**Stack code:**

#include<stdio.h>

#include<malloc.h>

typedef struct node

{

int data;

struct node \*next;

}NODE;

typedef NODE\* NODEPTR;

NODEPTR top,newNode;

void push(int gd)

{

newNode = (NODEPTR)malloc(sizeof(NODE));

newNode -> data = gd;

newNode -> next = NULL;

if( top == NULL )

top = newNode;

else

{

newNode -> next = top;

top = newNode;

}

}

void display()

{

NODEPTR tptr;

for(tptr = top ; tptr ; tptr = tptr -> next)

printf("%d ",tptr->data);

}

void pop()

{

NODE \*safe;

if(top==NULL)

printf("\nUnderflow");

else

{

safe=top;

top=top->next;

free(safe);

}

}

int main()

{

int num;

do

{

scanf("%d",&num);

if( num == -1) break;

push(num);

}while(1);

printf("\nStack elements after insertion:\n");

display();

pop();

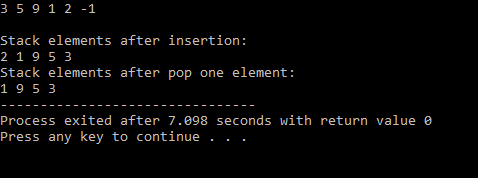
printf("\nStack elements after pop one element:\n");

display();

return 0;

}

**OUTPUT:**

****

**Queue code:**

#include<stdio.h>

#include<malloc.h>

typedef struct node

{

int data;

struct node \*next;

}NODE;

typedef NODE\* NODEPTR;

NODEPTR front, rear;

NODEPTR newNode , tptr;

void display( )

{

for( tptr = front ; tptr ; tptr = tptr -> next )

printf("%d ",tptr -> data);

}

void enque( int givenData)

{

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

newNode -> data = givenData;

newNode -> next = NULL;

if( front == NULL )

front=rear = newNode;

else

rear -> next = rear = newNode;//note here rear moves to next node

}

int deque()

{

NODEPTR safe ;

safe=front;

front=front->next;

free(safe);

}

int main()

{

int num;

do

{

scanf("%d",&num);

if( num == -1) break;

enque(num);

}while(1);

printf("\nQueue elements after insertion:\n");

display();

deque();

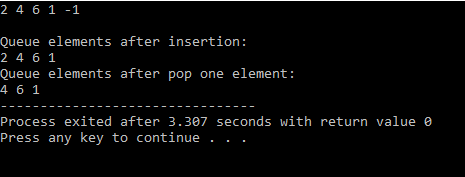
printf("\nQueue elements after pop one element:\n");

display();

return 0;

}

**OUTPUT:**

****