**Program1: Sparese Matrix**

#include<stdio.h>

#include<conio.h>

int main(){

int a[10][10], s[10][3], i, j, k=1, r, c, z=0, nz=0;

//taking input from the user and printing it

printf("Enter the number of rows and columns:\n");

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

printf("Enter the elements to the matrix\n");

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

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

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

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

z++;

else

nz++;

}

}

printf("You've entered:\n");

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

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

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

}

printf("\n");

}

if(nz>z){

printf("Not a sparse matrix");

}

else{

//computing the sparse representation

printf("It is a sparse matrix");

s[0][0]=r;

s[0][1]=c;

s[0][2]=nz;

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

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

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

s[k][0]=i;

s[k][1]=j;

s[k][2]=a[i][j];

k++;

}

}

}

printf("\nThe sparse matrix representation is given below:\n");

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

for(j=0; j<3; j++){

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

}

printf("\n");

}

//computing the transpose representation

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

s[i][0] = s[i][1] + s[i][0];

s[i][1] = s[i][0] - s[i][1];

s[i][0] = s[i][0] - s[i][1];

}

printf("\nThe representation of transpose of the sparse matrix is given below:\n");

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

for(j=0; j<3; j++){

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

}

printf("\n");

}

}

return 0;

}

**Program 2 : INSERTION SORT**

#include<stdio.h>

#include<conio.h>

void main()

{

int i, j, n, temp, A[25];

printf("Enter the value of n: ");

scanf("%d",&n);

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

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

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

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

{

temp=A[i];

j=i-1;

while((temp<A[j])&&(j>=0))

{

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

j=j-1;

}

A[j+1]=temp;

}

printf("Order of Sorted elements: ");

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

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

}

**Program 3 : Shell SORT**

/\*#include<stdio.h>

int main()

{

int i, j, n, temp, a[25];

printf("Enter the array limit: ");

scanf("%d",&n);

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

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

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

printf("Array before sorting: ");

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

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

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

{

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

{ if(a[j]>a[j+i])

{temp=a[j];

a[j]=a[j+i];

a[j+i]=temp;

}

}

}

printf("\n Sorted array: ");

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

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

return 0;

}

**Program 4 : Quick sort**

#include<conio.h>

#include<stdio.h>

int partition(int arr[],int p,int r);

void quicksort(int arr[],int p,int r);

void swap(int\* a,int\*b);

void printarray(int arr[],int size);

void main()

{ int arr[20],i,n,j;

clrscr();

printf("Enter number of elements : ");

scanf("%d",&n);

printf("\nEnter the elements : ");

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

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

}

quicksort(arr,0,n-1);

printarray(arr,n);

getch();

}

void quicksort(int arr[],int p,int r)

{ int q;

if(p<r)

{ q=partition(arr,p,r);

quicksort(arr,p,q-1);

quicksort(arr,q+1,r);

}

}

int partition(int arr[],int p,int r)

{ int x,i,j,temp;

x=arr[r];

i=p-1;

for(j=p;j<=(r-1);j++)

{ if(arr[j]<=x)

{ i++;

swap(&arr[i],&arr[j]);

}

}

swap(&arr[r],&arr[i+1]);

return (i+1);

}

void swap(int\* a,int\* b)

{ int temp;

temp=\*a;

\*a=\*b;

\*b=temp;

}

void printarray(int arr[],int size)

{ int i;

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

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

}

}

**Program 5: STACK OPERATIONS**

//Implementation of Stack

#include<stdio.h>

#include<conio.h>

int a[10],top=-1,ele,ss=10;

//function to add an element

void push()

{

if(top!=ss-1)

{

printf("\nEnter the element to be added:");

scanf("%d",&ele);

top++;

a[top]=ele;

}

else

printf("\nStack Overflow");

}

//function to delete an element

void pop()

{

if(top!=-1)

{

printf("\n%d is deleted",a[top]);

top--;

}

else

printf("\nStack Underflow");

}

//function to display elements of stack

void display()

{

int i;

if(top!=-1)

{

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

{

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

}

}

else

printf("\nStack Underflow");

}

void main()

{

int choice,chance=1;

while(chance)

{

printf("\nMenu Of Operations:");

printf("\n1.Push\n2.Pop\n3.Display");

printf("\nEnter choice:");

scanf("%d",&choice);

//Menu of operations

switch(choice)

{

case 1:push();

break;

case 2:pop();

break;

case 3:display();

break;

default:printf("\nWrong Choice!");

}

//Condition to continue

printf("\n1.Continue\n0.Exit");

printf("\nEnter your choice:");

scanf("%d",&chance);

}

getch();

}

**Program 6: INFIX TO POST FIX**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<ctype.h>

char s[30],pos[30],top=-1,in[30];

int j=0,len,i;

void push(char ch)

{

top++;

s[top]=ch;

}

char pop()

{

char ch;

ch=s[top];

top--;

return ch;

}

int prec(char ch)

{

if(ch=='+'||ch=='-')

{

return 1;

}

else if(ch=='\*'||ch=='/')

{

return 2;

}

else

{

return 0;

}

}

void main()

{

char ch;

clrscr();

printf("\n INPUT THE INFIX EXPRESSION \n");

gets(in);

len=strlen(in);

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

{

ch=in[i];

if(ch=='(')

{

push(ch);

}

else if(isalpha(ch))

{

pos[j]=ch;

j++;

}

else if(ch==')')

{

while(s[top]!='(')

{

pos[j]=pop();

j++;

}

pop();

}

else

{

while(prec(s[top])>=prec(ch))

{

pos[j]=pop();

j++;

}

push(ch);

}

}

while(top!=-1)

{

pos[j]=pop();

j++;

}

pos[j]='\0';

puts(pos);

getch();

}

**Program 7: POST FIX EVALUATION**

#include<stdio.h>

#include<conio.h>

#include<ctype.h>

#include<string.h>

float s[30];

int top=-1;

void push(float a)

{

top++;

s[top]=a;

}

float pop()

{

float a;

a=s[top];

top--;

return a;

}

void main()

{

char pos[30];

char ch;

float op1,op2,len,res,temp,i;

clrscr();

printf("ENTER THE POSTFIX EXPRESSION\n");

gets(pos);

len=strlen(pos);

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

{

ch=pos[i];

if(isalpha(ch))

{ printf("Enter the value of the operand:");

scanf("%f",&temp);

push(temp);

}

else

{

op2=pop();

op1=pop();

if(ch=='+')

push(op1+op2);

else if(ch=='-')

push(op1-op2);

else if(ch=='\*')

push(op1\*op2);

else

push(op1/op2);

}

}

res=pop();

printf("The evaluated expression is %f :",res);

}

**Program 8: QUEUE**

#include <stdio.h>

#include<stdlib.h>

#define MAX 50

void insert();

void delete();

void display();

int queue\_array[MAX];

int rear = - 1;

int front = - 1;

int main()

{

int choice;

printf("1.Insert element to queue \n");

printf("2.Delete element from queue \n");

printf("3.Display all elements of queue\n");

printf("4.Quit n");

do

{

printf("Enter your choice : ");

scanf("%d", &choice);

switch(choice)

{

case 1:

insert();

break;

case 2:

delete();

break;

case 3:

display();

break;

case 4:

exit(1);

default:

printf("Wrong choice n");

}

} while(choice!=4);

}

void insert()

{

int item;

if(rear == MAX - 1)

printf("Queue Overflow n");

else

{

if(front== - 1)

front = 0;

printf("Inset the element in queue : ");

scanf("%d", &item);

rear = rear + 1;

queue\_array[rear] = item;

}

}

void delete()

{

if(front == - 1 || front > rear)

{

printf("Queue Underflow \n");

return;

}

else

{

printf("Element deleted from queue is :%d\n", queue\_array[front]);

front = front + 1;

}

}

void display()

{

int i;

if(front == - 1)

printf("Queue is empty n");

else

{

printf("Queue is : n");

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

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

printf("n");

}

}

**Program 9: CIRCULAR QUEUE**

#include<stdio.h>

#include<conio.h>

int q[20],ele,i,f=0,r=-1,c=0,qs=8,k;

//Here f is front end,r is rear end and c is count

//qs is Queue Size

//To insert element

void Insert()

{

if(c==qs)

printf("Queue Overflow");

else

{ r=(r+1)%qs;

printf("Enter element");

scanf("%d",&ele);

q[r]=ele;

c++;

}

}

//To Delete element

void Delete()

{

if(c==0)

printf("Queue Underflow");

else

{

printf("%d has been deleted from queue",q[f]);

f=(f+1)%qs;

c--;

}

}

//To Display elements

void Display()

{

if (c==0)

printf("Queue Underflow");

else

{

for(i=1,k=f;i<=c;i++)

{

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

k=(k+1)%qs;

}

}

}

void main()

{

int choice,z;

While(i)

{

printf("\nMENU\n1=Insert\n2=Delete\n3=Display\n4=EXIT\nEnter your choice");

scanf("%d",&z);

switch(z)

{

case 1:Insert();

break;

case 2:Delete();

break;

case 3:Display();

break;

case 4: exit(0);

}

}

getch();

}