc( sizeof (NODE));

ptr->info=item;

ptr->next=loc->next;;

loc->next=ptr;

}

void main()

{

int choice,item,after;

char ch;

clrscr();

createmptylist(start);

do

{

printf("1.Insert element at begin \n");

printf("2. insert element at end positon\n");

printf("3. insert specific the position\n");

printf("4.travers the list in order\n");

printf("5. exit\n");

printf("enter your choice\n");

scanf("%d",&choice);

switch (choice)

{

Manjyot Singh

141071

Program No. : 8

To implement Stack usingArray

#include <stdio.h>

#include <conio.h>

#define MAXSIZE 10

void push();

int pop();

void traverse();

int stack[MAXSIZE];

int Top=-1;

void main()

{

int choice;

char ch;

do

{

clrscr();

printf("\n1. PUSH ");

printf("\n2. POP ");

printf("\n3. TRAVERSE ");

printf("\nEnter your choice");

scanf("%d",&choice);

switch (choice)

{

case 1: push();

break ;

case 2: printf("\nThe deleted element is %d",pop());

break ;

case 3: traverse();

break ;

default : printf("\nYou Entered Wrong Choice");

}

printf("\nDo You Wish To Continue (Y/N)");

fflush(stdin);

scanf("%c",&ch);

}

while (ch=='Y' || ch=='y');

}

void push()

{

int item;

if (Top == MAXSIZE - 1)

{

printf("\nThe Stack Is Full");

getch();

exit(0);

}

else

{

printf("Enter the element to be inserted");

scanf("%d",&item);

Top= Top+1;

stack[Top] = item;

}

}

int pop()

{

int item;

if (Top == -1)

{

printf("The stack is Empty");

getch();

exit(0);

}

else

{

item = stack[Top];

Top = Top-1;

}

return (item);

}

void traverse()

{

int i;

if (Top == -1)

{

printf("The Stack is Empty");

getch();

exit(0);

}

else

{

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

{

printf("Traverse the element");

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

}

}

}

Manjyot singh

141071

Program No. : 9

To implement Queue in an array

#include<stdio.h>

#include<conio.h>

#define size 10

void enqueue(int);

void dequeue();

void display();

int queue[size], front=-1, rear=-1;

void main()

{

int value, choice;

clrscr();

while(1)

{

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

printf("1.insertion\n2.deletion\n3.display\n.exit");

printf("\nenter your choice");

scanf("%d",&choice);

switch(choice)

{

case 1:

printf("enter the value to be insert");

scanf("%d",&value);

enqueue(value);

break;

case 2:

dequeue();

break;

case 3:

display();

break;

case 4:

exit(0);

default:

printf("\nwrong selection");

}}}

void enqueue(int value)

{

if(rear==size-1)

{

printf("\nqueue is full");

}

else if(front==-1)

{

front=0;

rear++;

queue[rear]=value;

printf("\ninsertion success!!!");

}

}

void dequeue()

{

if(front==rear)

printf("\nqueue is empty");

else{

printf("\ndeleted:%d",queue[front]);

front++;

if(front==rear)

front=rear=-1;

}

}

void display()

{

if(rear==-1)

printf("\nqueue is empty");

else{

int i;

printf("\n queue elements are\n");

for(i=front; i<=rear; i++)

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

}

getch();

}

Manjyot Singh

141071

Program No. : 10

To implement Tree Traversing

#include<stdio.h>

#include<conio.h>

#include<alloc.h>

struct node

{

int num;

struct node \*left,\*right;

}\*root=NULL;

struct node \*insert(struct node \*,int);

void preorder(struct node \*);

void postorder(struct node \*);

void inorder(struct node \*);

int count=1;

void main()

{

int ch, n;

char ch1;

clrscr();

do

{

printf("\n 1. Create");

printf("\n 2. inorder");

printf("\n 3. preorder");

printf("\n 4. postorder");

printf("\n 5. exit");

printf("\n enter ur choice");

scanf("%d",&ch);

switch(ch)

{

case 1 : printf("\n Enter the value which you want to insert");

scanf("%d",&n);

root=insert(root,n);

break;

case 2 : printf("\n the inorder traversal is");

inorder (root);

break;

case 3 : preorder (root);

break;

case 4 : postorder (root);

break;

case 5 : exit(1);

}

printf("\n Do you want to continue (Y/N)");

fflush(stdin);

scanf("%c",&ch1);

}while(ch1 == 'y' ||ch1 =='Y');

getch();

}

struct node \*insert(struct node \*p,int digit)

{

if(p==NULL)

{

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

p->left = p->right = NULL;

p->num = digit;

++count;

}

else if(count%2==0)

{

p->left = insert(p->left,digit);

}

else

{

p->right = insert(p->right,digit);

}

return(p);

}

void preorder(struct node\*p)

{

if(p!=NULL)

{

printf("\n%d",p->num);

preorder(p->left);

preorder(p->right);

}

}

void inorder(struct node\*p)

{

if(p!=NULL)

{

inorder(p->left);

printf("\n%d",p->num);

inorder(p->right);

}

}

void postorder(struct node\*p)

{

if(p!=NULL)

{

postorder(p->left);

postorder(p->right);

printf("\n%d",p->num);

}

}