AIM:Implementation of a queue using linked list

ALOGIRTHIM:

Insertion:  
Step 1:allocation space for the new node ptr.

Step 2:ptr->val=val

If front=null.

Set front=rear=ptr

Set front->next=rear->next=null

Else

Set rear->next=ptr

Set rear=ptr

Set rear->next=null

End if

Step 3:end

Deletion

Step 1:if front==null

Write underflow

End if

Step 2:set ptr=front

Step 3:set front=front->next

Step 4:free ptr.

End

Display:

Step 1:if ptr==null

Write empty queue

Step 2:else

While(ptr!=null)

Print all the values and increment the pointer.

PROGRAM:

//Implementation of queue using Linked List

#include<stdio.h>

#include<stdlib.h>

#define max 5

void insert();

void delete();

void display();

int queue[max];

struct node

{

int data;

struct node \*next;

};

struct node \*front ,\*rear;

void insert()

{

struct node \*ptr;

int item;

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

if (ptr == NULL)

{

printf("Overflow\n");

return;

}

else

{ printf("Enter the Element :\n");

scanf("%d",&item);

ptr->data=item;

if (front == NULL)

{

front=rear=ptr;

front -> next=NULL;

rear -> next=NULL;

}

else

{

rear -> next=ptr;

rear=ptr;

rear -> next=NULL;

}

}

}

void delete()

{

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

if(front==NULL)

{

printf("UnderFlow\n");

return;

}

else

{

ptr=front;

printf("Deleted Element %d\n",ptr->data);

front=front->next;

free(ptr);

}

}

void display()

{

struct node \*ptr;

ptr=front;

if(ptr==NULL)

{

printf("UnderFlow\n");

return;

}

else

{

printf("Printing Elements:\n");

while(ptr!=NULL)

{

printf("%d\n",ptr->data);

ptr=ptr->next;

}

}

}

int main()

{

int choice;

while(1)

{

printf("Operation performed by Queue:\n");

printf("1.Insert\n2.Delete\n3.Display\n4.Exit\n");

printf("Enter Your Choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1: insert();

break;

case 2: delete();

break;

case 3: display();

break;

case 4: exit(0);

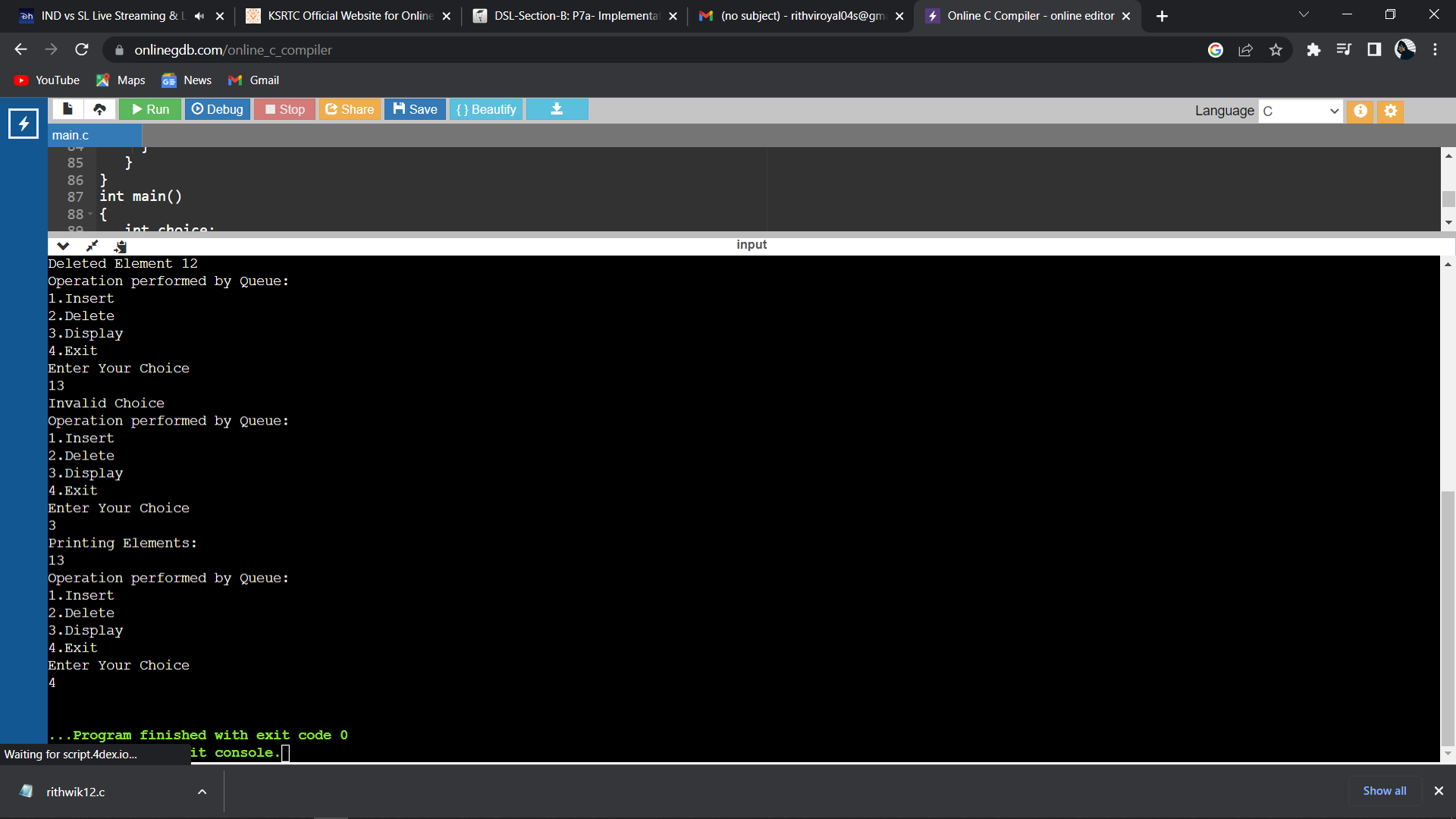
default: printf("Invalid Choice\n");

}

}

}

OUTPUT:



GITHUN LINK: