**A complete C code to implement Stack**

#include<stdio.h>

#include<conio.h>

#include<process.h>

#define max 5

void push();

void pop();

void displaytop();

void traverse();

int stack [max];

int top=-1;

void main()

{

int ch;

clrscr();

printf("enter any one among the numbers{1,2,3,4}");

printf("\nnumber 1 to push an item ");

printf("\nnumber 2 to pop an item ");

printf("\nnumber 3 to display top of an item ");

printf("\nnumber 4 to traverse all element ");

printf("\nnumber 5 to exit");

scanf("%d",&ch);

switch(ch)

{

case 1:

push();

break;

case 2:

pop();

break;

case 3:

displaytop();

break;

case 4:

traverse();

break;

case 5:

exit(0);

default:

printf("enter the option 1 to 5");

}

getch();

}

void push()

{

int item;

printf("Enter the item to be inserte\d");

scanf("%d",&item);

if(top==max-1)

{

printf("Stack overflow");

}

else

{

top++;

stack[top]=item;

printf("%ditem inserted successflly",top+1);

}

}

void pop()

{

int item;

if(top==-1)

{

printf("Stack underflow");

}

else

{

item=stack[top];

top--;

printf("Popped element=%d",item);

}

}

void displaytop()

{

int item;

item=stack[top];

printf("Top element=%d",item);

}

void traverse()

{

int i;

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

{

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

}

}

**A Complete C Code to implement Queue**

#include<stdio.h>

#include<process.h>

#include<conio.h>

#define max 5

void enqueue();

void dequeue();

void traverse();

int queue[max];

int front=0;

int rear=-1;

void enqueue()

{

if(rear==max-1)

{

printf("\nQueue overflow");

}

else

{

int item;

printf("\nEnter the item to be enqueued");

scanf("%d",&item);

rear++;

queue[rear]=item;

printf("\n One item inserted successfully");

}

}

void dequeue()

{

int item;

if(front>rear)

{

printf("Queue underflow");

}

else

{

item=queue[front];

printf("Dequeued item= %d ",item);

front++;;

}

}

void traverse()

{

printf("Items in the queue");

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

{

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

}

}

int main()

{

int ch;

int tmp;

printf("\n1: insert an item");

printf("\n2: Delete an item");

printf("\n3: Traverse all items");

printf("\n4: Exit");

scanf("%d",&ch);

switch(ch)

{

case 1:

enqueue();

break;

case 2:

dequeue();

break;

case 3:

traverse();

break;

case 4:

exit(0);

break;

default:

printf("\nPlease select option 1-4");

break;

}

getch();

}

**A complete C code to implement Circular Queue**

/\* Circular queue using array \*/

/\* Data Structures Using C by UDIT AGARWAL \*/

#include <stdio.h>

#include <conio.h>

#include<process.h>

#define MAX 5

void insert();

void del();

void display();

int cqueue[MAX];

int front = -1;

int rear = -1;

void main ( )

{

clrscr();

int choice;

while (1)

{

printf ("1.Insert\n");

printf ("2.Delete\n");

printf ("3.Display\n");

printf ("4.Quit\n");

printf ("Enter your choice :");

scanf ("%d", &choice);

switch(choice)

{

case 1 :

insert( );

break;

case 2 :

del( );

break;

case 3 :

display( );

break;

case 4 :

exit(0);

default:

printf("Wrong choice\n");

}

}

}

void insert( )

{

int item;

if((front==0 && rear==MAX-1) || (front==rear+1))

{

printf("Queue is Overflow\n");

return;

}

if (front==-1)/\*If queue is empty\*/

{

front = 0;

rear = 0;

}

else if (rear==MAX-1) /\*rear is at last position of queue\*/

{

rear = 0;

}

else

{

rear = rear + 1;

}

printf("Input the element for insertion :");

scanf("%d", &item);

cqueue[rear] = item;

}

void del( )

{

if (front == -1)

{

printf("Queue Underflow\n");

return;

}

printf ("Deleted element from queue is : %d\n", cqueue[front]);

if(front == rear) /\* queue has only one element \*/

{

front = -1;

rear = -1;

}

else if(front==MAX-1)

{

front = 0;

}

else

{

front = front + 1;  
 }

}

void display( )

{

int front\_pos = front, rear\_pos = rear;

if(front == -1)

{

printf("Queue is empty\n");

return;

}

printf ("Queue elements are:\n");

if(front\_pos <= rear\_pos)

{

while(front\_pos <= rear\_pos)

{

printf(" %d\n", cqueue[front\_pos]);

front\_pos++;

}

}

else

{

while(front\_pos <= MAX-1)

{

printf(" %d\n", cqueue[front\_pos]);

front\_pos++;

}

front\_pos = 0;

while(front\_pos <= rear\_pos)

{

printf(" %d\n", cqueue[front\_pos]);

front\_pos++;

}

}

printf("\n");

}

**A complete C code to implement Linear Search**

#include<stdio.h>

#include<conio.h>

#define N 10

int myarray[N];

void linearsearch();

void main()

{

linearsearch();

getch();

}

void linearsearch()

{

int key,loc=0;

printf("Enter the items");

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

{

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

}

printf("Enter the search key");

scanf("%d",&key);

while(loc<N)

{

if(myarray[loc]==key)

{

break;

}

loc++;

}

if(loc<N)

{

printf("Search is successful and found at position %d",loc);

}

else

{

printf("Search is unsuccessful");

}

}