Name - Shantanu Deshpande
Roll no. - 7
Class - SYCSE
Batch - S1
Sub. - D.S

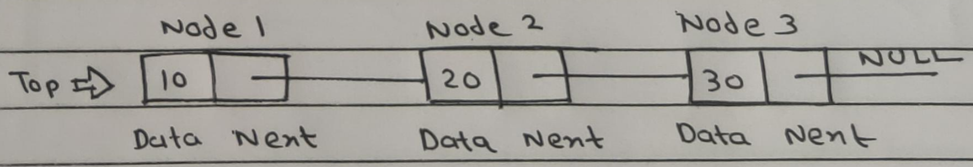
Practical No. 4

Aim ÷ Implementation of stack and queue using linked list

Theory ÷

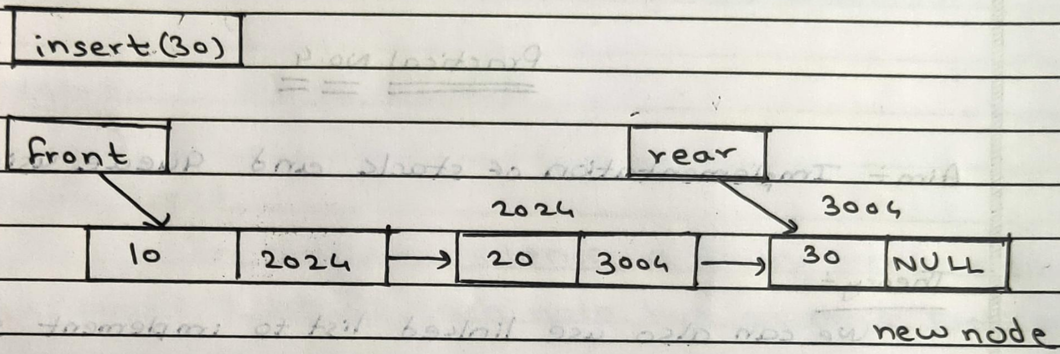
We can also use linked list to implement stack. Linked list allocates the memory dynamically. However, time complexity in both the scenario is same for all the operations i.e. push, pop and peek.

In linked list implementation of stack, the nodes are maintained non-contiguously in the memory. Each node contains a pointer to its immediate successor node in the stack. stack is said to be overflown if the space left in the memory heap is not enough to create a node.



stack implementation using Linked list

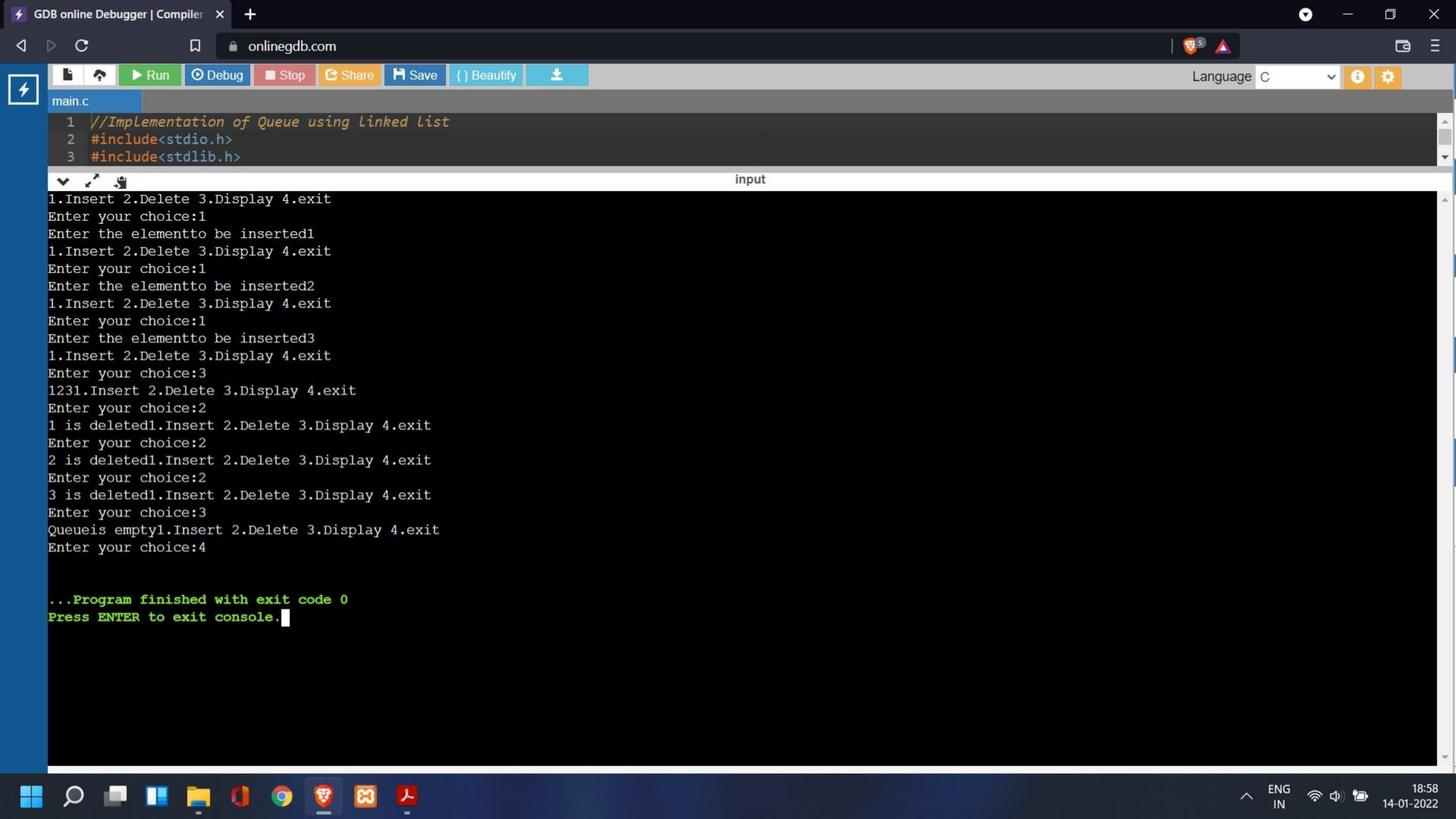
The array implementation can not be used for the large scale applications where the queues are implemented. One of the alternative of array implementation is linked list implementation of queue.



Conclusion:- Thus we have implemented stack and queue using linked list.

Queue using linked list=>

```
//Implementation of Queue using linked list
#include<stdio.h>
#include<stdlib.h>
struct node
{
int info;
struct node *next;
}*front=NULL,*rear=NULL,*p,*temp,*q;
void insert(int);
void del();
void display();
void main()
{
int ch=1,x;
while(ch!=4)
{
printf("1.Insert 2.Delete 3.Display 4.exit \nEnter your choice:");
scanf("%d",&ch);
switch(ch)
{
case 1:
printf("Enter the elementto be inserted");
scanf("%d",&x);
insert(x);
break;
case 2:
del();
break;
case 3:
display();
break;
case 4:
break;
}
}
}
void insert(int ele)
{
p=(struct node*)malloc(sizeof(struct node));
p->info=ele;
p->next=NULL;
if(p==NULL)
{
printf("Queue overflow");
}
else
{
if(front==NULL) //Queue is empty
{
front=p;
rear=p;
}
else
{
rear->next=p;
rear=p;
}
}
}
void del()
{
struct node *temp;
if(front== NULL)
{
printf("Queue underflow");
}
else
{
temp=front;
front=front->next;
printf("%d is deleted",temp->info);
// if(front==NULL)
//rear=NULL;
free(temp);
}
}
void display()
{
if(front==NULL)
printf("Queueis empty");
else
{
q=front;
while(q!=NULL)
{
printf("%d",q->info);
q=q->next;
}
}
}
```



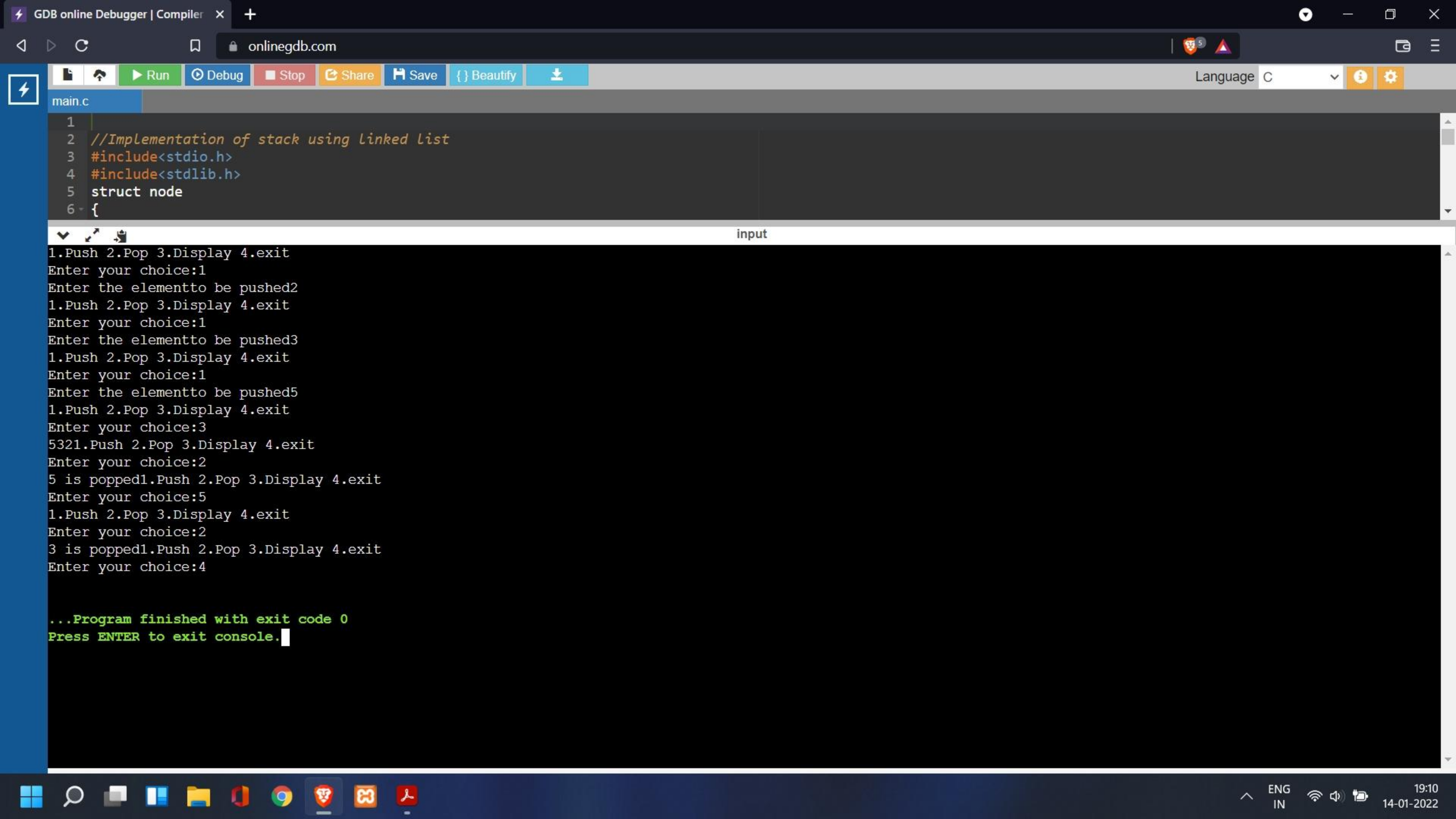
```c
//Implementation of Queue using linked list
#include<stdio.h>
#include<stdlib.h>
```

```
1.Insert 2.Delete 3.Display 4.exit
Enter your choice:1
Enter the elementto be inserted1
1.Insert 2.Delete 3.Display 4.exit
Enter your choice:1
Enter the elementto be inserted2
1.Insert 2.Delete 3.Display 4.exit
Enter your choice:1
Enter the elementto be inserted3
1.Insert 2.Delete 3.Display 4.exit
Enter your choice:3
1231.Insert 2.Delete 3.Display 4.exit
Enter your choice:2
1 is deleted1.Insert 2.Delete 3.Display 4.exit
Enter your choice:2
2 is deleted1.Insert 2.Delete 3.Display 4.exit
Enter your choice:2
3 is deleted1.Insert 2.Delete 3.Display 4.exit
Enter your choice:3
Queueis empty1.Insert 2.Delete 3.Display 4.exit
Enter your choice:4


...Program finished with exit code 0
Press ENTER to exit console.
```

Stack using linked list=>

```
//Implementation of stack using linked list
#include<stdio.h>
#include<stdlib.h>
struct node
{
int info;
struct node *next;
}*top=NULL,*p,*q;
void push(int);
void pop();
void display();
void main()
{
int ch=1,x;
while(ch !=4)
{
printf("1.Push 2.Pop 3.Display 4.exit \nEnter your choice:");
scanf("%d",&ch);
switch(ch)
{
case 1:
printf("Enter the elementto be pushed");
scanf("%d",&x);
push(x);
break;
case 2:
pop();
break;
case 3:
display();
break;
case 4:
break;
}
}
}
void push(int ele)
{
p=(struct node*)malloc(sizeof(struct node));
if(p==NULL)
{
printf("stack overflow");
}
else
{
p->info=ele;
p->next=top;
top=p;
}
}
void pop()
{
struct node *temp;
if(top==NULL)
printf("stack underflow");
else
{
temp=top;
top=top->next;
printf("%d is popped",temp->info);
free(temp);
}
}
void display()
{
for (q=top;q!=NULL;q=q->next)
printf("%d",q->info);
}
```



```
//Implementation of stack using linked list
#include<stdio.h>
#include<stdlib.h>
struct node
{
```

```
1.Push 2.Pop 3.Display 4.exit
Enter your choice:1
Enter the elementto be pushed2
1.Push 2.Pop 3.Display 4.exit
Enter your choice:1
Enter the elementto be pushed3
1.Push 2.Pop 3.Display 4.exit
Enter your choice:1
Enter the elementto be pushed5
1.Push 2.Pop 3.Display 4.exit
Enter your choice:3
5321.Push 2.Pop 3.Display 4.exit
Enter your choice:2
5 is popped1.Push 2.Pop 3.Display 4.exit
Enter your choice:5
1.Push 2.Pop 3.Display 4.exit
Enter your choice:2
3 is popped1.Push 2.Pop 3.Display 4.exit
Enter your choice:4


...Program finished with exit code 0
Press ENTER to exit console.
```