Data Structures

Experiment no. 4

Develop code to implement Stack and Queue using Linked List

Q. WAP in C to implement Stack using Linked List.

Code:

```
#include<stdio.h>
#include<conio.h>

struct Node* createNode(int);

struct Node
{
            int data;
            struct Node* next;
};

struct Node *node,*tos=NULL,*temp=NULL;

struct Node* createNode(int info)
{
            struct Node *node = (struct Node*)malloc(sizeof(struct Node));
            node -> data=info;
            node -> next=NULL;
            return node;
}

void push(int info)
{
            node=createNode(info);
            if(tos==NULL)
            {
                        tos=node;
            }
            else
            {
                        node -> next=tos;
                        tos=node;
            }
}
```

```
void pop()
{
        if(tos==NULL)
        {
                printf("\nStack is Empty!");
        }
        else
        {
                temp=tos;
                tos=tos -> next;
                printf("%d",temp->data);
                free(temp);
        }
}

void stackTop()
{
        if(tos==NULL)
        {
                printf("\nStack is Empty!");
        }
        else
        {
                printf("%d",tos->data);
        }
}

void display()
{
        if(tos==NULL)
        {
                printf("\nStack is Empty!");
        }
        else
        {
                temp=tos;
                while(temp!=NULL)
                {
                        printf("%d\t",temp->data);
                        temp=temp -> next;
                }
        }
}

void main()
{
        int info,op;
        clrscr();
```

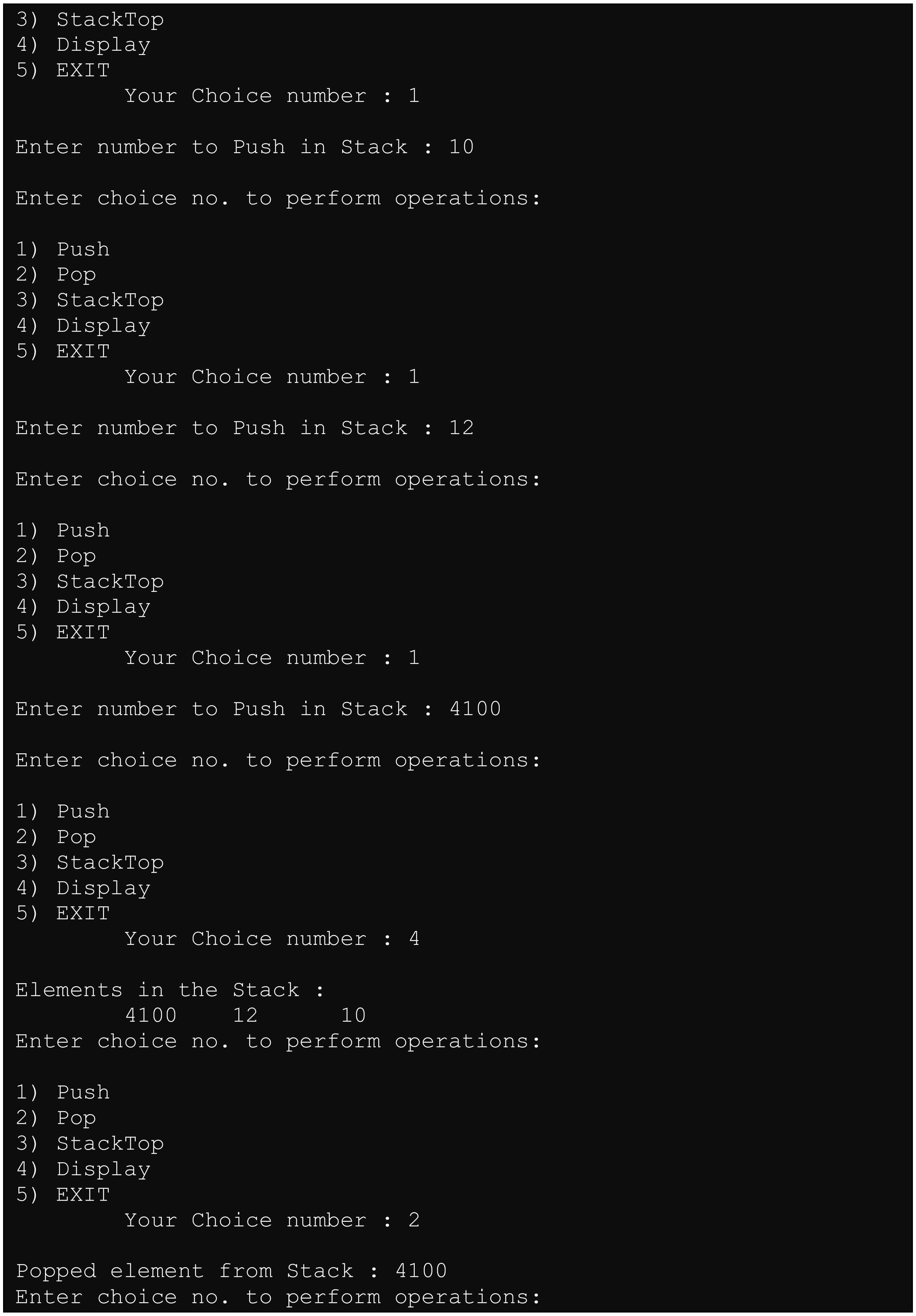
- Chaitanya Shah

```
        do
        {
                printf("\nEnter choice no. to perform operations:\n");
                printf("\n1) Push\n2) Pop\n3) StackTop\n4) Display\n5) EXIT\n\tYour Choice
number : ");
                scanf("%d",&op);

                switch(op)
                {
                        case 1:
                                printf("\nEnter number to Push in Stack : ");
                                scanf("%d",&info);
                                push(info);
                                break;

                        case 2:
                                printf("\nPopped element from Stack : ");
                                pop();
                                break;

                        case 3:
                                printf("\nTop element of Stack : ");
                                stackTop();
                                break;

                        case 4:
                                printf("\nElements in the Stack : \n\t");
                                display();
                                break;

                        case 5:
                                exit();
                                break;

                        default:
                                printf("\n\tEnter a valid choice number!");
                }
        } while(op!=5);
        getch();
}
```

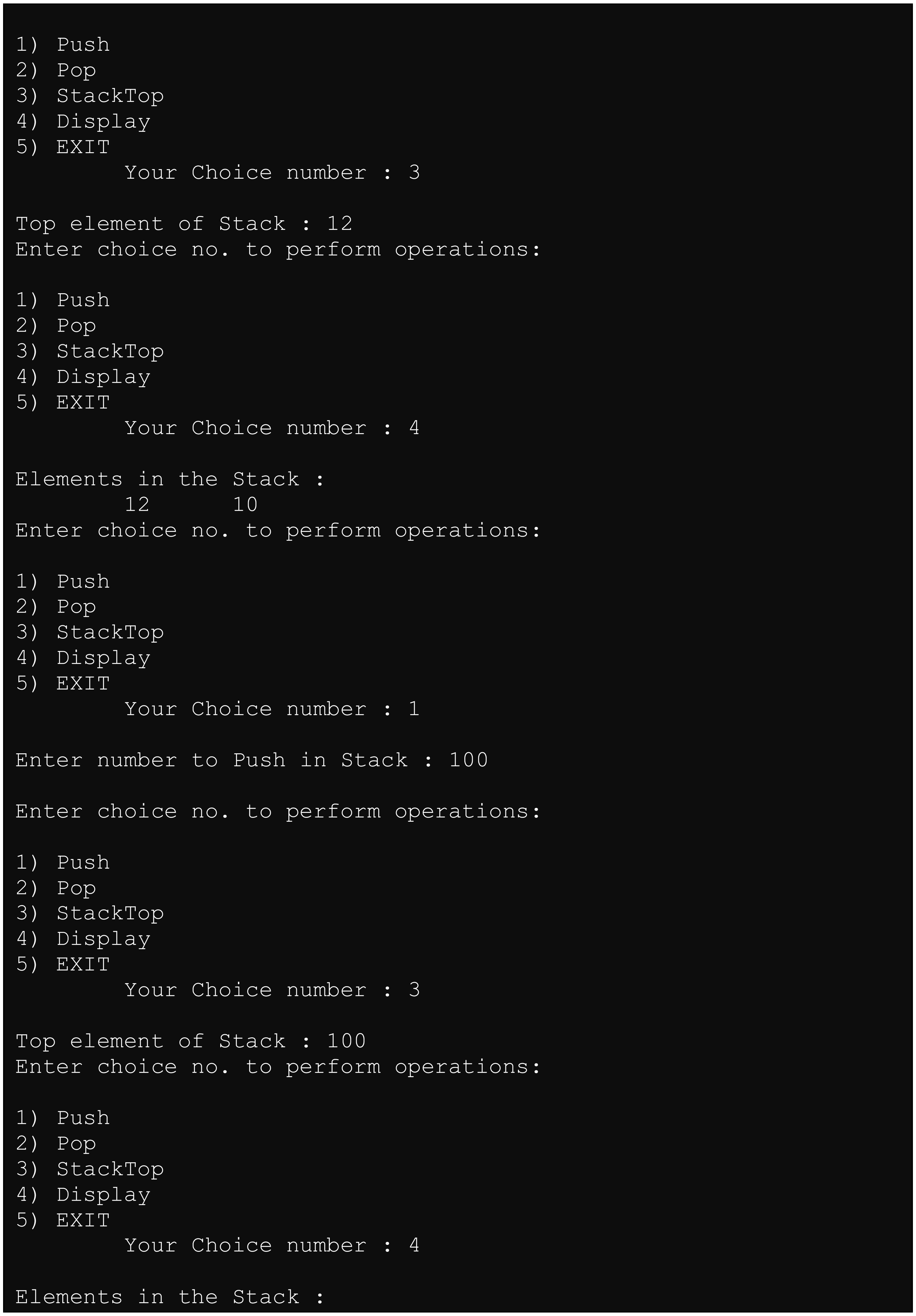
Output:

```
Enter choice no. to perform operations:

1) Push
2) Pop
```

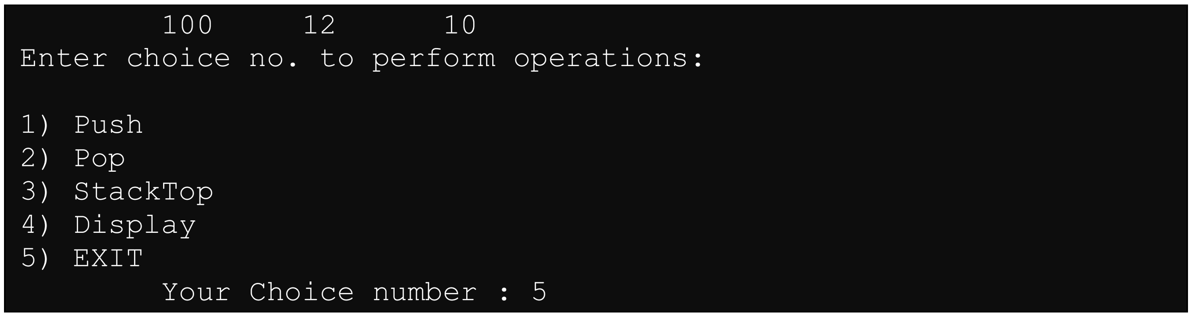
- Chaitanya Shah



```
3) StackTop
4) Display
5) EXIT
        Your Choice number : 1

Enter number to Push in Stack : 10

Enter choice no. to perform operations:

1) Push
2) Pop
3) StackTop
4) Display
5) EXIT
        Your Choice number : 1

Enter number to Push in Stack : 12

Enter choice no. to perform operations:

1) Push
2) Pop
3) StackTop
4) Display
5) EXIT
        Your Choice number : 1

Enter number to Push in Stack : 4100

Enter choice no. to perform operations:

1) Push
2) Pop
3) StackTop
4) Display
5) EXIT
        Your Choice number : 4

Elements in the Stack :
        4100    12      10
Enter choice no. to perform operations:

1) Push
2) Pop
3) StackTop
4) Display
5) EXIT
        Your Choice number : 2

Popped element from Stack : 4100
Enter choice no. to perform operations:
```



```
1) Push
2) Pop
3) StackTop
4) Display
5) EXIT
        Your Choice number : 3

Top element of Stack : 12
Enter choice no. to perform operations:

1) Push
2) Pop
3) StackTop
4) Display
5) EXIT
        Your Choice number : 4

Elements in the Stack :
        12      10
Enter choice no. to perform operations:

1) Push
2) Pop
3) StackTop
4) Display
5) EXIT
        Your Choice number : 1

Enter number to Push in Stack : 100

Enter choice no. to perform operations:

1) Push
2) Pop
3) StackTop
4) Display
5) EXIT
        Your Choice number : 3

Top element of Stack : 100
Enter choice no. to perform operations:

1) Push
2) Pop
3) StackTop
4) Display
5) EXIT
        Your Choice number : 4

Elements in the Stack :
```



```
        100     12      10
Enter choice no. to perform operations:

1) Push
2) Pop
3) StackTop
4) Display
5) EXIT
        Your Choice number : 5
```

- Chaitanya Shah

Q. WAP in C to implement Queue using Linked List.

Code:

```
#include<stdio.h>
#include<conio.h>

struct Node* createNode(int);

struct Node
{
	int data;
	struct Node* next;
};

struct Node *node,*front=NULL, *rear=NULL, *temp=NULL;

struct Node* createNode(int info)
{
	struct Node *node = (struct Node*)malloc(sizeof(struct Node));
	node -> data=info;
	node -> next=NULL;
	return node;
}

void Enqueue(int info)
{
	node=createNode(info);
	if(front==NULL && rear==NULL)
	{
		front=node;
		rear=node;
	}
	else
	{
		rear -> next=node;
		rear=node;
	}
}

void Dequeue()
{
	if(front==NULL && rear==NULL)
	{
		printf("\nQueue is Empty!");
	}
	else
	{
```

- Chaitanya Shah

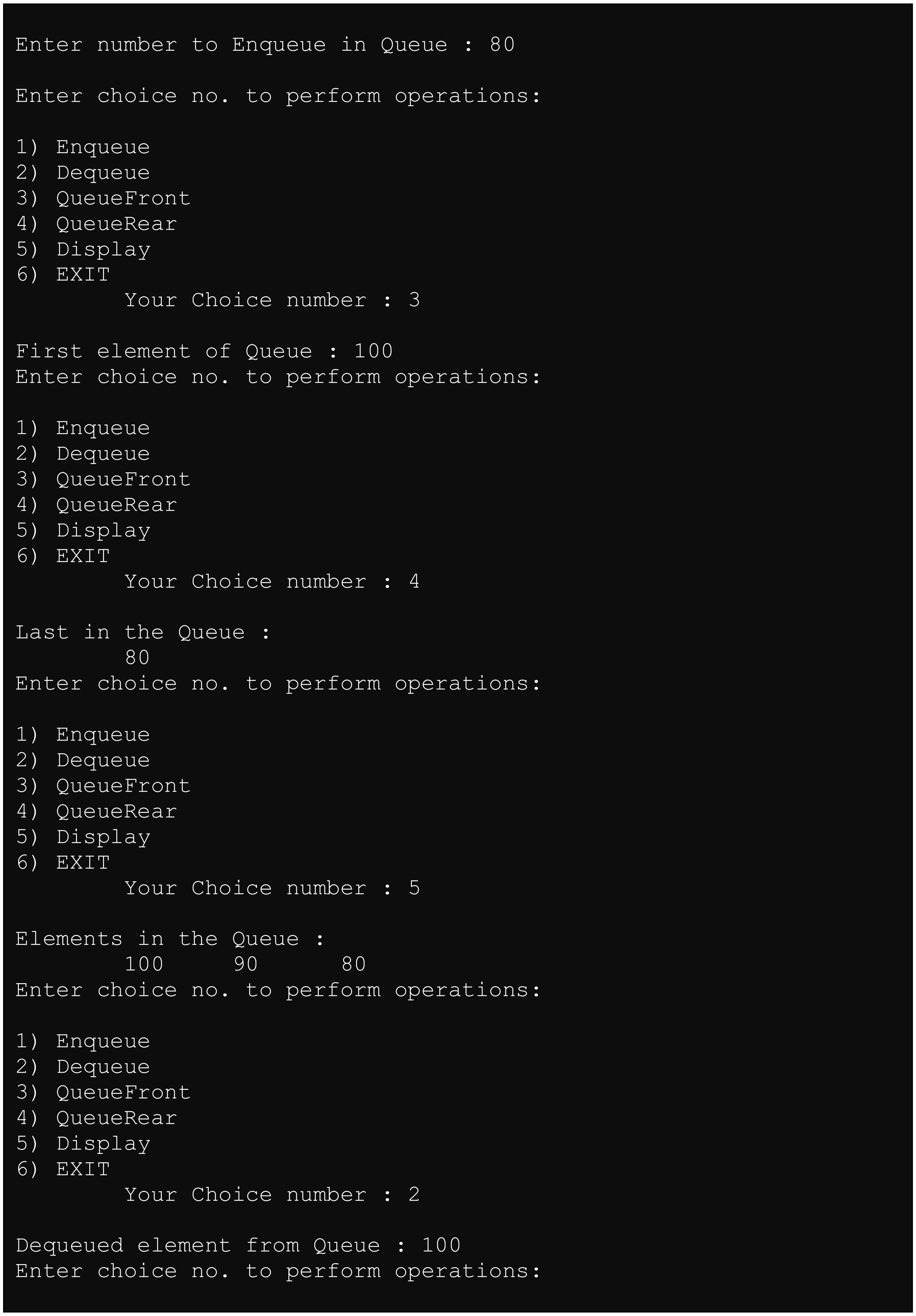
```
                temp=front;
                front=front -> next;
                printf("%d",temp->data);
                free(temp);
        }
}

void QueueFront()
{
        if(front==NULL && rear==NULL)
        {
                printf("\nQueue is Empty!");
        }
        else
        {
                printf("%d",front->data);
        }
}

void QueueRear()
{
        if(front==NULL && rear==NULL)
        {
                printf("\nQueue is Empty!");
        }
        else
        {
                printf("%d",rear->data);
        }
}

void display()
{
        if(front==NULL && rear==NULL)
        {
                printf("\nQueue is Empty!");
        }
        else
        {
                temp=front;
                while(temp!=NULL)
                {
                        printf("%d\t",temp->data);
                        temp=temp->next;
                }
        }
}
```

```
void main()
{
	int info,op;
	clrscr();
	do
	{
		printf("\nEnter choice no. to perform operations:\n");
		printf("\n1) Enqueue\n2) Dequeue\n3) QueueFront\n4) QueueRear\n5) 
Display\n6) EXIT\n\tYour Choice number : ");
		scanf("%d",&op);

		switch(op)
		{
			case 1:
				printf("\nEnter number to Enqueue in Queue : ");
				scanf("%d",&info);
				Enqueue(info);
				break;

			case 2:
				printf("\nDequeued element from Queue : ");
				Dequeue();
				break;

			case 3:
				printf("\nFirst element of Queue : ");
				QueueFront();
				break;

			case 4:
				printf("\nLast in the Queue : \n\t");
				QueueRear();
				break;

			case 5:
				printf("\nElements in the Queue : \n\t");
				display();
				break;

			case 6:
				exit();
				break;

			default:
				printf("\n\tEnter a valid choice number!");
		}
	} while(op!=6);
	getch();
```

```
}
```

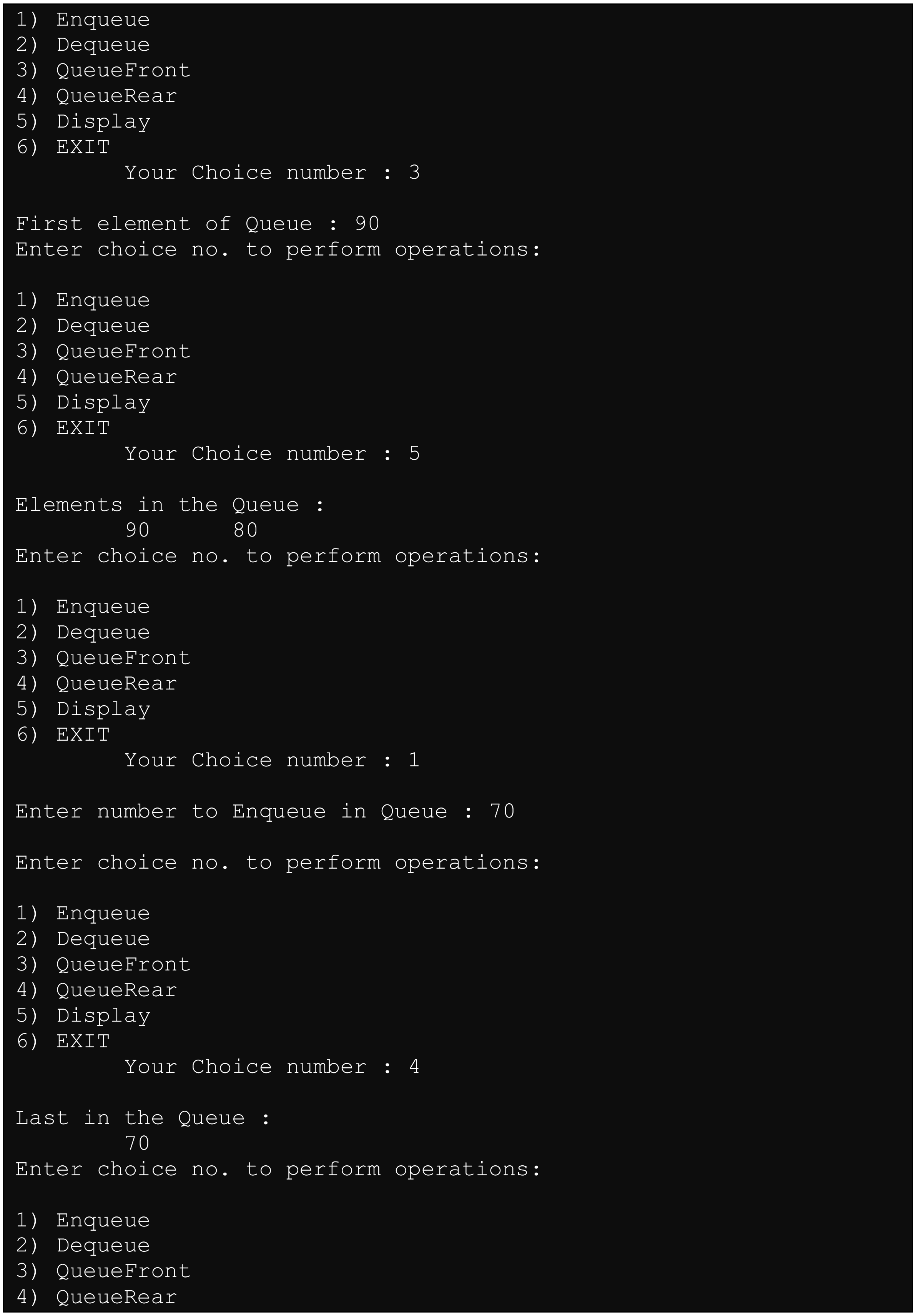
Output:

```
Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 5

Elements in the Queue :

Queue is Empty!
Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 1

Enter number to Enqueue in Queue : 100

Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 1

Enter number to Enqueue in Queue : 90

Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 1
```



```
Enter number to Enqueue in Queue : 80

Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 3

First element of Queue : 100
Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 4

Last in the Queue :
        80
Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 5

Elements in the Queue :
        100     90      80
Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 2

Dequeued element from Queue : 100
Enter choice no. to perform operations:
```

- Chaitanya Shah



```
1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 3

First element of Queue : 90
Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 5

Elements in the Queue :
        90      80
Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 1

Enter number to Enqueue in Queue : 70

Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 4

Last in the Queue :
        70
Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
```

- Chaitanya Shah

```
5) Display
6) EXIT
        Your Choice number : 5

Elements in the Queue :
        90      80      70
Enter choice no. to perform operations:

1) Enqueue
2) Dequeue
3) QueueFront
4) QueueRear
5) Display
6) EXIT
        Your Choice number : 6
```

- Chaitanya Shah