4.Linked Stack

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
int data;
struct node *link;
};
typedef struct node ll;
ll * first = NULL, *top=NULL;
int data = 0, choice = 0, sizeoflist;
int main ()
{
while (choice != 4)
 {
  menu ();
  scanf ("%d", &choice);
  switch (choice)
   {
   case 1: push();
             break;
   case 2: pop();
            break ;
   case 3: print();
```

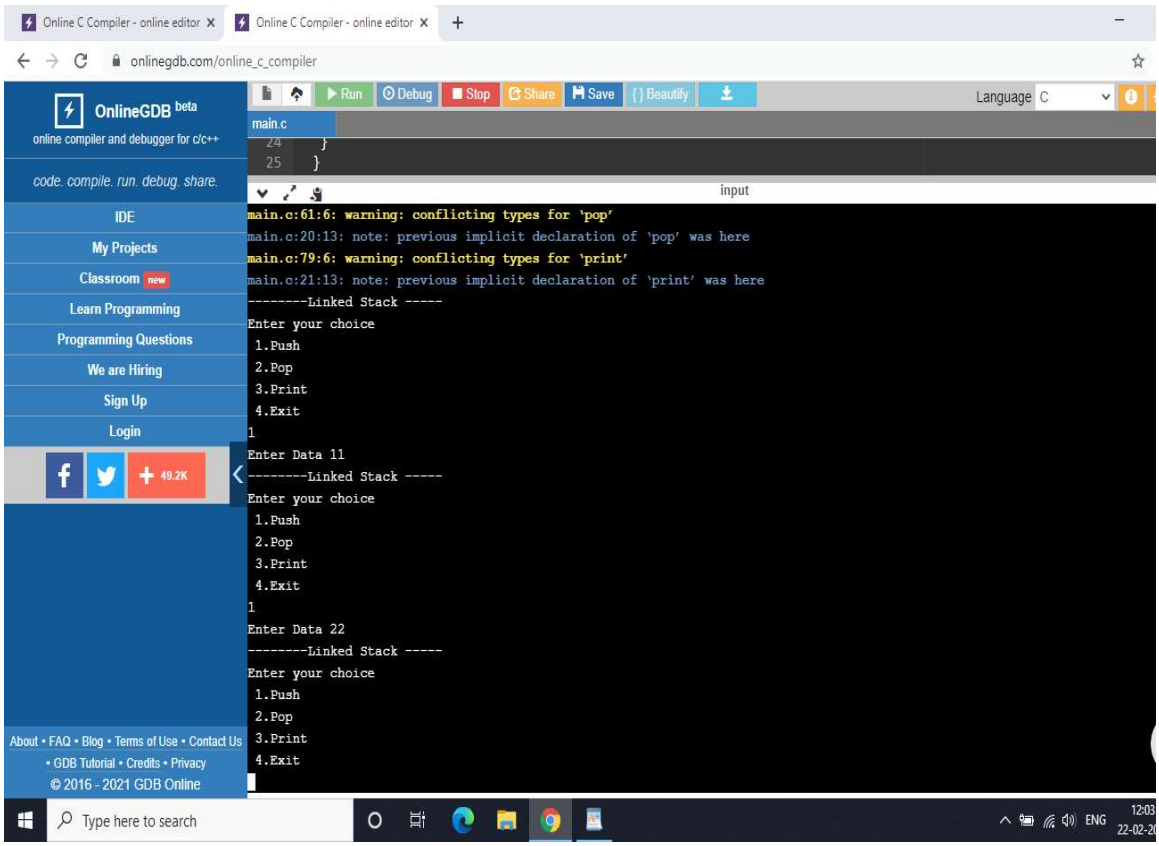
```
            break;
   case 4: exit (0);
   default: printf ("Wrong choice");
  }
 }
 return 0;
}
void menu (){
  printf("--------Linked Stack -----");
   printf ("\nEnter your choice \n");
   printf(" 1.Push \n 2.Pop \n 3.Print \n 4.Exit \n");
   return;
}
void push()//Push opeartion
{
 ll *temp;int i;ll * temp1;
 if(first == NULL)
 {
    first = (ll*) malloc(sizeof(ll));
    printf("Enter Data ");
    scanf("%d",&data);
    first->data =data;
    first->link = NULL;
    top = first;
```

```
    return;
 }
 temp = first;
 for(i=0;temp->link != NULL;)
 {
   temp=temp->link;
 }
    temp1 = (ll*) malloc(sizeof(ll));
    printf("Enter Data ");
    scanf("%d",&data);
    temp1->data =data;
    temp->link = temp1;
    return;
}
void pop()//Pop operation
{
 ll *t1=NULL, *t2=NULL;
 if(first == NULL)
 {
    printf("Stack Empty ");
    return;
 }
 t2 = first;
 for(;t2->link != NULL;)
```

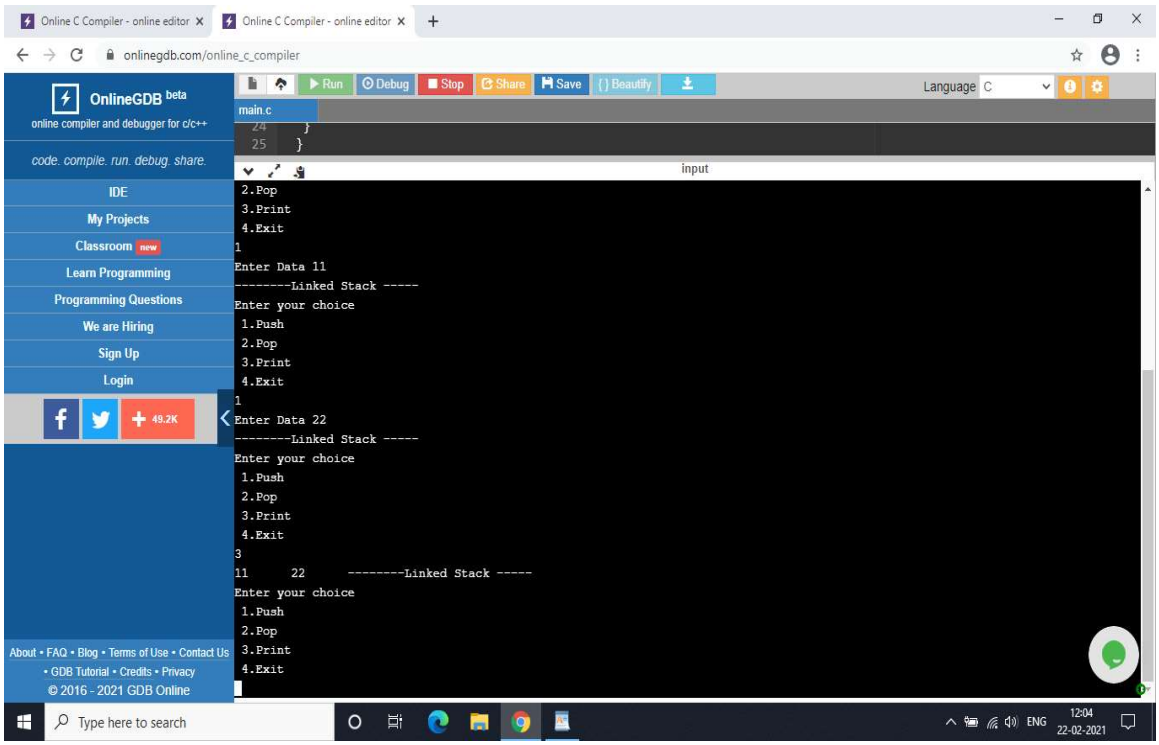
```
{
   t1 = t2;
   t2=t2->link;
}
printf("Poped Element is %d \n",t2->data);
t1->link = NULL;
free(t2);
}
void print()//Printing the elements
{
ll *t2=NULL;
if(first == NULL)
{
   printf("Stack Empty ");
   return;
}
t2 = first;
for(;t2!=NULL;)
{
   printf("%d \t",t2->data);
   t2=t2->link;
}

}
```

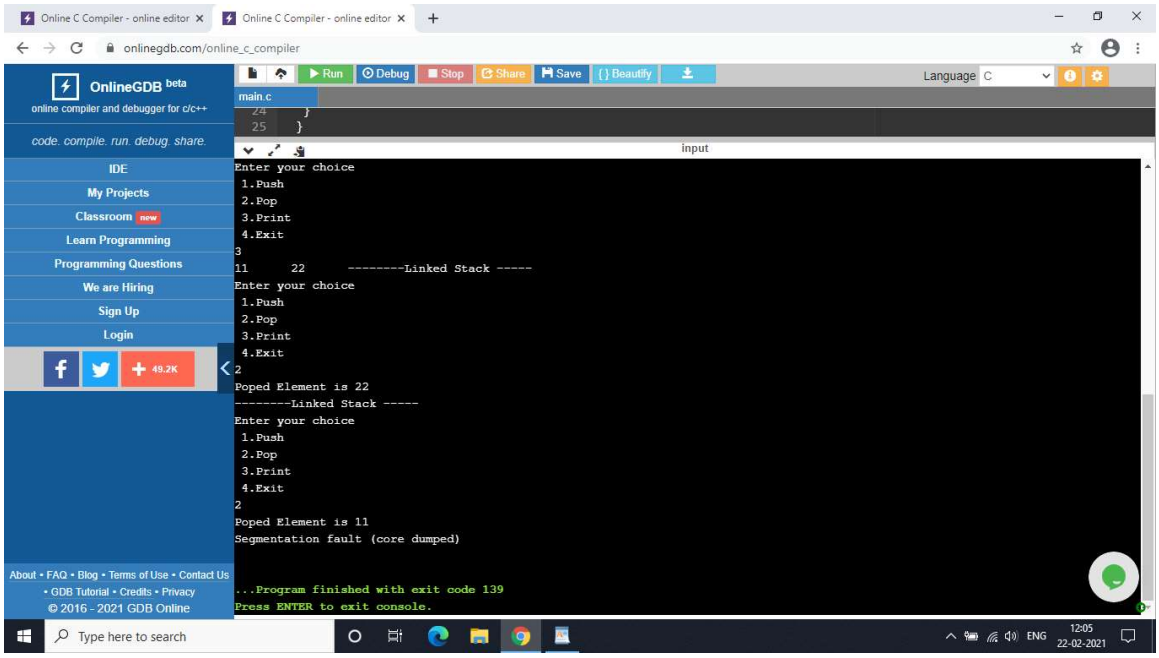
Output:


Online C Compiler - online editor
Online C Compiler - online editor
onlinegdb.com/online_c_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
We are Hiring
Sign Up
Login
49.2K
About • FAQ • Blog • Terms of Use • Contact Us
• GDB Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
24 }
25 }
input
main.c:61:6: warning: conflicting types for 'pop'
main.c:20:13: note: previous implicit declaration of 'pop' was here
main.c:79:6: warning: conflicting types for 'print'
main.c:21:13: note: previous implicit declaration of 'print' was here
--------Linked Stack -----
Enter your choice
1.Push
2.Pop
3.Print
4.Exit
1
Enter Data 11
--------Linked Stack -----
Enter your choice
1.Push
2.Pop
3.Print
4.Exit
1
Enter Data 22
--------Linked Stack -----
Enter your choice
1.Push
2.Pop
3.Print
4.Exit
Type here to search
ENG

PUSH


Online C Compiler - online editor
onlinegdb.com/online_c_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
We are Hiring
Sign Up
Login
49.2K
About • FAQ • Blog • Terms of Use • Contact Us
• GDB Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online
Run
Debug
Stop
Share
Save
{} Beautify
Language C
main.c
input
2.Pop
3.Print
4.Exit
1
Enter Data 11
--------Linked Stack -----
Enter your choice
1.Push
2.Pop
3.Print
4.Exit
1
Enter Data 22
--------Linked Stack -----
Enter your choice
1.Push
2.Pop
3.Print
4.Exit
3
11 22 --------Linked Stack -----
Enter your choice
1.Push
2.Pop
3.Print
4.Exit
Type here to search
ENG
12:04
22-02-2021

PRINT


Online C Compiler - online editor
onlinegdb.com/online_c_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
We are Hiring
Sign Up
Login
49.2K
About • FAQ • Blog • Terms of Use • Contact Us
• GDB Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online
Run
Debug
Stop
Share
Save
{} Beautify
Language C
main.c
input
Enter your choice
1.Push
2.Pop
3.Print
4.Exit
3
11 22 --------Linked Stack -----
Enter your choice
1.Push
2.Pop
3.Print
4.Exit
2
Poped Element is 22
--------Linked Stack -----
Enter your choice
1.Push
2.Pop
3.Print
4.Exit
2
Poped Element is 11
Segmentation fault (core dumped)
...Program finished with exit code 139
Press ENTER to exit console.
Type here to search
ENG
12:05
22-02-2021

POP

5.6 LINKED LIST Opeartion:Insert at Beg,Insert at position

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
 int data;
 struct node *link;
};
typedef struct node ll;
ll *first=NULL,*temp,sizeoflist;
int data=0,choice=0;
int main ()
{
while(choice != 4)
{
 menu();
 scanf("%d",&choice);
 switch(choice)
 {
 case 1:insertbeg();
       break;
 case 2:insertpos();
       break;
```

```
case 3:print();
      break;
case 4: exit(0);
default : printf("Wrong choice");
}
}
return 0;
}
void menu()
{
printf("\nEnter your choice\n");
printf("1.Insert Beg \n2.Insert at position \n3.Print");
return;
}

void insertbeg()
{
if(first == NULL) // if the list is empty
{
first = (ll *)malloc(sizeof(ll));
printf("Enter data for the node \n");
scanf("%d",&data);
first->data = data;
first->link = NULL;
```

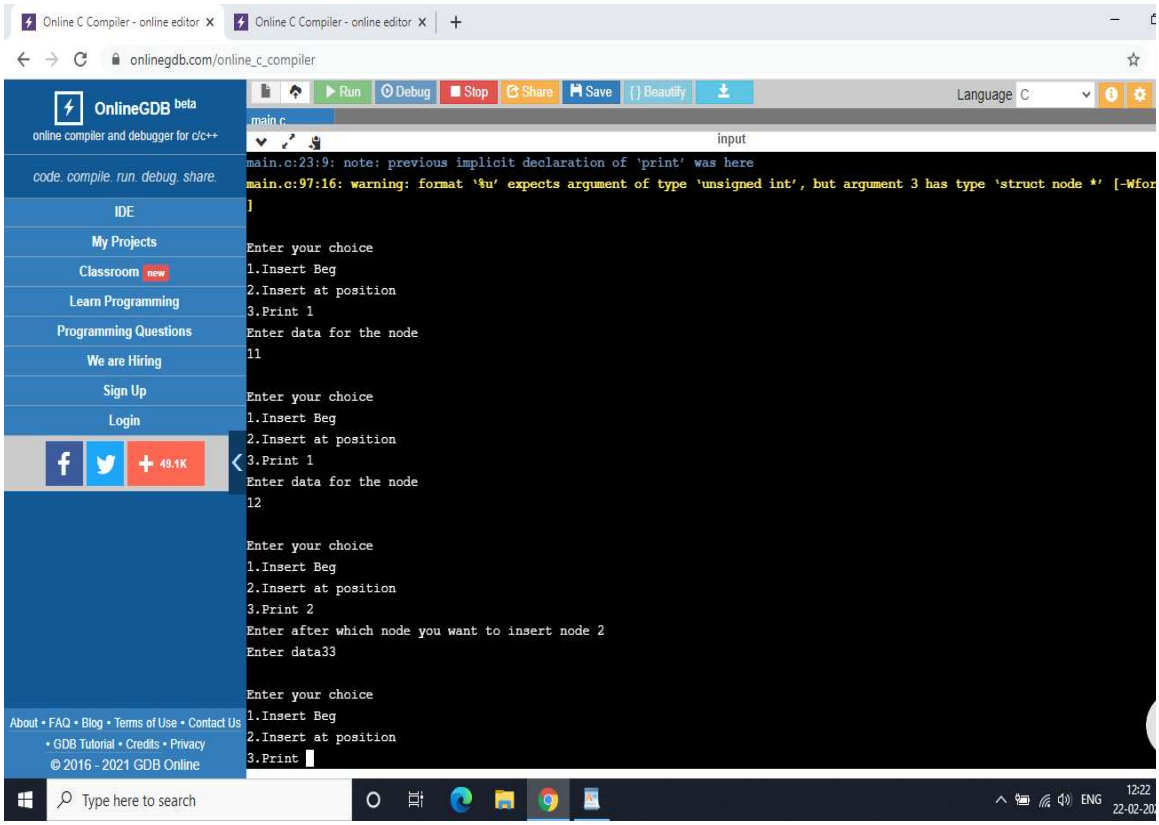
```
}

else
{
temp = (ll *)malloc(sizeof(ll));
printf("Enter data for the node \n");
scanf("%d",&data);
temp->data = data;
temp->link = first;
first = temp;
}
}
void insertpos()
{
  int position=0; int tdata; ll * temp1;
  printf("Enter after which node you want to insert node ");
  scanf("%d",&position);
   if(first == NULL)
{
  printf("Linked List Does not exit ");
  return;
}
if(position == 0)
{
```

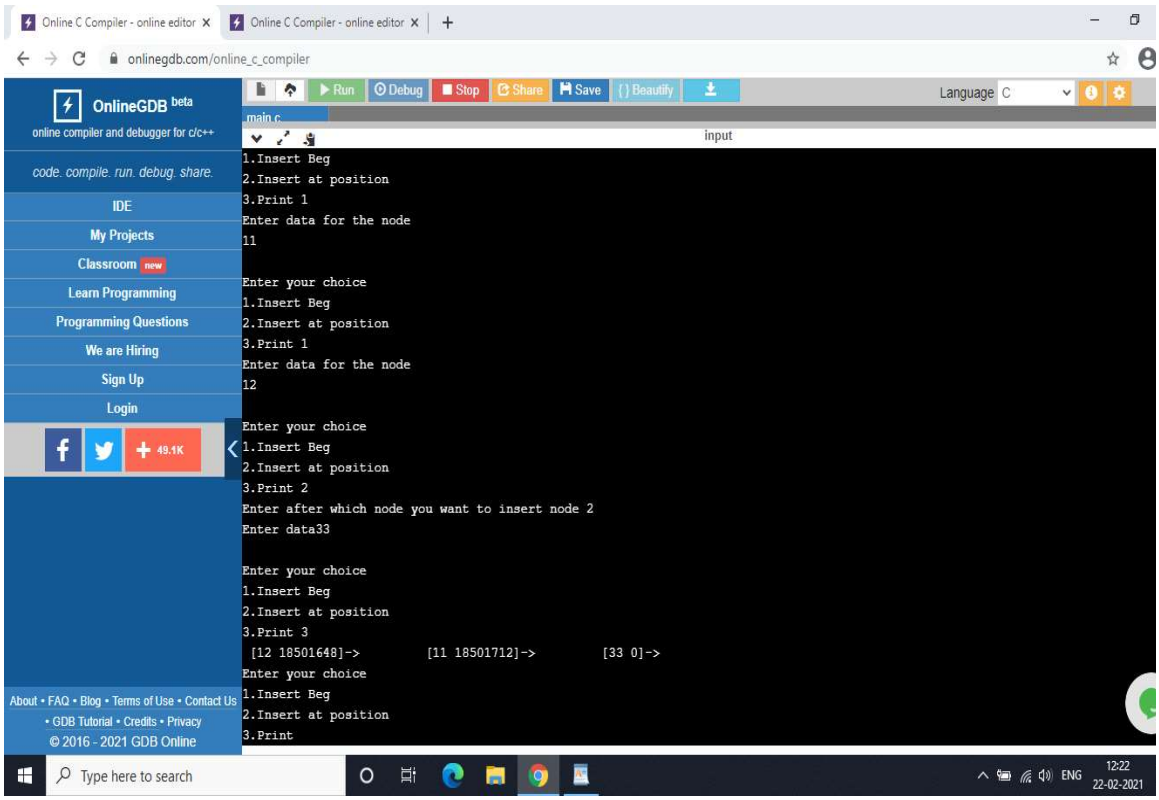
```
 insertbeg();
 return;
}

temp = first;
for (int i = 1; i < position; i++) {
temp = temp->link;
}
printf("Enter data");
scanf("%d",&tdata);
temp1 = (ll *) malloc(sizeof(ll));
temp1->link = temp->link;
temp->link = temp1;
temp1->data = tdata;
}
void print(){
 if(first == NULL)
{
   printf("List is empty ");
}
else
{
  temp = first;
 while(temp != NULL)
```

```
{
printf(" [%d %u]-> \t",temp->data,temp->link);
temp = temp -> link;
}
}
}
}
```



INSERT AT BEG AND INSERT AT POSITION

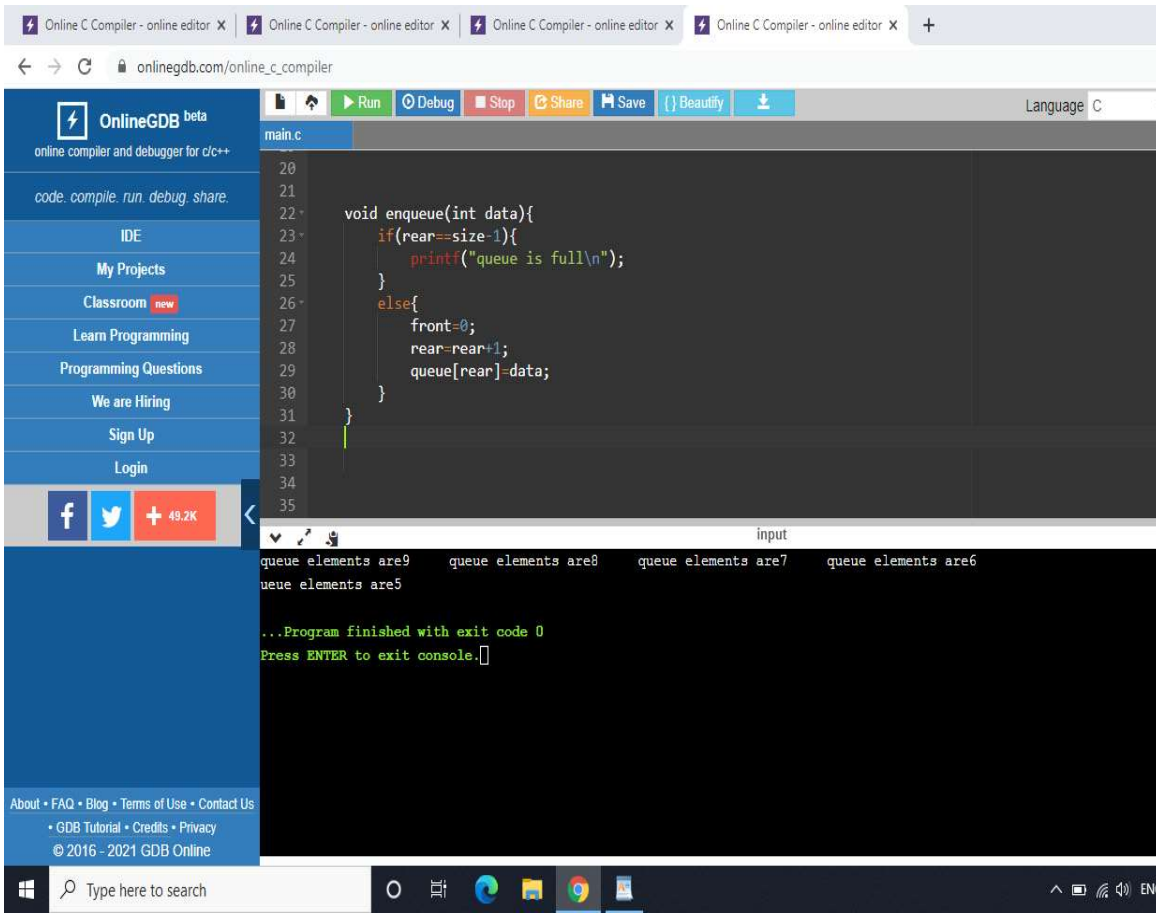


PRINTING THE ELEMENTS

3.QUEUE

```
#include<stdio.h>
void enqueue(int data);
int size=5,front=-1,rear=-1,data;
int queue[5];
int main()
{
   int data[5];
    data[0]=9;
    data[1]=8;
    data[2]=7;
```

```
    data[3]=6;
    data[4]=5;
    for(int i=0;i<5;i++){
        enqueue(data[i]);
       printf("queue elements are%d\t",queue[i]);
    }
    return 0;
}
   void enqueue(int data){
      if(rear==size-1){
         printf("queue is full\n");
      }
      else{
         front=0;
         rear=rear+1;
         queue[rear]=data;
      }
}
```



Queued elements