PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043
Department of Electronics & Telecommunication
ASSESMENT YEAR: 2020-2021 CLASS: SE V
SUBJECT: Data Structure and Algorithm
Assg No: 7 Roll No: 22119 Date: 26/11/2020

Programmer Name: Param Chordiya
Batch: E5
Problem Statement:

- Implement stack using linked list and perform push, pop operations on it.
- Implement queue using linked list and perform insert/enque, delete/deque operations on it.

(a) STACK USING LINKED LIST:

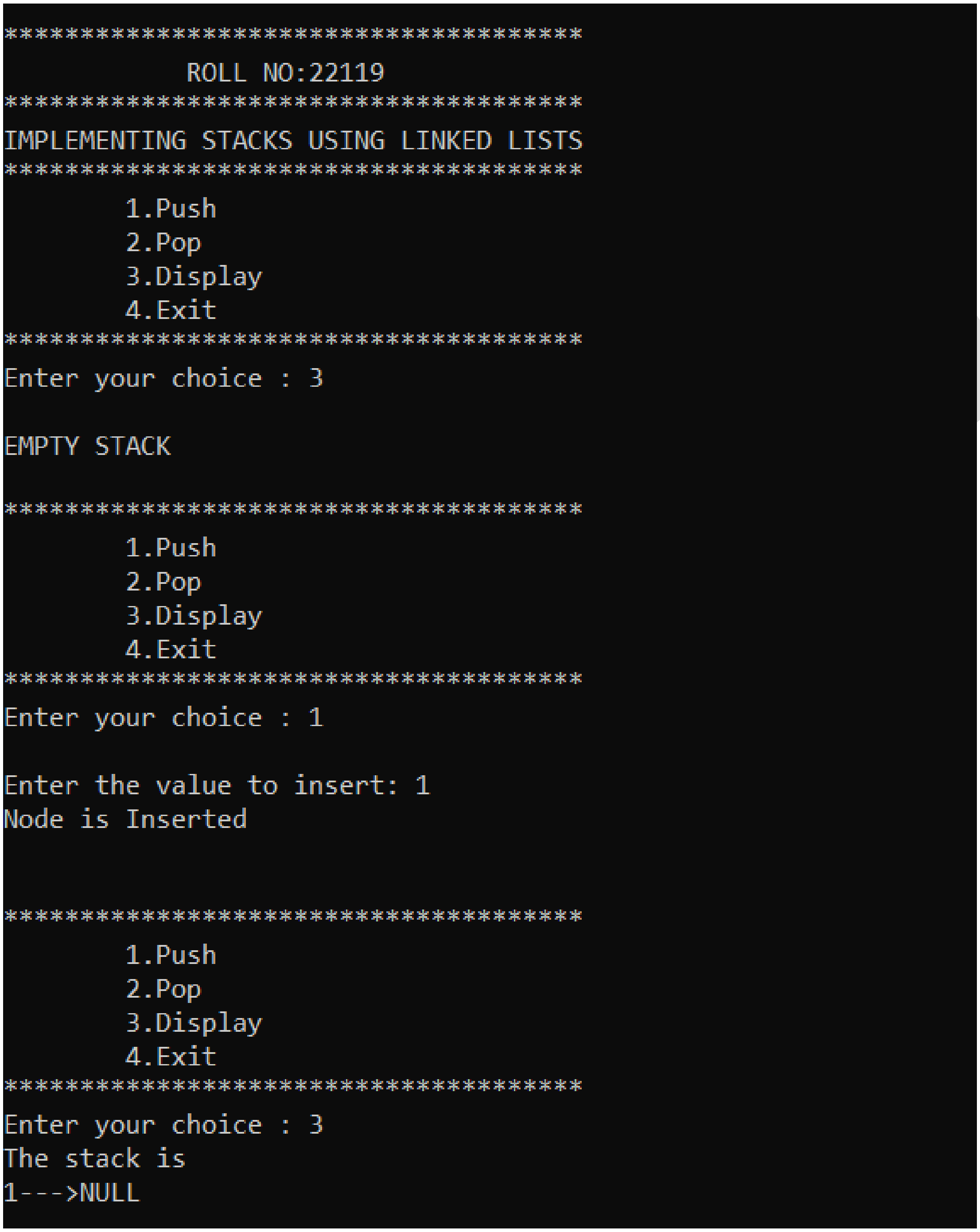
Input:

```
#include<stdio.h>
#include<stdlib.h>
struct Node
{
        int data;
        struct Node *next;
}*top = NULL;
void push(int);
void pop();
void display();
void main()
{
        printf("\n****************************************\n");
        printf("\t    ROLL NO:22119");
        printf("\n****************************************");
        printf("\nIMPLEMENTING STACKS USING LINKED LISTS");
        int choice, value;
        while(1)
        {
                printf("\n****************************************");
                printf("\n\t1.Push\n\t2.Pop\n\t3.Display\n\t4.Exit\n");
                printf("****************************************");
                printf("\nEnter your choice : ");
                scanf("%d",&choice);
                switch(choice)
                {
                        case 1:
                        printf("\nEnter the value to insert: ");
```

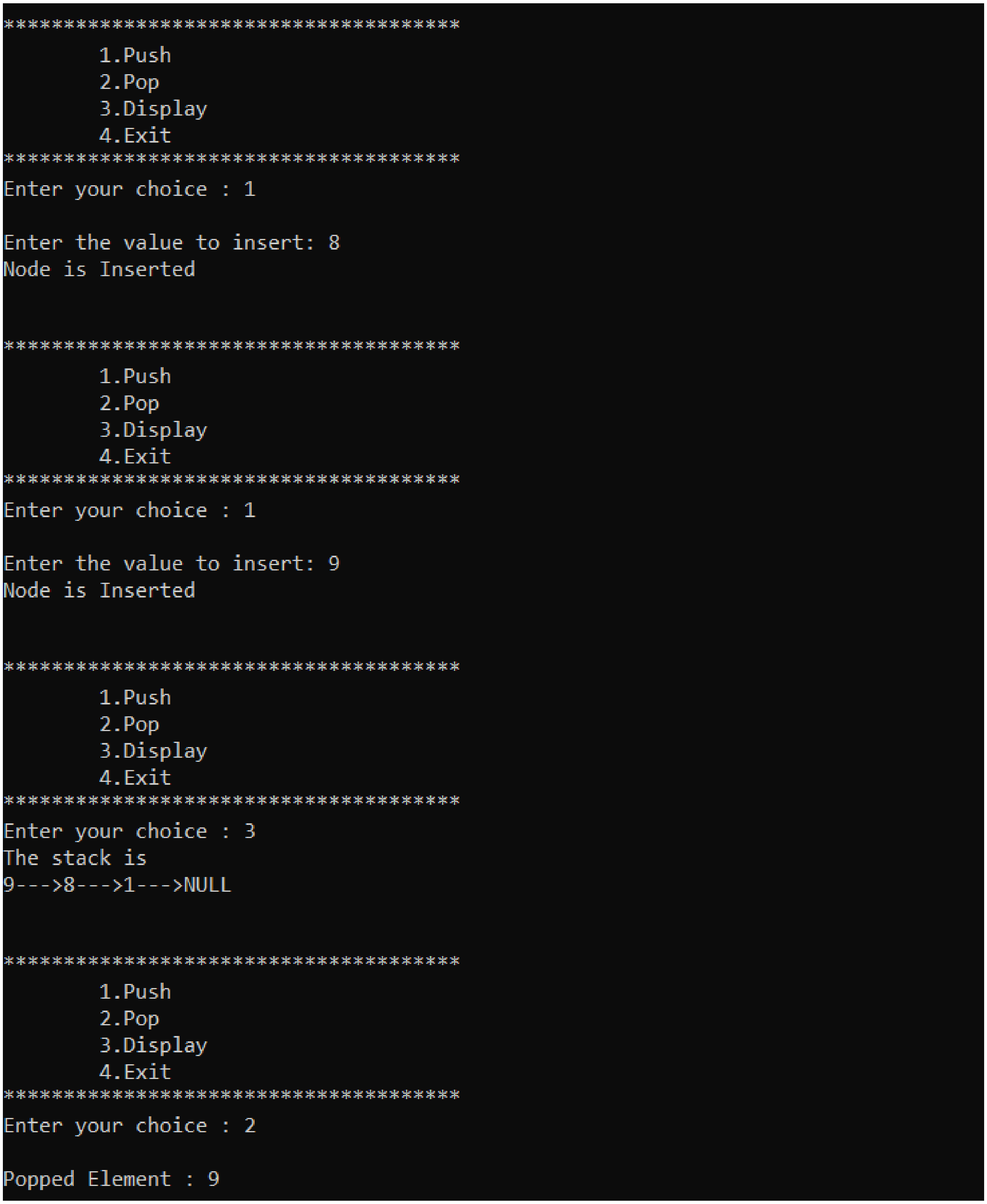
```
            scanf("%d", &value);
            push(value);
            break;
            case 2:
            pop();
            break;
            case 3:
            display();
            break;
            case 4:
            exit(0);
            break;
            default:
            printf("\nInvalid Choice\n");
        }
    }
}
void push(int value)
{
    struct Node *newNode;
    newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    if(top == NULL)
    newNode->next = NULL;
    else
    newNode->next = top;
    top = newNode;
    printf("Node is Inserted\n\n");
}
void pop()
{
    if(top == NULL)
    printf("\nEMPTY STACK\n");
    else
    {
        struct Node *temp = top;
        printf("\nPopped Element : %d", temp->data);
```

```
                printf("\n");
                top = temp->next;
                free(temp);
        }
}
void display()
{

        if(top == NULL)
        printf("\nEMPTY STACK\n");
        else
        {
                printf("The stack is \n");
                struct Node *temp = top;
                while(temp->next != NULL)
                {
                        printf("%d--->",temp->data);
                        temp = temp -> next;
                }
        printf("%d--->NULL\n\n",temp->data);
        }
}
```

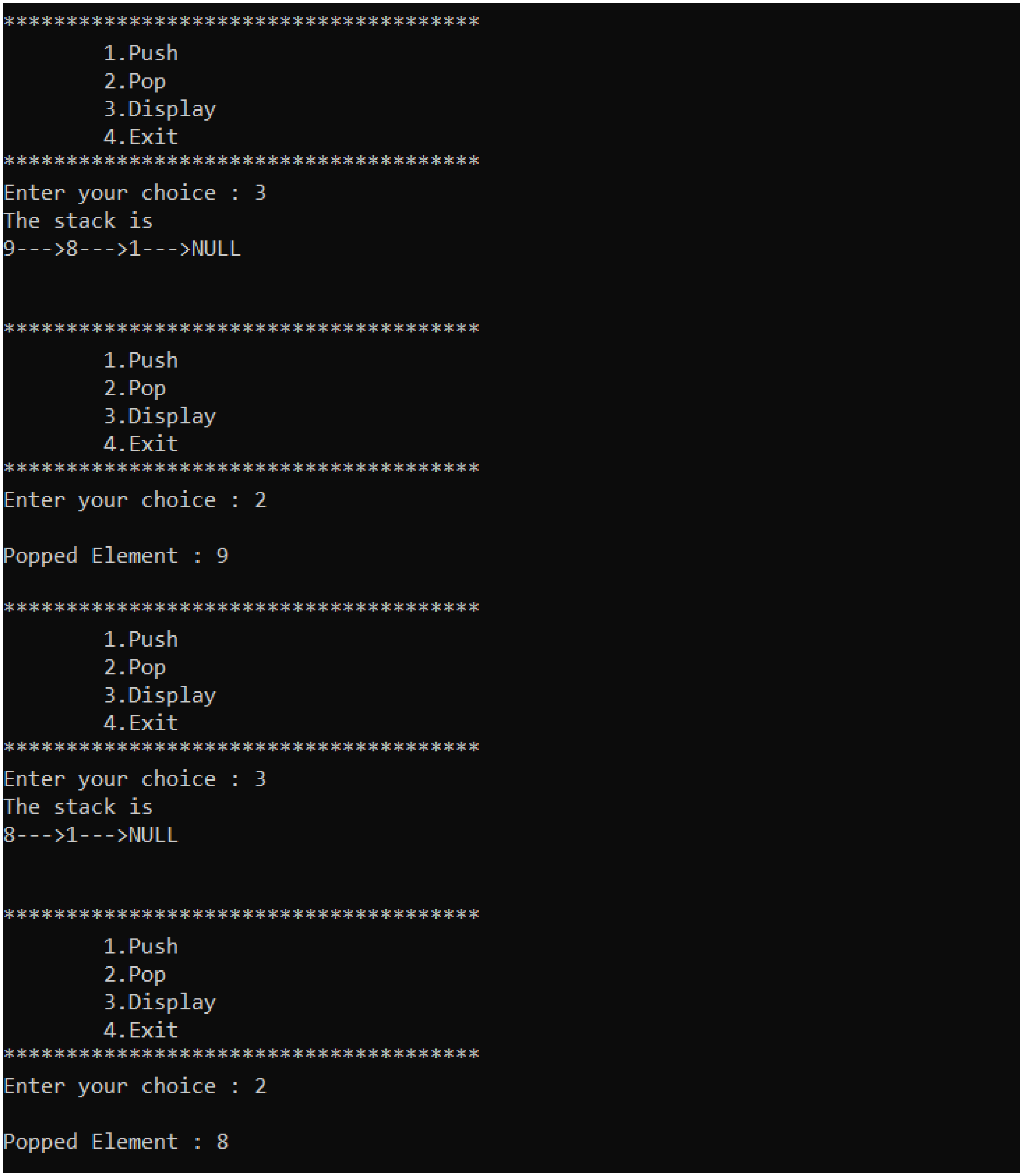
Output:



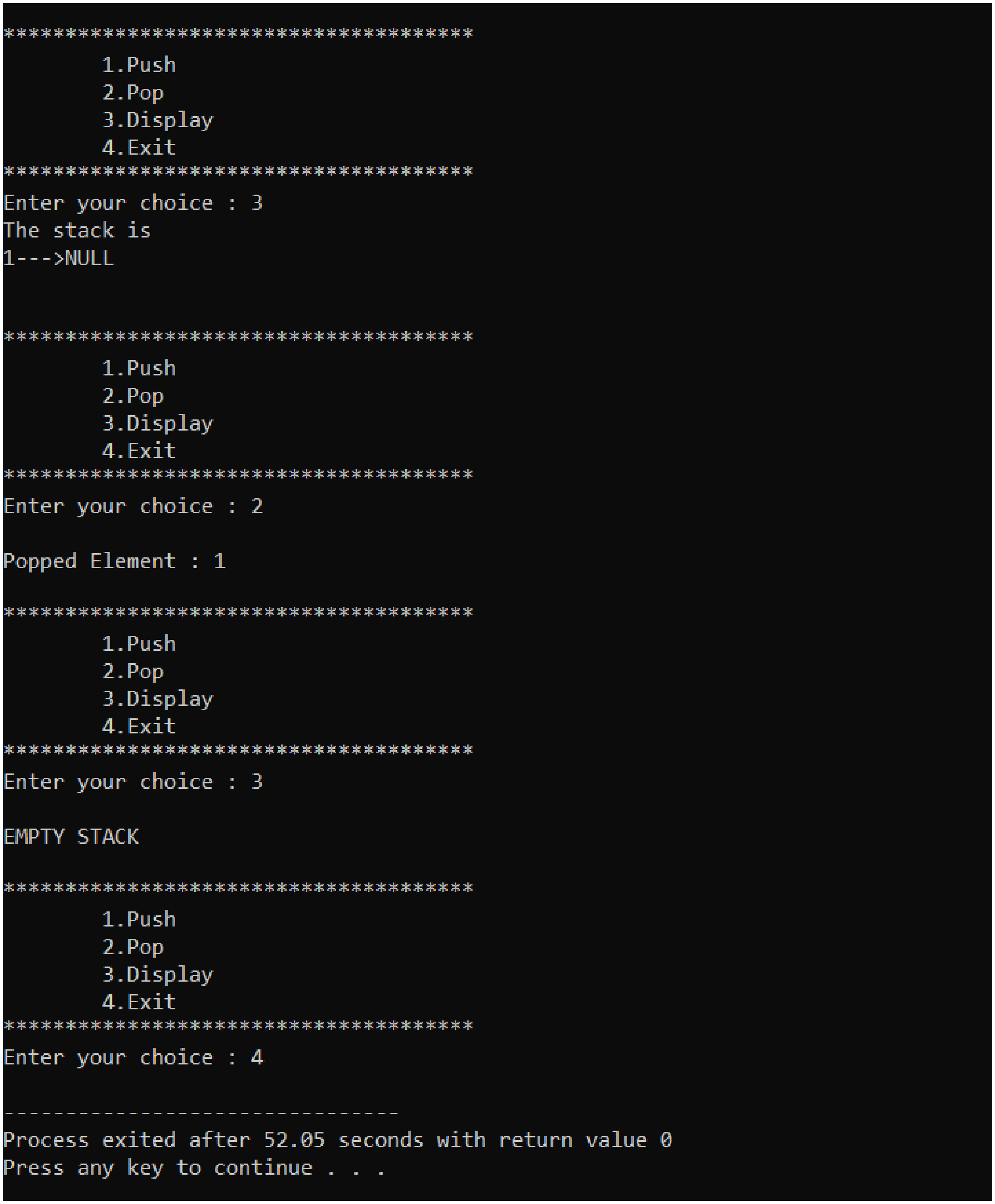
```
**************************************
            ROLL NO:22119
**************************************
IMPLEMENTING STACKS USING LINKED LISTS
**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 3

EMPTY STACK

**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 1

Enter the value to insert: 1
Node is Inserted


**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 3
The stack is
1--->NULL
```



```
**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 1

Enter the value to insert: 8
Node is Inserted


**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 1

Enter the value to insert: 9
Node is Inserted


**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 3
The stack is
9--->8--->1--->NULL


**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 2

Popped Element : 9
```



```
**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 3
The stack is
9--->8--->1--->NULL


**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 2

Popped Element : 9

**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 3
The stack is
8--->1--->NULL


**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 2

Popped Element : 8
```



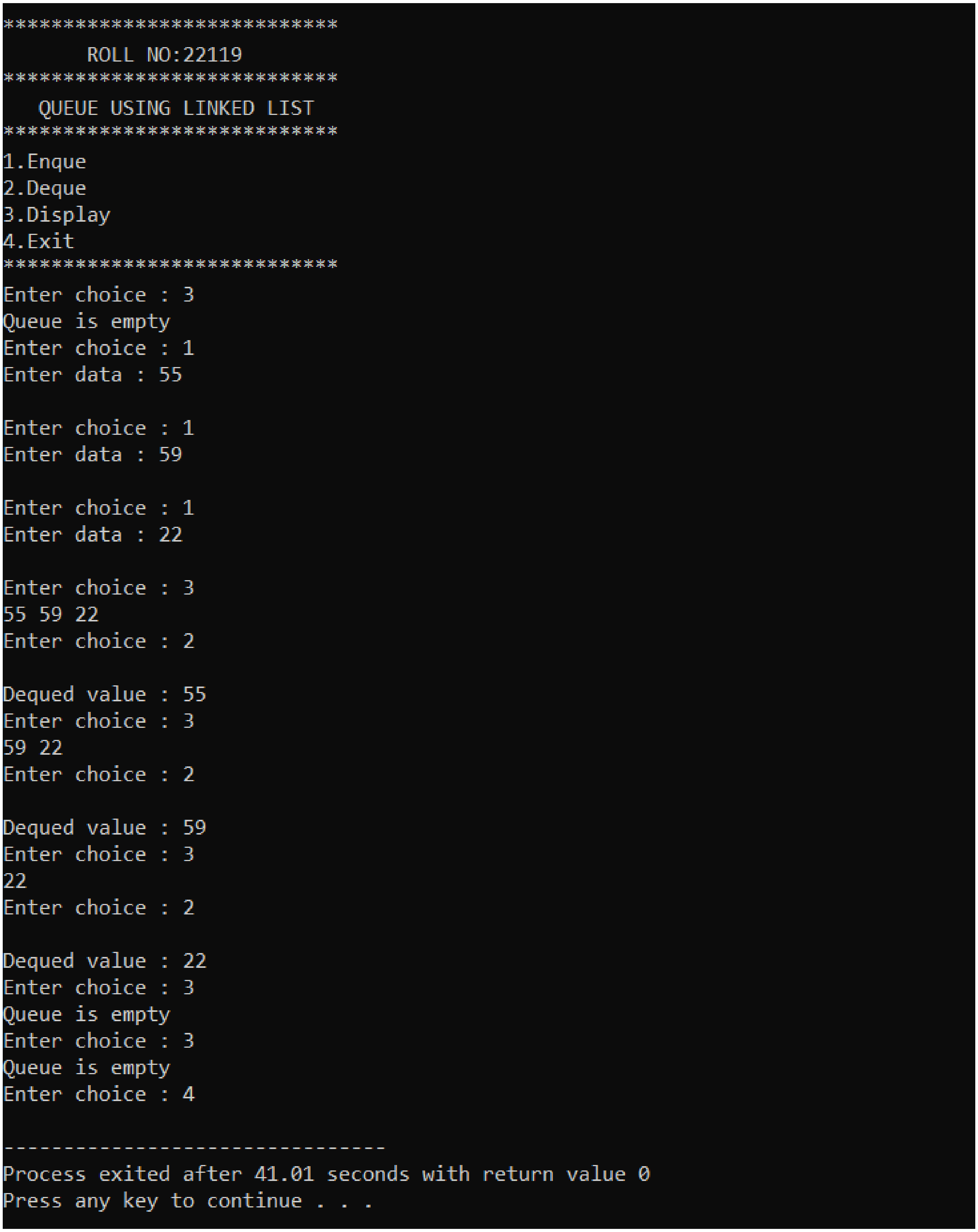
```
**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 3
The stack is
1--->NULL


**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 2

Popped Element : 1

**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 3

EMPTY STACK

**************************************
        1.Push
        2.Pop
        3.Display
        4.Exit
**************************************
Enter your choice : 4

--------------------------------
Process exited after 52.05 seconds with return value 0
Press any key to continue . . .
```

(b) QUEUE USING LINKED LIST:

Input:

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

struct node
{
   int info;
   struct node *ptr;
}*front,*rear,*temp,*front1;
void enq(int data);
void deq();
void display();
void create();
int count = 0;

void main()
{
   int no, ch, e;
   printf("\n****************************");
   printf("\n       ROLL NO:22119        ");
   printf("\n****************************\n");
   printf("   QUEUE USING LINKED LIST   ");
   printf("\n****************************");
   printf("\n1.Enque");
   printf("\n2.Deque");
   printf("\n3.Display");
   printf("\n4.Exit");
   printf("\n****************************");
   create();
                                while (1)
   {
      printf("\nEnter choice : ");
      scanf("%d", &ch);
      switch (ch)
      {
      case 1:
          printf("Enter data : ");
          scanf("%d", &no);
          enq(no);
          break;
      case 2:
          deq();
          break;
      case 3:
      display();
          break;
      case 4:
          exit(0);
      default:
          printf("Wrong choice, Please enter correct choice  ");
          break;
                                      }
                                }
}

void create()
{
   front = rear = NULL;
}

void enq(int data)
{
   if (rear == NULL)
   {
      rear = (struct node *)malloc(1*sizeof(struct node));
      rear->ptr = NULL;
      rear->info = data;
      front = rear;
   }
   else
   {
      temp=(struct node *)malloc(1*sizeof(struct node));
      rear->ptr = temp;
        temp->info = data;
        temp->ptr = NULL;

        rear = temp;
    }
    count++;
}

void display()
{
    front1 = front;

    if ((front1 == NULL) && (rear == NULL))
    {
        printf("Queue is empty");
        return;
    }
    while (front1 != rear)
    {
        printf("%d ", front1->info);
        front1 = front1->ptr;
    }
    if (front1 == rear)
        printf("%d", front1->info);
}

void deq()
{
    front1 = front;

    if (front1 == NULL)
    {
        printf("\nError: Trying to display elements from empty queue");
        return;
    }
    else
        if (front1->ptr != NULL)
        {
           front1 = front1->ptr;
           printf("\nDequed value : %d", front->info);
           free(front);
           front = front1;
        }
        else
        {
           printf("\nDequed value : %d", front->info);
           free(front);
           front = NULL;
           rear = NULL;
        }
        count--;
}
```

Output:



```
****************************
       ROLL NO:22119
****************************
   QUEUE USING LINKED LIST
****************************
1.Enque
2.Deque
3.Display
4.Exit
****************************
Enter choice : 3
Queue is empty
Enter choice : 1
Enter data : 55

Enter choice : 1
Enter data : 59

Enter choice : 1
Enter data : 22

Enter choice : 3
55 59 22
Enter choice : 2

Dequed value : 55
Enter choice : 3
59 22
Enter choice : 2

Dequed value : 59
Enter choice : 3
22
Enter choice : 2

Dequed value : 22
Enter choice : 3
Queue is empty
Enter choice : 3
Queue is empty
Enter choice : 4

--------------------------------
Process exited after 41.01 seconds with return value 0
Press any key to continue . . .
```