VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

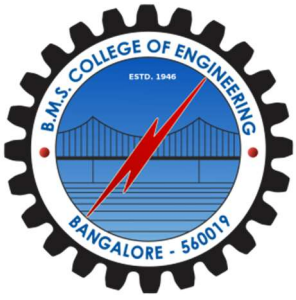
LAB REPORT
On

DATA STRUCTURES

Submitted by:

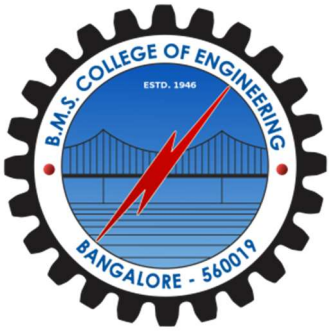
Kshitij S (1BM21CS093)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Oct 2022-Feb 2023

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled "**DATA STRUCTURES**" carried out by **Kshitij S (1BM21CS093),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of **Data structures Lab - (22CS3PCDST)** work prescribed for the said degree.

Dr. Seema Patil
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

Table of Contents

1. Course Outcomes:

CO1: Apply the concept of linear and nonlinear data structures.

CO2: Analyse data structure operations for a given problem.

CO3: Design and develop solutions using the operations of linear and nonlinear data structure for a given specification.

CO4: Conduct practical experiments for demonstrating the operations of different data structures.

2. Experiments:

2.1 Experiment: 1

2.1.1 Question:

Write a program to simulate the working of stack using an array with the following:

a) Push
b) Pop
c) Display

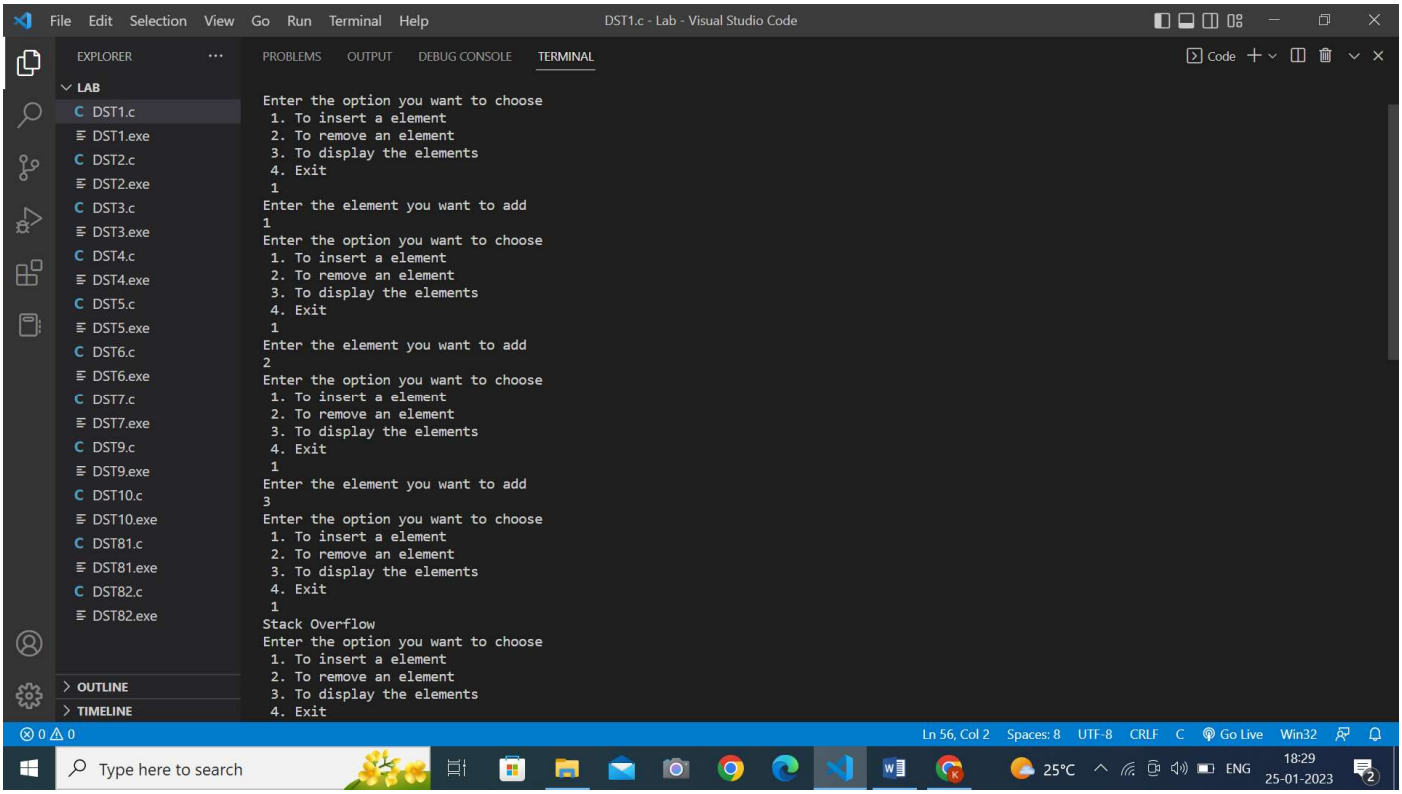
The program should print appropriate messages for stack overflow, stack underflow.

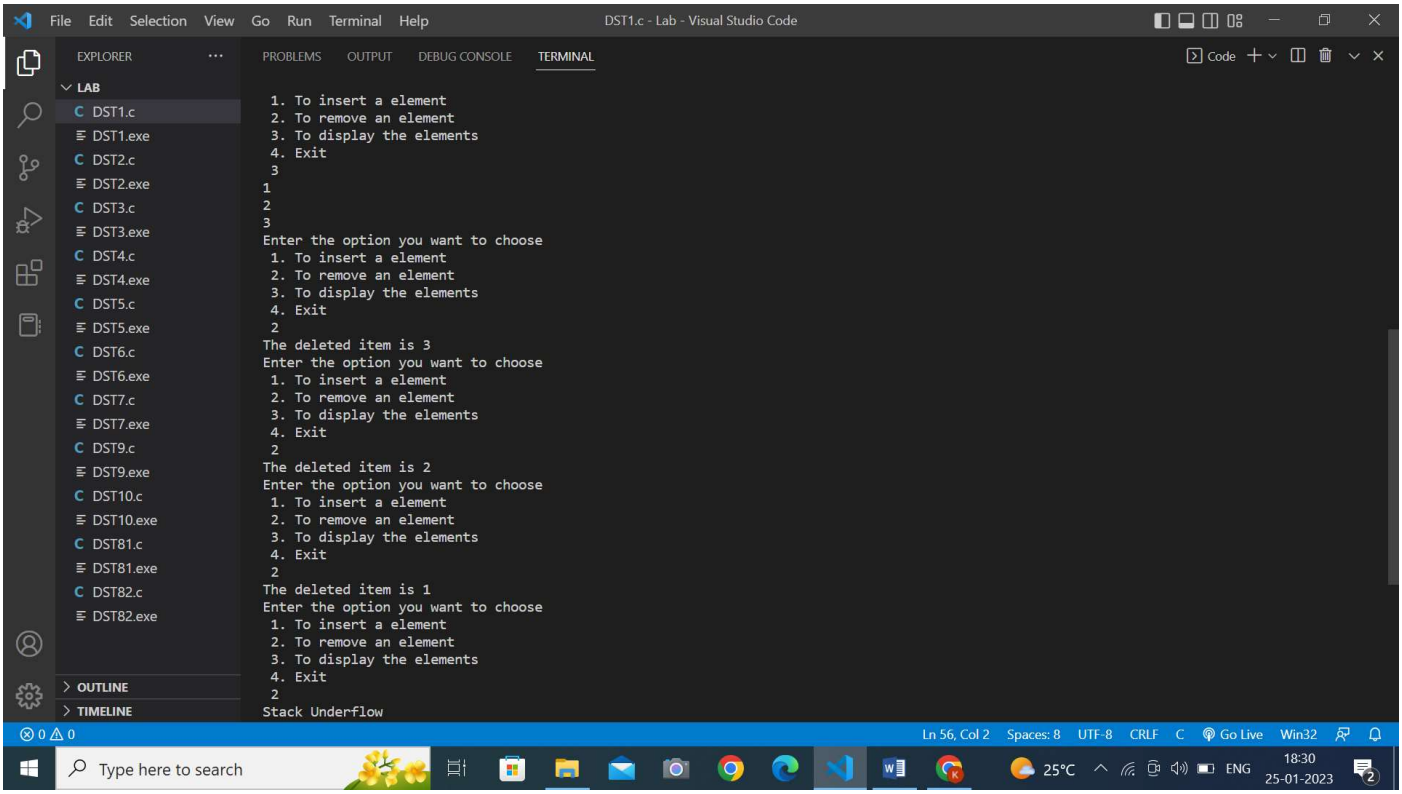
2.1.2 Code:

```
#include<stdio.h>
#include<stdlib.h>
#define size 3

void push(int st[],int *top)
{   int data;
        if(*top==(size - 1))
                printf("Stack Overflow \n");
        else
        {
            printf("Enter the element you want to add\n");
       scanf("%d",&data);
                (*top)++;
                st[*top] = data;
        }
}
int pop(int st[],int *top)
{       int n;
                if(*top==-1)
                        printf("Stack Underflow \n");
                else
                {
                        n = st[*top];
                        (*top)--;
                        return n;
```

```
                }
}
void display(int st[],int*top)
{
        if(*top == -1)
                printf("Stack Underflow,so,time to display");
        else{
   for(int i=0; i <=*top;i++)
      printf("%d \n", st[i]);
        }
}
int main()
{   int n,r,a[3],top=-1,item,del_value;
   while(1)
   {
      printf("Enter the option you want to choose \n 1. To insert a element \n 2. To remove an
element \n 3. To display the elements \n 4. Exit\n ");
      scanf("%d",&n);
      switch(n)
      {
         case 1:push(a,&top);
                break;
         case 2:del_value=pop(a,&top);
                printf("The deleted item is %d\n",del_value);
                break;
         case 3: display(a,&top);
                break;
         default:exit(0);
                break;
      }
   }
}
```

2.1.3 Output:


File Edit Selection View Go Run Terminal Help
DST1.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
DST5.c
DST5.exe
DST6.c
DST6.exe
DST7.c
DST7.exe
DST9.c
DST9.exe
DST10.c
DST10.exe
DST81.c
DST81.exe
DST82.c
DST82.exe
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Enter the option you want to choose
1. To insert a element
2. To remove an element
3. To display the elements
4. Exit
1
Enter the element you want to add
1
Enter the option you want to choose
1. To insert a element
2. To remove an element
3. To display the elements
4. Exit
1
Enter the element you want to add
2
Enter the option you want to choose
1. To insert a element
2. To remove an element
3. To display the elements
4. Exit
1
Enter the element you want to add
3
Enter the option you want to choose
1. To insert a element
2. To remove an element
3. To display the elements
4. Exit
1
Stack Overflow
Enter the option you want to choose
1. To insert a element
2. To remove an element
3. To display the elements
4. Exit
Ln 56, Col 2 Spaces: 8 UTF-8 CRLF C Go Live Win32
Type here to search
25°C
ENG
18:29
25-01-2023


File Edit Selection View Go Run Terminal Help
DST1.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
DST5.c
DST5.exe
DST6.c
DST6.exe
DST7.c
DST7.exe
DST9.c
DST9.exe
DST10.c
DST10.exe
DST81.c
DST81.exe
DST82.c
DST82.exe
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
1. To insert a element
2. To remove an element
3. To display the elements
4. Exit
3
1
2
3
Enter the option you want to choose
1. To insert a element
2. To remove an element
3. To display the elements
4. Exit
2
The deleted item is 3
Enter the option you want to choose
1. To insert a element
2. To remove an element
3. To display the elements
4. Exit
2
The deleted item is 2
Enter the option you want to choose
1. To insert a element
2. To remove an element
3. To display the elements
4. Exit
2
The deleted item is 1
Enter the option you want to choose
1. To insert a element
2. To remove an element
3. To display the elements
4. Exit
2
Stack Underflow
Ln 56, Col 2 Spaces: 8 UTF-8 CRLF C Go Live Win32
Type here to search
25°C
ENG
18:30
25-01-2023

2.2 Experiment: 2

2.2.1 Question:

WAP to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators:

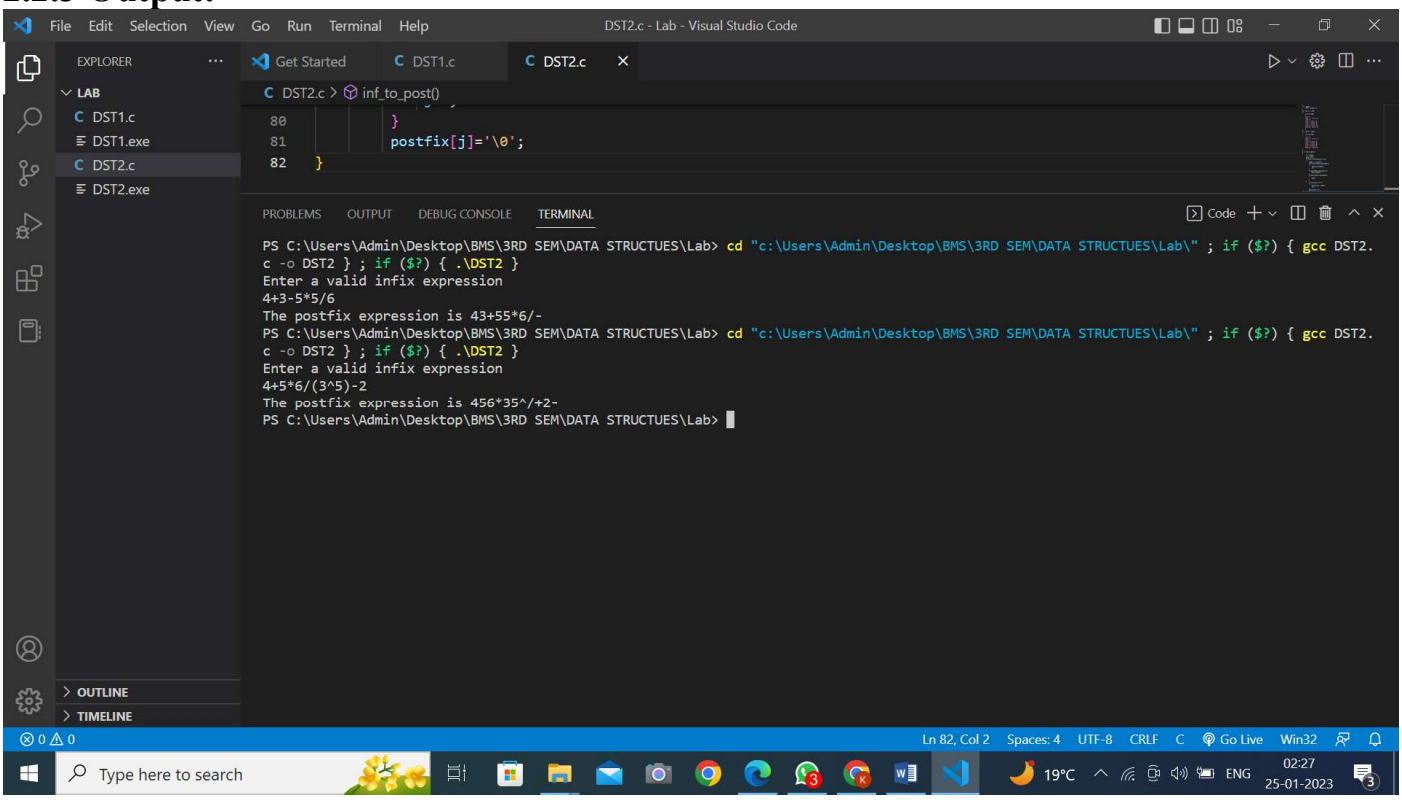
+ (plus), - (minus), * (multiply) and / (divide)

2.2.2 Code:

```
#include<stdio.h>
#include<string.h>
int top = -1;
char s[10];
char infix[20];
char postfix[20];

void inf_to_post();
int sp(char);
int ip(char);
void push(char);
char pop();

void main(){
   printf("Enter a valid infix expression \n");
   scanf("%s",infix);
   inf_to_post();
   printf("The postfix expression is %s",postfix);
}
void push(char item)
{
   s[++top]=item;
}
char pop(){
   return s[top--];
}
int sp(char item)
{
   switch(item)
   {
   case'+':
   case'-': return 2;
   case'*':
   case'/':return 4;
   case'^':return 5;
   case'(':return 0;
   case'#':return -1;
   default: return 8;
```

```
    }
}
int ip(char item)
{
    switch(item)
{
    case'+':
    case'-': return 1;
    case'*':
    case'/':return 3;
    case'^':return 6;
    case'(':return 9;
    case')':return 0;
    default: return 7;
}
}
void inf_to_post()
{
     int i,j=0;
     char symbol;
     push('#');
     for(i=0;i<strlen(infix);i++)
     {
        symbol = infix[i];
        while(sp(s[top])>ip(symbol))
        {
           postfix[j]=pop();
           j++;
        }
        if(sp(s[top])<ip(symbol)){
           push(symbol);
        }
        if(sp(s[top])==ip(symbol))
        {
           pop();
        }
     }
        while(s[top] !='#')
        {
           postfix[j] = pop();
           j++;
        }
        postfix[j]='\0';
}
```

2.2.3 Output:


File Edit Selection View Go Run Terminal Help
DST2.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
Get Started
DST1.c
DST2.c
DST2.c > inf_to_post()
}
postfix[j]='\0';
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab> cd "c:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab\" ; if ($?) { gcc DST2.c -o DST2 } ; if ($?) { .\DST2 }
Enter a valid infix expression
4+3-5*5/6
The postfix expression is 43+55*6/-
PS C:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab> cd "c:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab\" ; if ($?) { gcc DST2.c -o DST2 } ; if ($?) { .\DST2 }
Enter a valid infix expression
4+5*6/(3^5)-2
The postfix expression is 456*35^/+2-
PS C:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab>
OUTLINE
TIMELINE
Ln 82, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32
Type here to search
19°C
ENG
02:27
25-01-2023

2.3 Experiment: 3

2.3.1 Question:

WAP to simulate the working of a queue of integers using an array. Provide the following operations:

a) Insert
b) Delete
c) Display

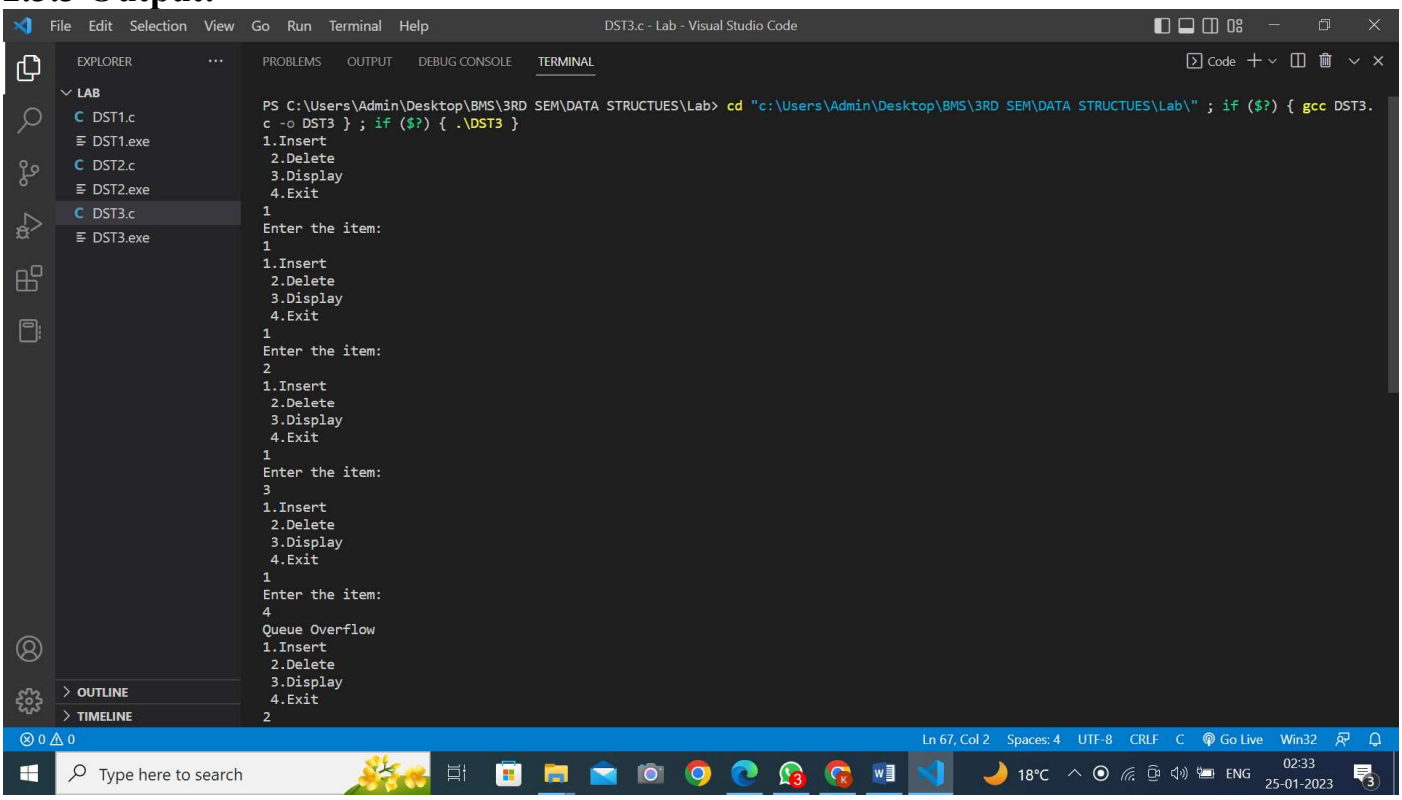
The program should print appropriate messages for queue empty and queue overflow conditions.

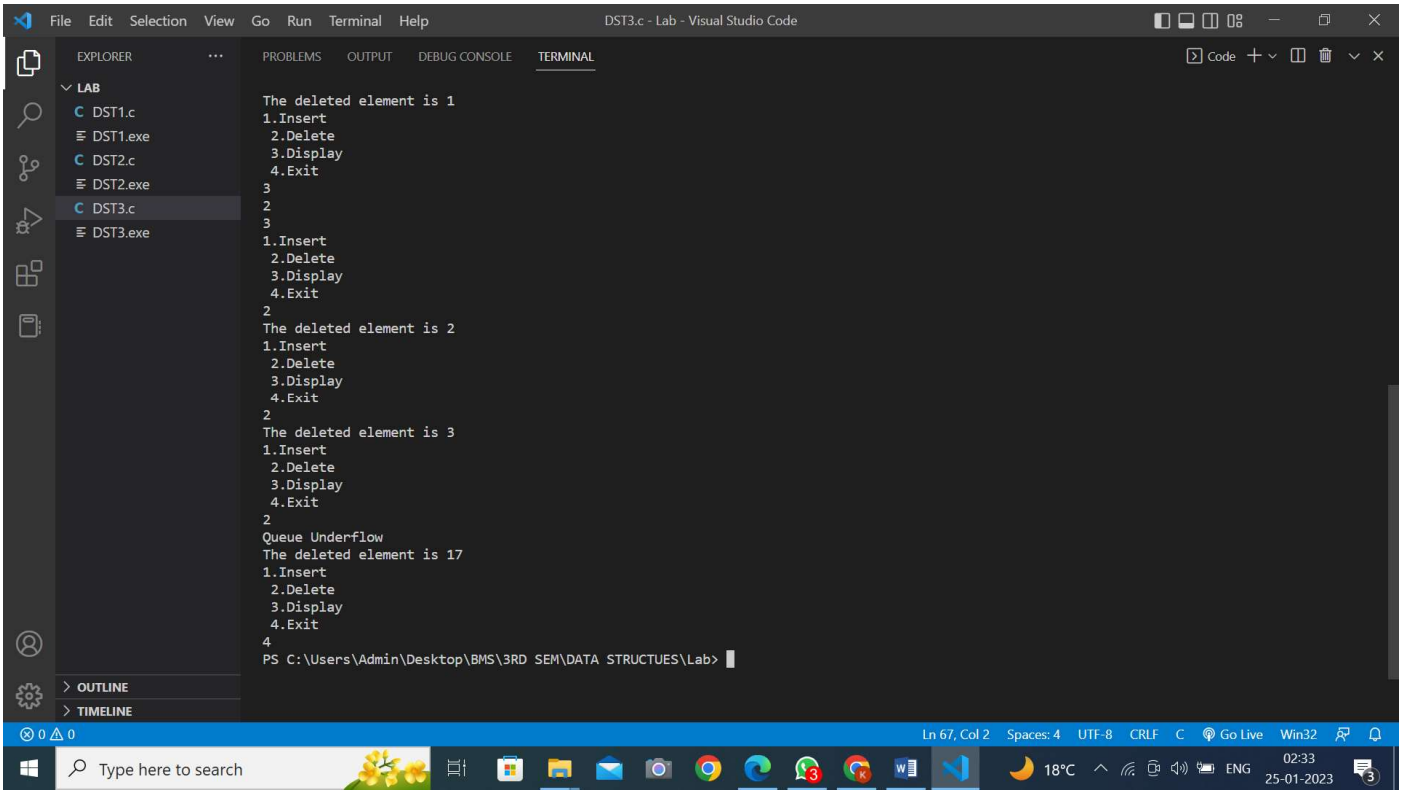
2.3.2 Code:

```
#include <stdio.h>
#include <stdlib.h>
#define size 3
void insert_rear(int q[],int *r,int item)
{
   if((*r)==(size - 1))
   {
      printf("Queue Overflow\n");
   }
   else
   {
      (*r)++;
      q[*r] = item;
   }
}
int delete_front(int q[],int *f,int *r)
{
   //int del_item;
   if((*f)>(*r))
   {
      printf("Queue Underflow \n");
   }
   else
   {
      //del_item = q[*f];
      //(*f)++;
      return q[(*f)++];
   }
}
void display(int q[],int *f,int *r)
{
   int i;
   if((*f)>(*r))
   {
```

```
        printf("Queue Underflow \n");
    }
    else
    {
        for(i = *f;i<=*r;i++)
        {
            printf("%d \n",q[i]);
        }
    }
}

void main()
{
    int a,item,q[3],r=-1,f=0,del;
    while(1)
    {
    printf("1.Insert \n 2.Delete \n 3.Display \n 4.Exit \n");
    scanf("%d",&a);
    switch(a)
    {
        case 1 : printf("Enter the item: \n");
                 scanf("%d",&item);
                 insert_rear(q,&r,item);
                 break;
        case 2 : del = delete_front(q,&f,&r);
                 printf("The deleted element is %d \n",del);
                 break;
        case 3 : display(q,&f,&r);
                 break;
        default: exit(0);
    }
    }
}   }
}
```

2.3.3 Output:


File Edit Selection View Go Run Terminal Help
DST3.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
PS C:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab> cd "c:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab\" ; if ($?) { gcc DST3.c -o DST3 } ; if ($?) { .\DST3 }
1.Insert
2.Delete
3.Display
4.Exit
1
Enter the item:
1
1.Insert
2.Delete
3.Display
4.Exit
1
Enter the item:
2
1.Insert
2.Delete
3.Display
4.Exit
1
Enter the item:
3
1.Insert
2.Delete
3.Display
4.Exit
1
Enter the item:
4
Queue Overflow
1.Insert
2.Delete
3.Display
4.Exit
2
OUTLINE
TIMELINE
Ln 67, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32
Type here to search
18°C
ENG
02:33
25-01-2023


File Edit Selection View Go Run Terminal Help
DST3.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
The deleted element is 1
1.Insert
2.Delete
3.Display
4.Exit
3
2
3
1.Insert
2.Delete
3.Display
4.Exit
2
The deleted element is 2
1.Insert
2.Delete
3.Display
4.Exit
2
The deleted element is 3
1.Insert
2.Delete
3.Display
4.Exit
2
Queue Underflow
The deleted element is 17
1.Insert
2.Delete
3.Display
4.Exit
4
PS C:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab>
OUTLINE
TIMELINE
Ln 67, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32
Type here to search
18°C
ENG
02:33
25-01-2023

2.4 Experiment: 4

2.4.1 Question:

WAP to simulate the working of a circular queue of integers using an array. Provide the following operations:

a) Insert
b) Delete
c) Display

The program should print appropriate messages for queue empty and queue overflow conditions.

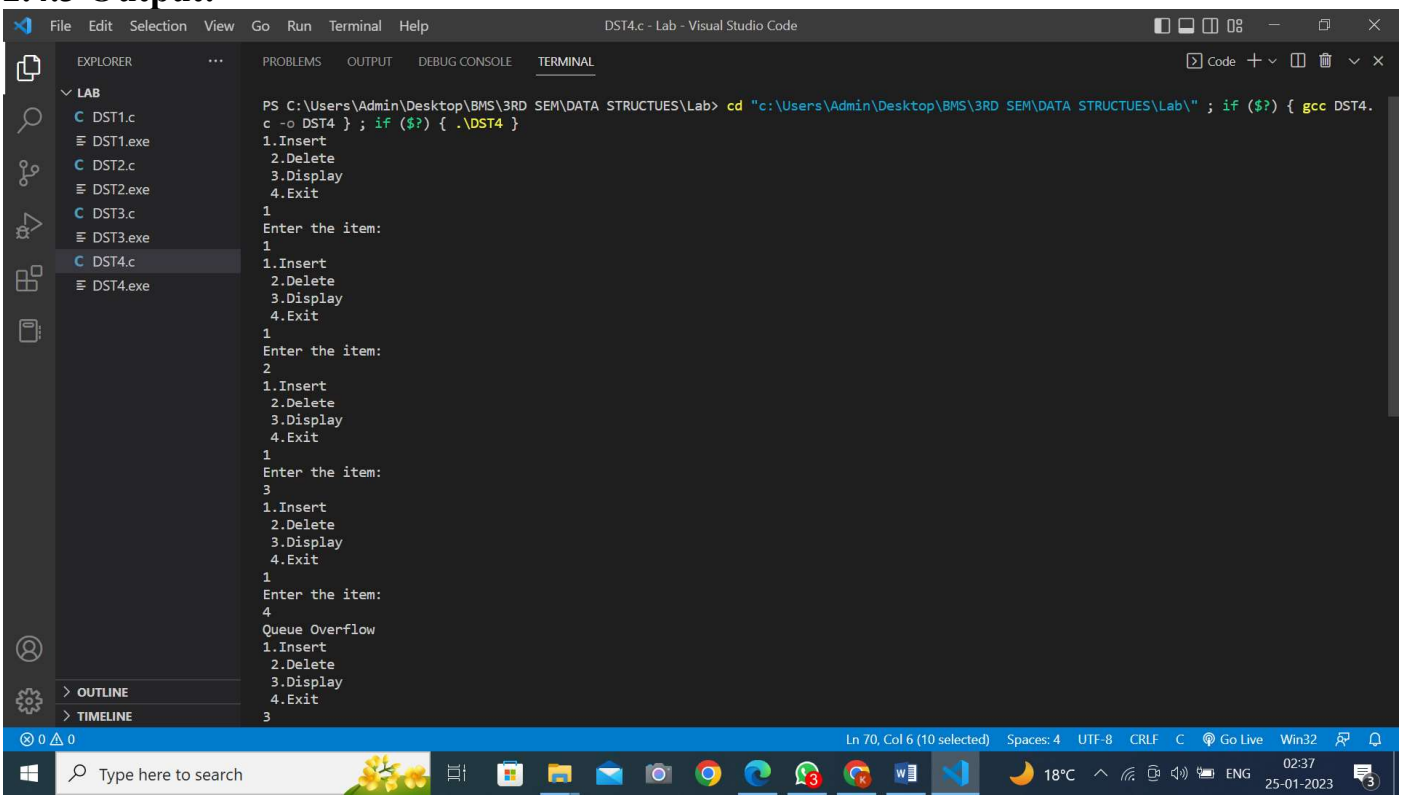
2.4.2 Code:

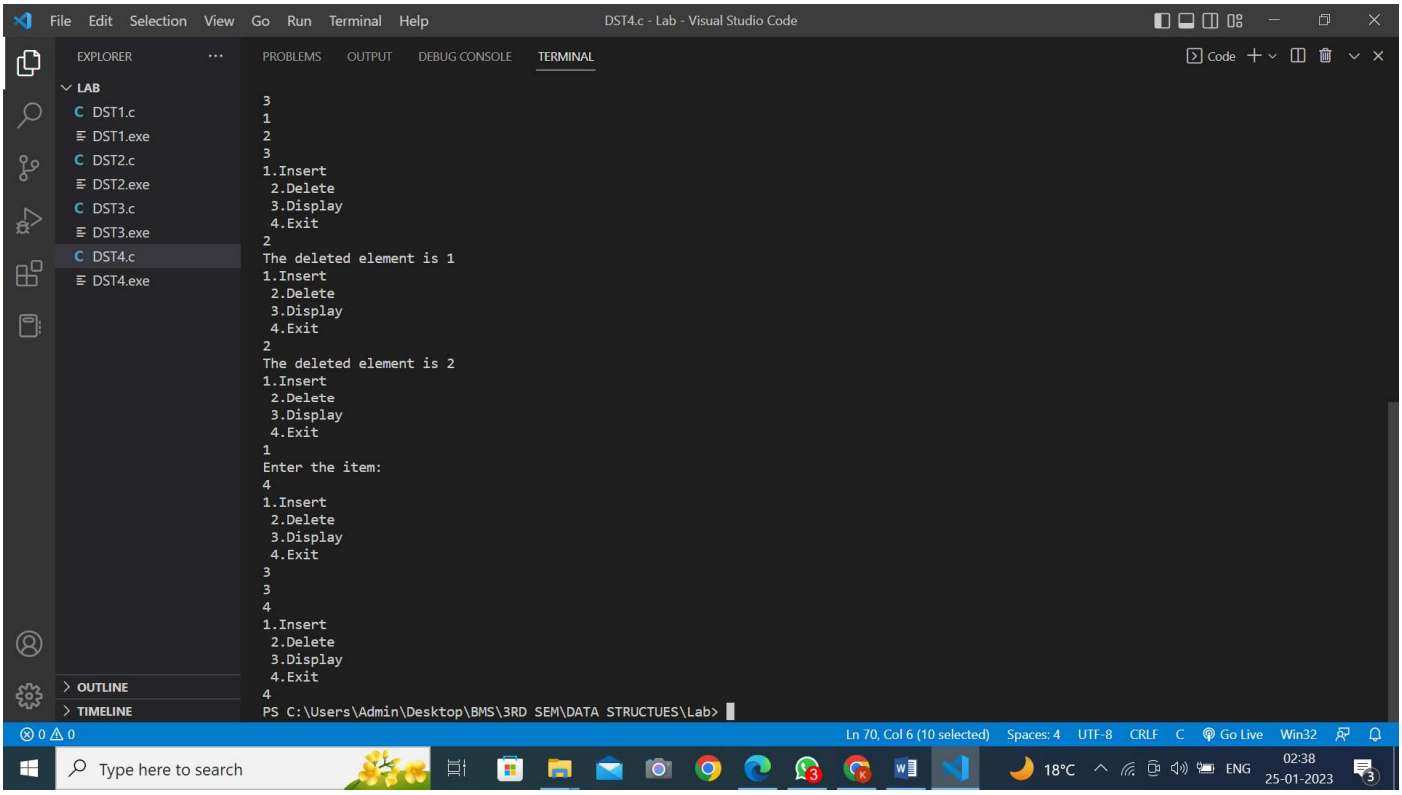
```
#include <stdio.h>

#include <stdlib.h>

#define Qsize 3

int count = 0;

void insert(int q[],int item,int *rear)

{

   if(count == Qsize)

   {

      printf("Queue Overflow\n");

   }

   else

   {

      (*rear)++;

      (*rear) = (*rear) % Qsize;

      q[*rear] = item;

      count++;

   }

}

int delete(int q[],int *front,int *rear)

{

   int del_item;

   if(count==0)
```

```
    {
        printf("Queue Underflow \n");
    }
    else
    {
        del_item = q[*front];
        (*front) = (++(*front)) % Qsize;
        count--;
        return del_item;
    }
}
void display(int q[],int *front)
{
    int temp,i;
    temp = *front;
    if(count == 0)
    {
        printf("Queue Underflow \n");
    }
    else
    {
        for(i = 0;i<count;i++)
        {
            printf("%d \n",q[temp]);
            temp = (temp+1) % Qsize;
        }
    }
}
void main()
{
```

```
    int a,item,q[3],r=-1,f=0,del;
    while(1)
    {
    printf("1.Insert \n 2.Delete \n 3.Display \n 4.Exit \n");
    scanf("%d",&a);
    switch(a)
    {
       case 1 : printf("Enter the item: \n");
                scanf("%d",&item);
                insert(q,item,&r);
                break;
       case 2 : del = delete(q,&f,&r);
                printf("The deleted element is %d \n",del);
                break;
       case 3 : display(q,&f);
                break;
       default: exit(0);
    }
    }
}
```

2.4.3 Output:


File Edit Selection View Go Run Terminal Help
DST4.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
PS C:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab> cd "c:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab\" ; if ($?) { gcc DST4.c -o DST4 } ; if ($?) { .\DST4 }
1.Insert
2.Delete
3.Display
4.Exit
1
Enter the item:
1
1.Insert
2.Delete
3.Display
4.Exit
1
Enter the item:
2
1.Insert
2.Delete
3.Display
4.Exit
1
Enter the item:
3
1.Insert
2.Delete
3.Display
4.Exit
1
Enter the item:
4
Queue Overflow
1.Insert
2.Delete
3.Display
4.Exit
3
Ln 70, Col 6 (10 selected) Spaces: 4 UTF-8 CRLF C Go Live Win32
Type here to search
18°C ENG 02:37 25-01-2023


File Edit Selection View Go Run Terminal Help
DST4.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
3
1
2
3
1.Insert
2.Delete
3.Display
4.Exit
2
The deleted element is 1
1.Insert
2.Delete
3.Display
4.Exit
2
The deleted element is 2
1.Insert
2.Delete
3.Display
4.Exit
1
Enter the item:
4
1.Insert
2.Delete
3.Display
4.Exit
3
3
4
1.Insert
2.Delete
3.Display
4.Exit
4
PS C:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab>
Ln 70, Col 6 (10 selected) Spaces: 4 UTF-8 CRLF C Go Live Win32
Type here to search
18°C ENG 02:38 25-01-2023

2.5 Experiment: 5

2.5.1 Question:

WAP to Implement Singly Linked List with following operations:

a) Create a linked list.
b) Insertion of a node at first position, at any position and at end of list.
c) Display the contents of the linked list.

2.5.2 Code:

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
   int value;
   struct node *next;
};
typedef struct node *NODE;
NODE getnode()
{
   NODE temp;
   temp = (NODE)malloc(sizeof(struct node));
   if (temp == NULL)
   {
      printf("Memory not allocated");
      return NULL;
   }
   return temp;
}
NODE insert_beg(NODE first, int item)
{
   NODE new;
   new = getnode();
   new->value = item;
```

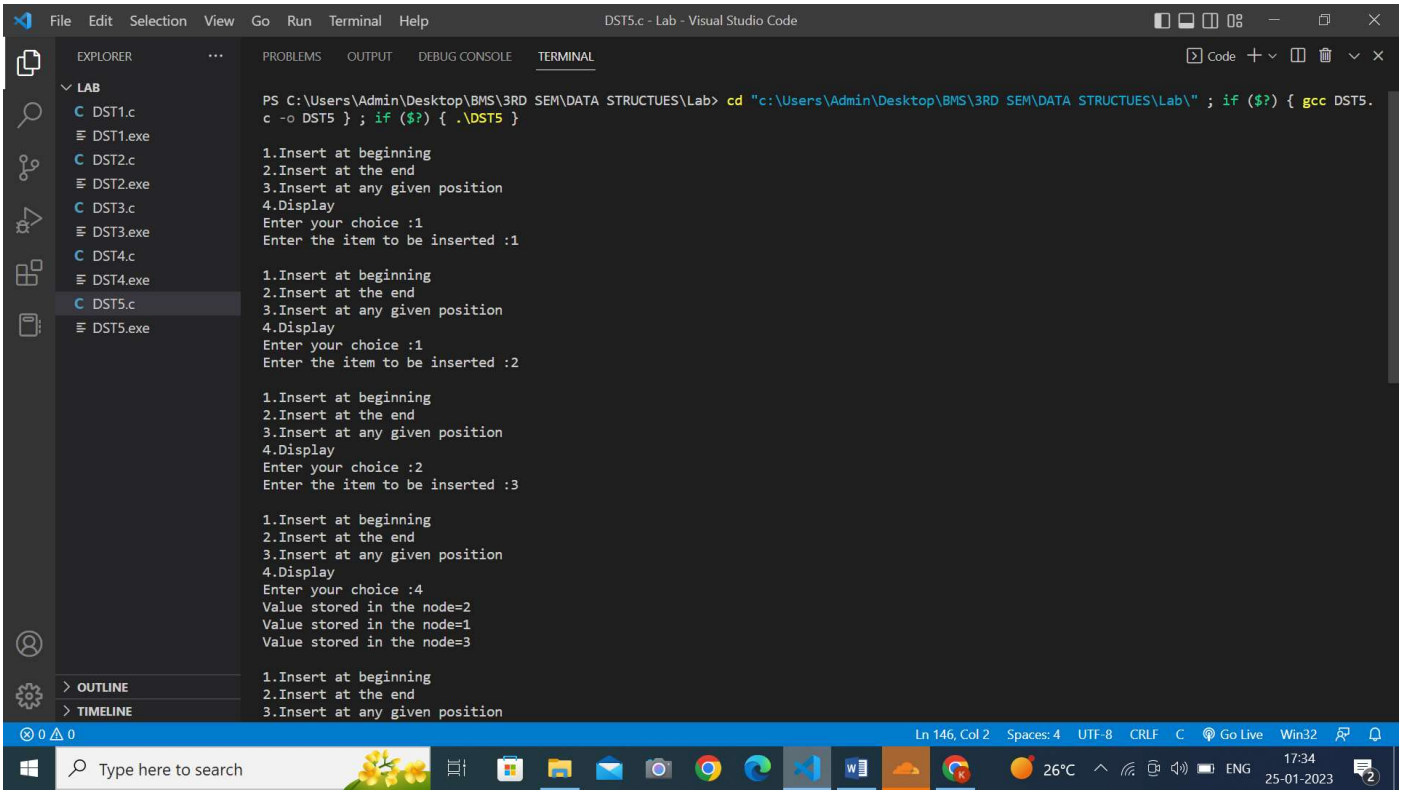
```
    new->next = NULL;
    if (first == NULL)
    {
        return new;
    }
    else
    {
        new->next = first;
        first = new;
        return first;
    }
}
NODE insert_end(NODE first, int item)
{
    NODE new, last;
    new = getnode();
    new->value = item;
    new->next = NULL;
    if (first == NULL)
    {
        return new;
    }
    if (first == NULL)
    {
        first->next = new;
        return first;
    }
    last = first;
    while (last->next != NULL)
    {
```

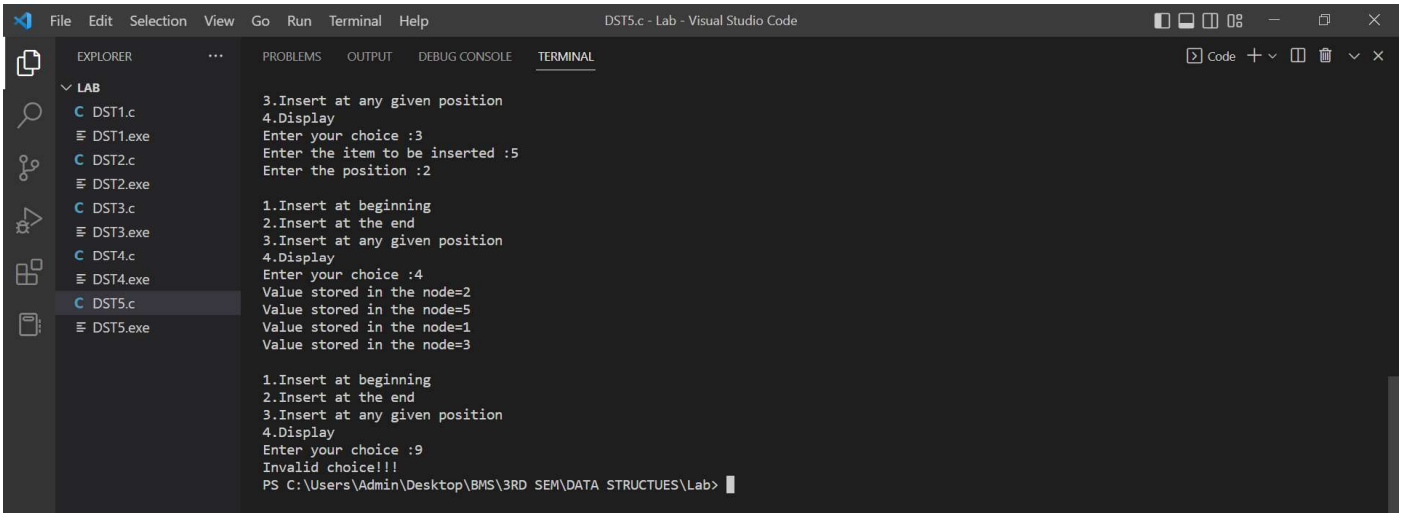
```
        last = last->next;
    }
    last->next = new;
    return first;
}
NODE insert_pos(NODE first, int item, int pos)
{
    int count = 1;
    int val = item;
    NODE new, prev, curr;
    new = getnode();
    new->value = item;
    new->next = NULL;
    curr = first;
    if (first == NULL && pos == 1)
        return new;
    while (count != pos && curr != NULL)
    {
        prev = curr;
        curr = curr->next;
        count++;
    }
    if (count == pos)
    {
        prev->next = new;
        new->next = curr;
        return first;
    }
    if (curr == NULL)
    {
```

```
        printf("Position not found\n");
        return first;
    }
    if (first != NULL && pos == 1)
    {
        first = insert_beg(first, val);
        return first;
    }
}
void display(NODE first)
{
    NODE temp;
    temp = first;
    if (first == NULL)
    {
        printf("List is empty\n");
    }
    while (temp != NULL)
    {
        printf("Value stored in the node=%d\n", temp->value);
        temp = temp->next;
    }
}
int main()
{
    int pos, item, c;
    NODE first = NULL;
    while (1) {
        printf("\n1.Insert at beginning\n2.Insert at the end\n3.Insert at any given
position\n4.Display\n");
```

```
        printf("Enter your choice :");
        scanf("%d", &c);
        switch (c)
        {
        case 1:
            printf("Enter the item to be inserted :");
            scanf("%d", &item);
            first = insert_beg(first, item);
            break;
        case 2:
            printf("Enter the item to be inserted :");
            scanf("%d", &item);
            first = insert_end(first, item);
            break;
        case 3:
            printf("Enter the item to be inserted :");
            scanf("%d", &item);
            printf("Enter the position :");
            scanf("%d", &pos);
            first = insert_pos(first, item, pos);
            break;
        case 4:
            display(first);
            break;
        default:
            printf("Invalid choice!!!");
            exit(0);
        }
    }
}
```

2.5.3 Output:


File Edit Selection View Go Run Terminal Help
DST5.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
DST5.c
DST5.exe
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
PS C:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab> cd "c:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab\" ; if ($?) { gcc DST5.c -o DST5 } ; if ($?) { .\DST5 }
1.Insert at beginning
2.Insert at the end
3.Insert at any given position
4.Display
Enter your choice :1
Enter the item to be inserted :1
1.Insert at beginning
2.Insert at the end
3.Insert at any given position
4.Display
Enter your choice :1
Enter the item to be inserted :2
1.Insert at beginning
2.Insert at the end
3.Insert at any given position
4.Display
Enter your choice :2
Enter the item to be inserted :3
1.Insert at beginning
2.Insert at the end
3.Insert at any given position
4.Display
Enter your choice :4
Value stored in the node=2
Value stored in the node=1
Value stored in the node=3
1.Insert at beginning
2.Insert at the end
3.Insert at any given position
Ln 146, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32
Type here to search
26°C
ENG
17:34
25-01-2023


File Edit Selection View Go Run Terminal Help
DST5.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
DST5.c
DST5.exe
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
3.Insert at any given position
4.Display
Enter your choice :3
Enter the item to be inserted :5
Enter the position :2
1.Insert at beginning
2.Insert at the end
3.Insert at any given position
4.Display
Enter your choice :4
Value stored in the node=2
Value stored in the node=5
Value stored in the node=1
Value stored in the node=3
1.Insert at beginning
2.Insert at the end
3.Insert at any given position
4.Display
Enter your choice :9
Invalid choice!!!
PS C:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab>

2.6 Experiment: 6

2.6.1 Question:

WAP to Implement Singly Linked List with following operations:

a) Create a linked list.
b) Deletion of first element, specified element and last element in the list.
c) Display the contents of the linked list.

2.6.2 Code:

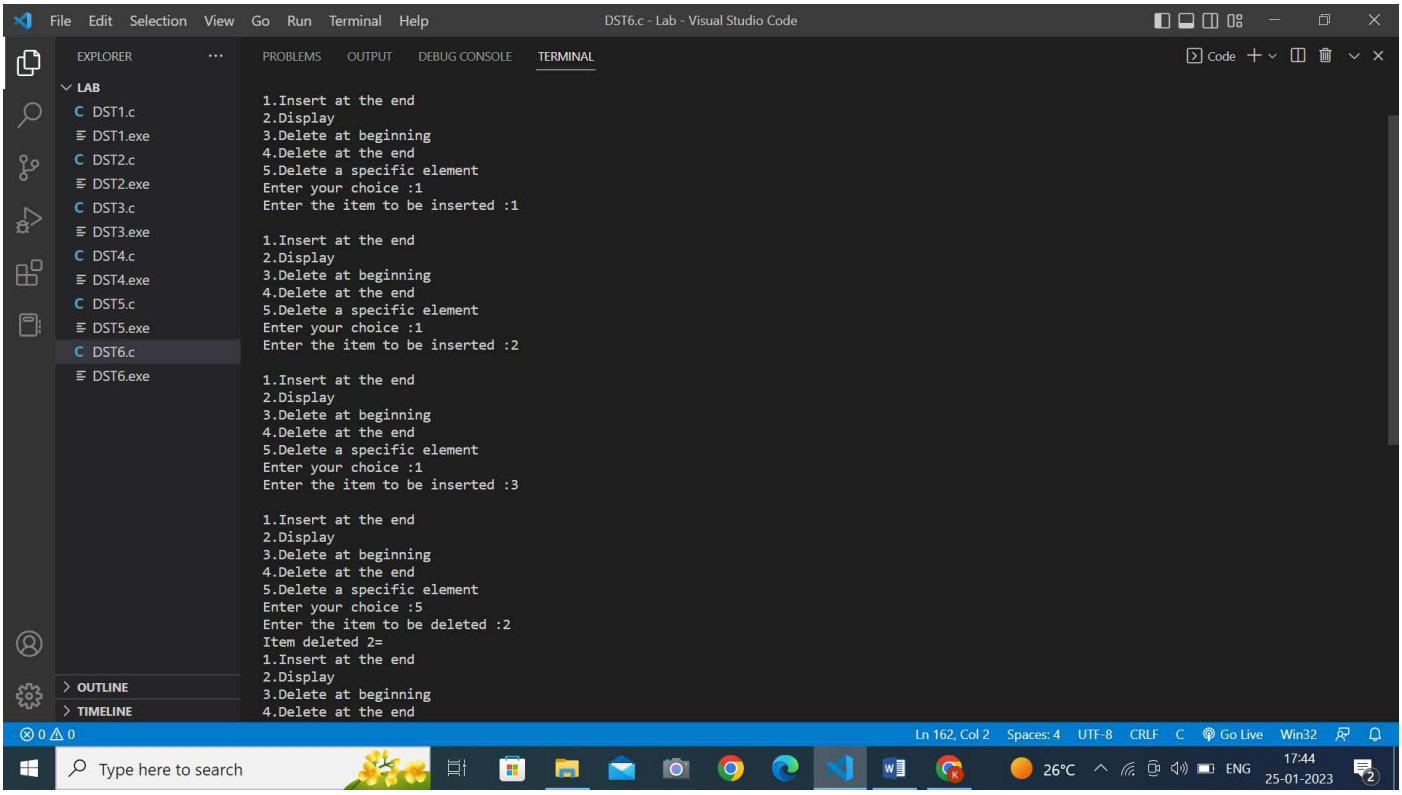
```
#include <stdio.h>
#include <stdlib.h>
struct node
{
   int value;
   struct node *next;
};
typedef struct node *NODE;
NODE getnode()
{
   NODE temp;
   temp = (NODE)malloc(sizeof(struct node));
   if (temp == NULL)
   {
      printf("Memory not allocated");
      return NULL;
   }
   return temp;
}
NODE insert_end(NODE first, int item)
{
   NODE new, last;
   new = getnode();
   new->value = item;
   new->next = NULL;
   if (first == NULL)
   {
      return new;
   }
   if (first == NULL)
   {
      first->next = new;
      return first;
   }
   last = first;
   while (last->next != NULL)
   {
```

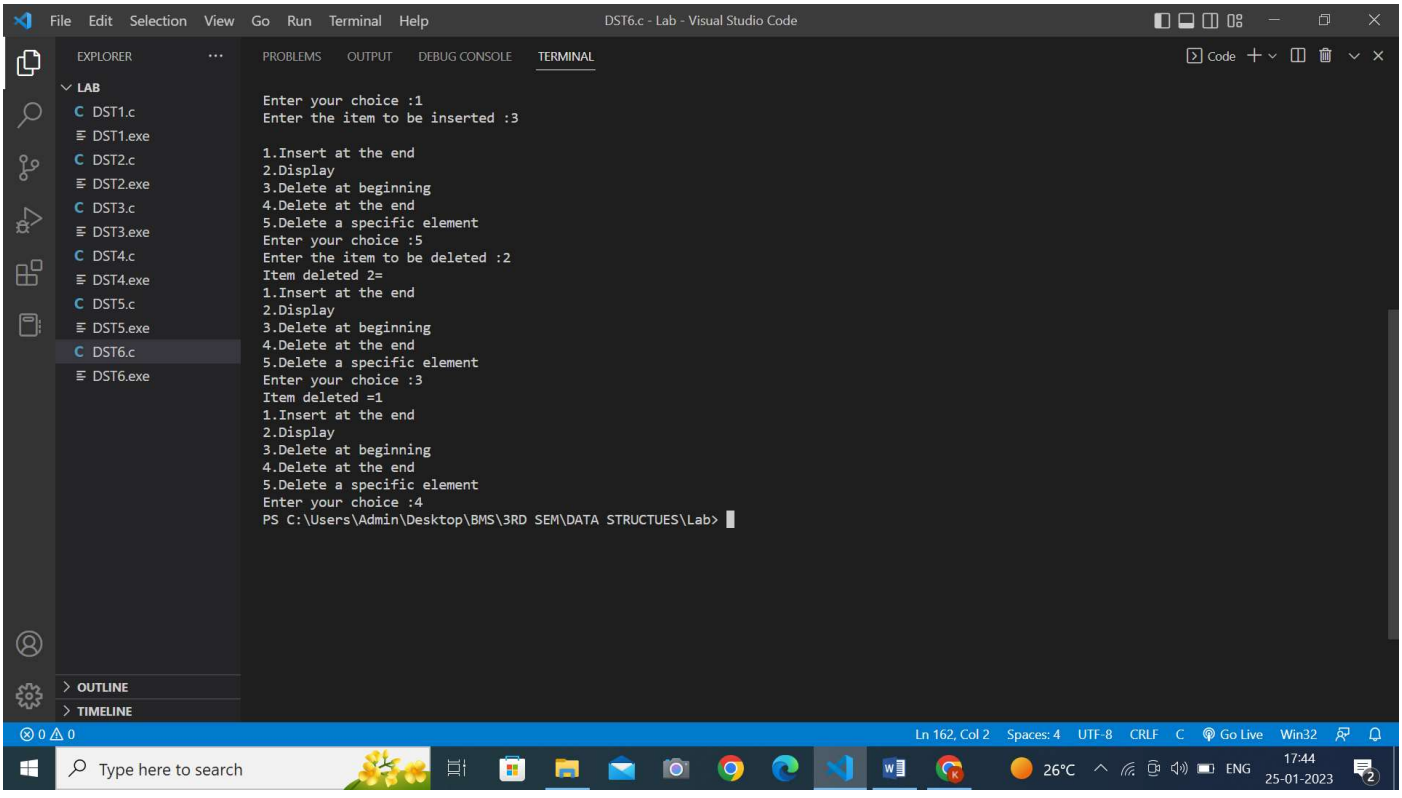
```
        last = last->next;
    }
    last->next = new;
    return first;
}
void display(NODE first)
{
    NODE temp;
    temp = first;
    if (first == NULL)
    {
        printf("List is empty\n");
    }
    while (temp != NULL)
    {
        printf("Value stored in the node=%d\n", temp->value);
        temp = temp->next;
    }
}
NODE del_beg(NODE first)
{
    NODE temp;
    if (first == NULL)
    {
        printf("list is empty\n");
        return first;
    }
    temp = first;
    temp = temp->next;
    printf("Item deleted =%d", first->value);
    free(first);
    return temp;
}
NODE del_end(NODE first)
{
    NODE curr, prev;
    if (first == NULL)
    {
        printf("list is empty\n");
        return first;
    }
    prev = NULL;
    curr = first;
    while (curr->next != NULL)
    {
        prev = curr;
```

```
        curr = curr->next;
    }
    prev->next = NULL;
    printf("%d", curr->value);
    free(curr);
    return first;
}
NODE del_value(NODE first, int key)
{
    NODE prev, curr;
    if (first == NULL)
    {
        printf("list is empty\n");
        return first;
    }
    curr = first;
    if (curr->value == key)
    {
        printf("Item deleted =%d", curr->value);
        first = first->next;
        free(curr);
        return first;
    }
    prev = NULL;
    curr = first;
    while (curr->value != key && curr != NULL)
    {
        prev = curr;
        curr = curr->next;
    }
    if (curr->value == key)
    {
        prev->next = curr->next;
        printf("Item deleted %d=", curr->value);
        free(curr);
        return first;
    }
    if (curr == NULL)
    {
        printf("End of list reached and item not found\n");
        return first;
    }
}
int main()
{
    int pos, item, c;
```

```
    NODE first = NULL;
    while (1)
    {
        printf("\n1.Insert at the end\n2.Display\n3.Delete at beginning\n4.Delete at the
end\n5.Delete a specific element\n");
        printf("Enter your choice :");
        scanf("%d", &c);
        switch (c)
        {
        case 1:
            printf("Enter the item to be inserted :");
            scanf("%d", &item);
            first = insert_end(first, item);
            break;
        case 2:
            display(first);
            break;
        case 3:
            first = del_beg(first);
            break;
        case 4:
            first = del_end(first);
            break;
        case 5:
            printf("Enter the item to be deleted :");
            scanf("%d", &item);
            first = del_value(first, item);
            break;
        default:
            printf("Invalid choice!!!");
            exit(0);
        }
    }
}
```

2.6.3 Output:


File Edit Selection View Go Run Terminal Help
DST6.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
DST5.c
DST5.exe
DST6.c
DST6.exe
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
1.Insert at the end
2.Display
3.Delete at beginning
4.Delete at the end
5.Delete a specific element
Enter your choice :1
Enter the item to be inserted :1
1.Insert at the end
2.Display
3.Delete at beginning
4.Delete at the end
5.Delete a specific element
Enter your choice :1
Enter the item to be inserted :2
1.Insert at the end
2.Display
3.Delete at beginning
4.Delete at the end
5.Delete a specific element
Enter your choice :1
Enter the item to be inserted :3
1.Insert at the end
2.Display
3.Delete at beginning
4.Delete at the end
5.Delete a specific element
Enter your choice :5
Enter the item to be deleted :2
Item deleted 2=
1.Insert at the end
2.Display
3.Delete at beginning
4.Delete at the end
Ln 162, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32
Type here to search
26°C
ENG
17:44
25-01-2023


File Edit Selection View Go Run Terminal Help
DST6.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
DST5.c
DST5.exe
DST6.c
DST6.exe
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Enter your choice :1
Enter the item to be inserted :3
1.Insert at the end
2.Display
3.Delete at beginning
4.Delete at the end
5.Delete a specific element
Enter your choice :5
Enter the item to be deleted :2
Item deleted 2=
1.Insert at the end
2.Display
3.Delete at beginning
4.Delete at the end
5.Delete a specific element
Enter your choice :3
Item deleted =1
1.Insert at the end
2.Display
3.Delete at beginning
4.Delete at the end
5.Delete a specific element
Enter your choice :4
PS C:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab>
Ln 162, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32
Type here to search
26°C
ENG
17:44
25-01-2023

2.7 Experiment: 7

2.7.1 Question:

WAP to Implement Single Link List with following operations:

a) Sort the linked list.
b) Reverse the linked list.
c) Concatenation of two linked lists.

2.7.2 Code:

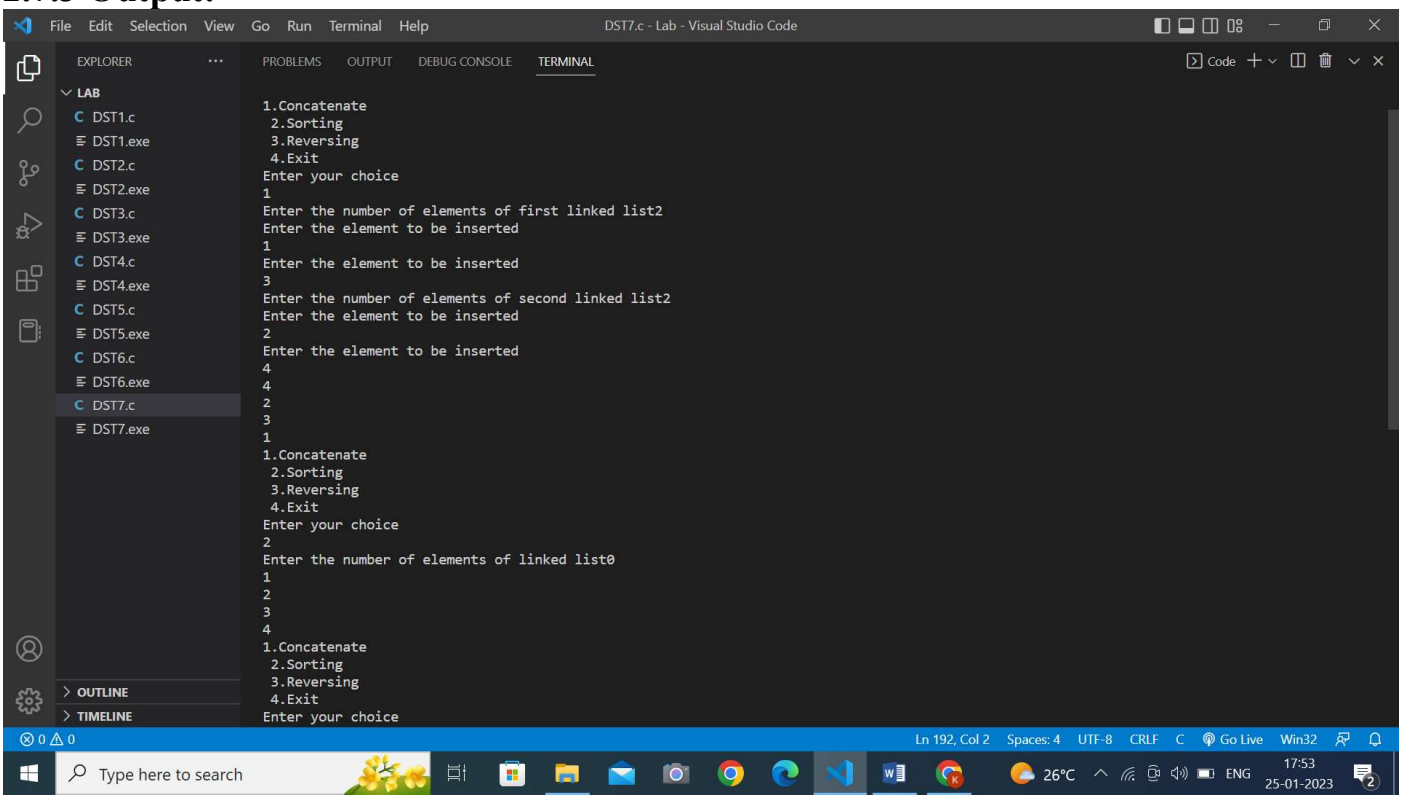
```
#include <stdio.h>
#include <stdlib.h>
struct node
{
   int value;
   struct node *next;
};
typedef struct node *NODE;
NODE getnode()
 {
        NODE temp;
        temp= (NODE)malloc(sizeof(struct node));
        if(temp==NULL)
   {
        printf("Memory not allocated");
        return NULL;
   }
   return temp;
}
NODE concatenate(NODE first, NODE second)
{
   NODE temp=first;
   if(temp==NULL && second==NULL)
   {
      return NULL;
   }
   else if(temp==NULL)
   {
      return second;
   }
   else
   {
      while(temp->next!=NULL)
      {
         temp=temp->next;
      }
      temp->next=second;
```

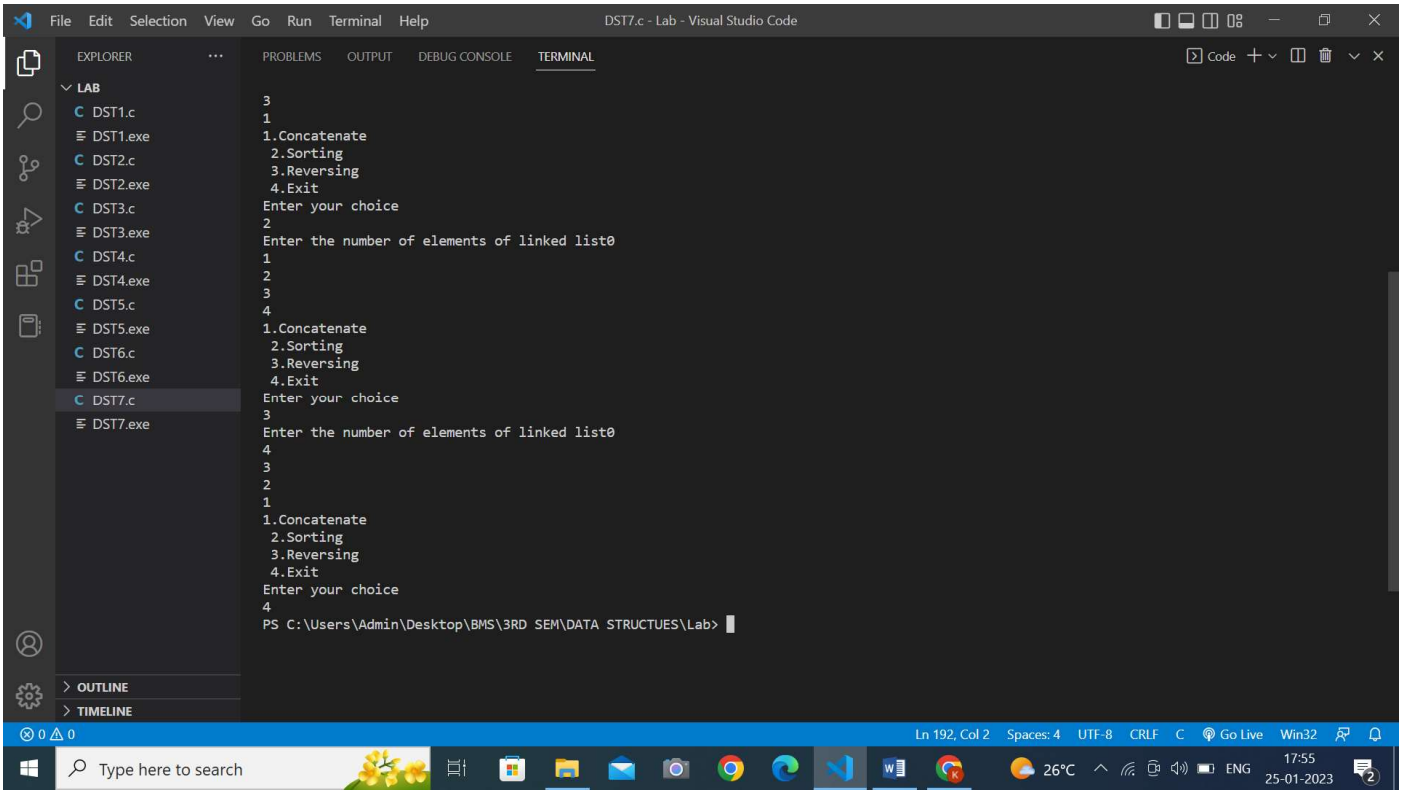
```
        return first;
    }
}
NODE sort(NODE first)
{
    int temp;
    NODE second;
    struct node*p=first;
    while(p->next !=NULL)
    {
        if(p==NULL && p->next==NULL)
        {
            return NULL;
        }
        second=p->next;
        while(second!=NULL)
        {
        if(p->value > second->value)
        {
            temp=p->value;
            p->value= second->value;
            second->value=temp;
        }
        second=second->next;

        }
        p=p->next;
    }
    return first;
}
NODE reverse(NODE first)
{
    NODE prev=NULL;
    NODE curr=first;
    NODE next=NULL;
    while(curr!=NULL)
    {
        next=curr->next;
        curr->next=prev;
        prev=curr;
        curr=next;
    }
    first=prev;
    return first;}
NODE insert_at_beg(int item, NODE first)
{
```

```
   NODE new;
   new=getnode();
   new->value=item;
   new->next=NULL;
   if(first==NULL)
   {
         return new;
   }
   else{


    new->next = first;
    first = new;
    return first;
}
}
void display(NODE first)
{
         NODE temp;
         temp=first;
         if(first==NULL)
         {
                  printf("List is empty \n");
         }
         while(temp!=NULL)
         {
                  printf("%d \n",temp-> value);
                  temp= temp->next;
         }
         }

int main()
{
    int n,i,item,ch;
    NODE first=NULL;
    NODE second=NULL;
    while(1)
    {
    printf("1.Concatenate \n 2.Sorting \n 3.Reversing \n 4.Exit \n");
    printf("Enter your choice\n");
    scanf("%d", &ch);
    switch(ch)
    {
       case 1: printf("Enter the number of elements of first linked list");
             scanf("%d",&n);
              for(i=1;i<=n;i++)
```

```
        {
           printf("Enter the element to be inserted \n");
           scanf("%d",&item);
           first=insert_at_beg(item,first);
        }

        printf("Enter the number of elements of second linked list");
        scanf("%d",&n);
        for(i=1;i<=n;i++)
        {
           printf("Enter the element to be inserted \n");
           scanf("%d",&item);
           first=insert_at_beg(item,first);
        }

        first=concatenate(first,second);
        display(first);
        break;
    case 2:  printf("Enter the number of elements of linked list");
        scanf("%d",&n);
        for(i=1;i<=n;i++)
        {
           printf("Enter the element to be inserted \n");
           scanf("%d",&item);
           first=insert_at_beg(item,first);
        }
        first=sort(first);
        display(first);
        break;
    case 3:printf("Enter the number of elements of linked list");
        scanf("%d",&n);
        for(i=1;i<=n;i++)
        {
           printf("Enter the element to be inserted \n");
           scanf("%d",&item);
           first=insert_at_beg(item,first);
        }
        first=reverse(first);
        display(first);
        break;

    case 4:exit(0);

    default:printf("Enter correct input value");
}
}}
```

2.7.3 Output:


File Edit Selection View Go Run Terminal Help
DST7.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
DST5.c
DST5.exe
DST6.c
DST6.exe
DST7.c
DST7.exe
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
1.Concatenate
2.Sorting
3.Reversing
4.Exit
Enter your choice
1
Enter the number of elements of first linked list2
Enter the element to be inserted
1
Enter the element to be inserted
3
Enter the number of elements of second linked list2
Enter the element to be inserted
2
Enter the element to be inserted
4
4
2
3
1
1.Concatenate
2.Sorting
3.Reversing
4.Exit
Enter your choice
2
Enter the number of elements of linked list0
1
2
3
4
1.Concatenate
2.Sorting
3.Reversing
4.Exit
Enter your choice
Ln 192, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32
Type here to search
26°C
ENG
17:53
25-01-2023


File Edit Selection View Go Run Terminal Help
DST7.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
DST5.c
DST5.exe
DST6.c
DST6.exe
DST7.c
DST7.exe
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
3
1
1.Concatenate
2.Sorting
3.Reversing
4.Exit
Enter your choice
2
Enter the number of elements of linked list0
1
2
3
4
1.Concatenate
2.Sorting
3.Reversing
4.Exit
Enter your choice
3
Enter the number of elements of linked list0
4
3
2
1
1.Concatenate
2.Sorting
3.Reversing
4.Exit
Enter your choice
4
PS C:\Users\Admin\Desktop\BMS\3RD SEM\DATA STRUCTUES\Lab>
Ln 192, Col 2 Spaces: 4 UTF-8 CRLF C Go Live Win32
Type here to search
26°C
ENG
17:55
25-01-2023

2.8 Experiment: 8

2.8.1 Question:
WAP to implement Stack & Queues using Linked Representation.

2.8.2 Code:
STACKS

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   int value;
   struct node *next;
};
typedef struct node *NODE;
NODE getnode()
{
   NODE temp;
   temp = (NODE)malloc(sizeof(struct node));
   if (temp == NULL)
   {
      printf("Memory not allocated");
      return NULL;
   }
   return temp;
}
NODE insert_beg(NODE first, int item)
{
   NODE new;
   new = getnode();
   new->value = item;
   new->next = NULL;
   if (first == NULL)
   {
      return new;
   }
   else
   {
      new->next = first;
      first = new;
      return first;
   }
}
void display(NODE first)
{
   NODE temp;
```

```
    temp = first;
    if (first == NULL)
    {
        printf("List is empty\n");
    }
    while (temp != NULL)
    {
        printf("Value stored in the node=%d\n", temp->value);
        temp = temp->next;
    }
}
NODE del_beg(NODE first)
{
    NODE temp;
    if (first == NULL)
    {
        printf("list is empty\n");
        return first;
    }
    temp = first;
    temp = temp->next;
    printf("Item deleted =%d", first->value);
    free(first);
    return temp;
}
int main()
{
    int pos, item, c;
    NODE first = NULL;
    while (1)
    {
        printf("\n1.Insert at beginning\n4.Display\n5.Delete at beginning\n6");
        printf("Enter your choice :");
        scanf("%d", &c);
        switch (c)
        {
        case 1:
            printf("Enter the item to be inserted :");
            scanf("%d", &item);
            first = insert_beg(first, item);
            break;
        case 4:
            display(first);
            break;
        case 5:
            first = del_beg(first);
```

```
            break;
        default:
            exit(0);
        }
    }
}
```

QUEUES

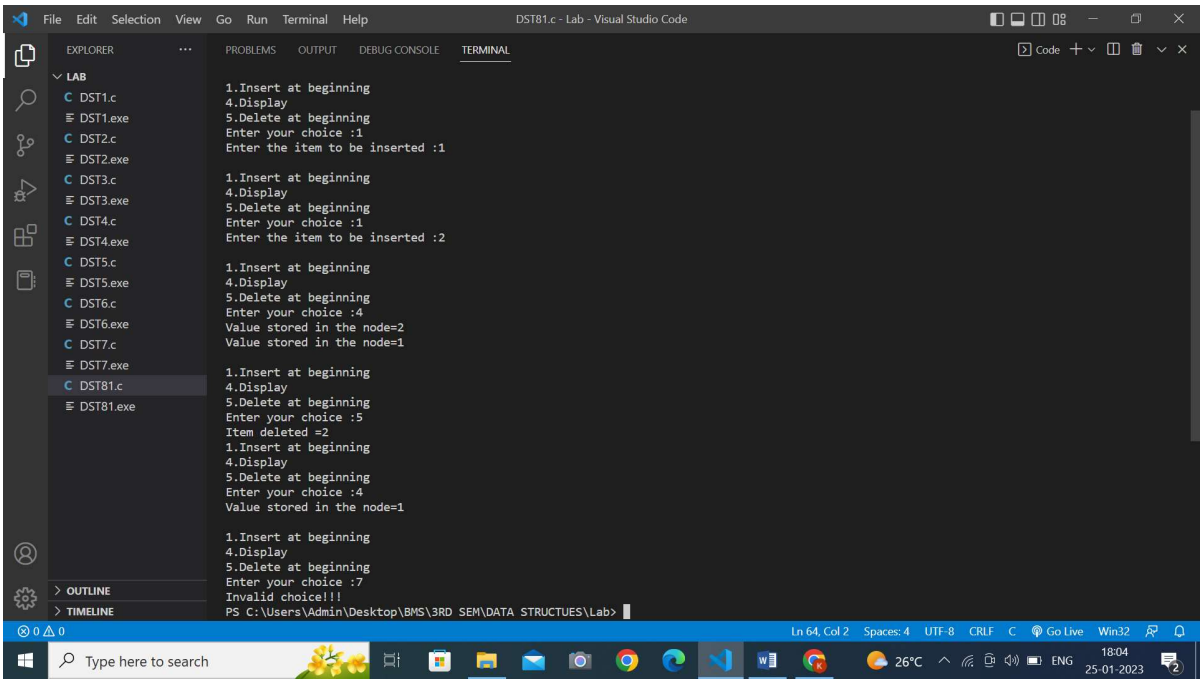
```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   int value;
   struct node *next;
};
typedef struct node *NODE;
NODE getnode()
{
   NODE temp;
   temp = (NODE)malloc(sizeof(struct node));
   if (temp == NULL)
   {
      printf("Memory not allocated");
      return NULL;
   }
   return temp;
}
NODE insert_beg(NODE first, int item)
{
   NODE new;
   new = getnode();
   new->value = item;
   new->next = NULL;
   if (first == NULL)
   {
      return new;
   }
   else
   {
      new->next = first;
      first = new;
      return first;
   }}
void display(NODE first)
{
   NODE temp;
```

```
    temp = first;
    if (first == NULL)
    {
        printf("List is empty\n");
    }
    while (temp != NULL)
    {
        printf("Value stored in the node=%d\n", temp->value);
        temp = temp->next;
    }
}
NODE del_beg(NODE first)
{
    NODE temp;
    if (first == NULL)
    {
        printf("list is empty\n");
        return first;
    }
    temp = first;
    temp = temp->next;
    printf("Item deleted =%d", first->value);
    free(first);
    return temp;
}
int main()
{
    int pos, item, c;
    NODE first = NULL;
    while (1)
    {
        printf("\n1.Insert at beginning\n4.Display\n5.Delete at beginning\n");
        printf("Enter your choice :");
        scanf("%d", &c);
        switch (c)
        {
        case 1:
            printf("Enter the item to be inserted :");
            scanf("%d", &item);
            first = insert_beg(first, item);
            break;
        case 4:
            display(first);
            break;
        case 5:
            first = del_beg(first);
```

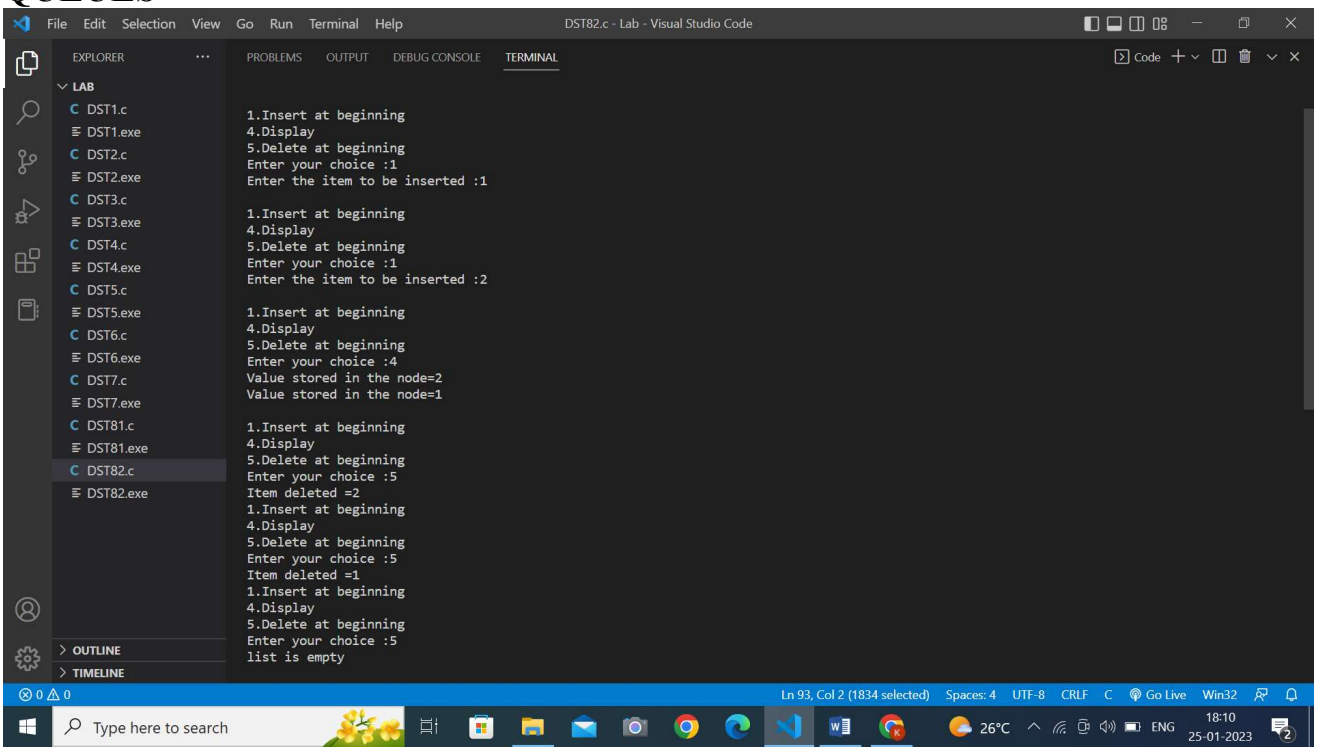
```
            break;
        default:
            exit(0);
        }
    }
}
```

Output:

STACKS



QUEUES



2.9 Experiment: 9

2.9.1 Question:

WAP to Implement doubly link list with primitive operations:

a) Create a doubly linked list.
b) Insert a new node to the left of the node.
c) Delete the node based on a specific value.
d) Display the contents of the list.

2.9.2 Code:

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   int value;
   struct node *prev;
   struct node *next;
};
typedef struct node *NODE;
NODE getnode()
{
   NODE temp;
   temp=(NODE)malloc(sizeof(struct node));
   if(temp==NULL)
   {
      printf("Memory not allocated\n");
      return NULL;
   }
   return temp;
}
NODE insert_beg(NODE first, int item)
{
   NODE new;
```

```
    new=getnode();
    new->value=item;
    new->next=NULL;
    new->prev=NULL;
    if(first==NULL)
        return new;
    new->next=first;
    first->prev=new;
    return new;
}
NODE insert_left(NODE first,int item,int key)
{
    NODE new,temp;
    new=getnode();
    new->value=item;
    new->next=NULL;
    new->prev=NULL;
    if(first==NULL)
    {
        printf("list is empty there no left element\n");
        return first;
    }
    if(first->next==NULL && first->value==key)
    {
        new->next=first;
        first->prev=new;
        return new;
    }
    if(first->next==NULL && first->value!=key)
    {
```

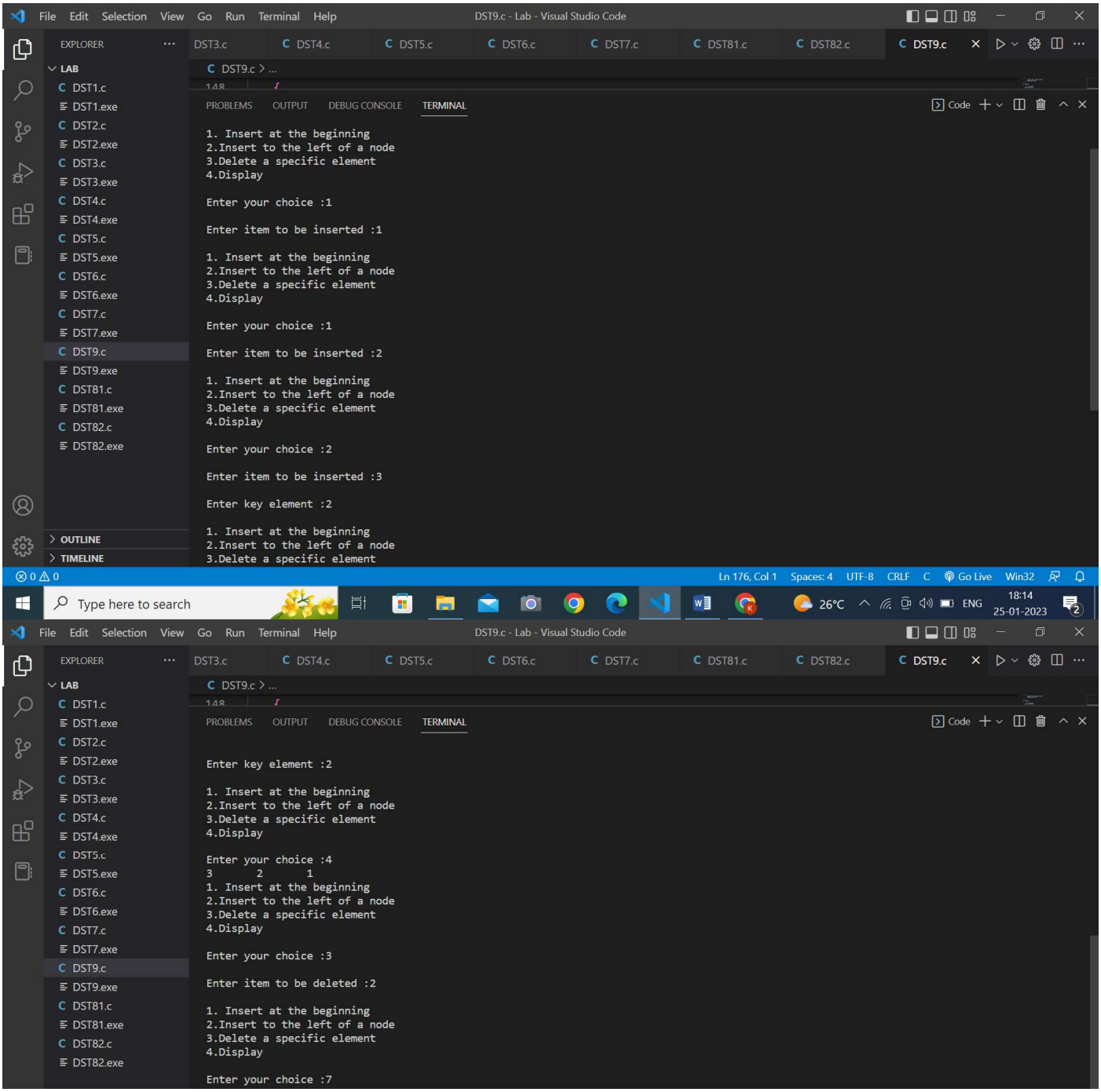
```
        printf("value not found cannot be inserted to left\n");
        return first;
    }
    if(first->value==key)
    {
        first=insert_beg(first,item);
        return first;
    }
    temp=first;
    while(temp->value!=key && temp->next!=NULL)
    {
        temp=temp->next;
    }
    if(temp->value==key)
    {
        new->next=temp;
        new->prev=temp->prev;
        temp->prev->next=new;
        temp->prev=new;
        return first;
    }
    if(temp->value!=key)
    {
        printf("value not found cannot insert to left\n");
        return first;
    }

}
NODE delete_specific(NODE first,int key)
{
```

```
NODE curr,temp;
if(first==NULL)
{
   printf("list is empty\n");
   return first;
}
if(first->next==NULL && first->value!=key)
{
    printf("value not found\n");
    return first;
}
if(first->next==NULL && first->value==key)
{
   free(first);
   return first;
}
if(first->value==key)
{
   first->next->prev=NULL;
   temp=first->next;
   free(first);
   return temp;
}
curr=first;
while(curr->value!=key && curr!=NULL)
{
   curr=curr->next;
   if(curr==NULL)
   {
      printf("element not found cannot delete\n");
```

```
            return first;
        }

    }
    if(curr->value==key)
    {
        curr->prev->next=curr->next;
        if(curr->next!=NULL)
        {
            curr->next->prev=curr->prev;
        }
        free(curr);
        return first;
    }
}
void display(NODE first)
{
    NODE temp;
    temp=first;
    if(temp==NULL)
    {
        printf("list is empty");
    }
    while(temp!=NULL)
    {
        printf("%d\t",temp->value);
        temp=temp->next;
    }
}
void main()
```

```
{
        int c,item,key;
   NODE first=NULL;
        while(1)
        {
                printf("\n1. Insert at the beginning \n2.Insert to the left of a node\n3.Delete a specific element\n4.Display\n");
                printf("\nEnter your choice :");
                scanf("%d",&c);
                switch(c)
                {
                        case 1: printf("\nEnter item to be inserted :");
                                        scanf("%d",&item);
                                        first=insert_beg(first,item);
                                        break;
                        case 2: printf("\nEnter item to be inserted :");
                                        scanf("%d",&item);
                printf("\nEnter key element :");
                                        scanf("%d",&key);
                first=insert_left(first,item,key);
                                        break;
                        case 3: printf("\nEnter item to be deleted :");
                                        scanf("%d",&key);
                first=delete_specific(first,key);
                break;
         case 4: display(first);
                                        break;
                        default:  exit(0);
                                         break;
                }} }
```

2.9.3 Output:


File Edit Selection View Go Run Terminal Help
DST9.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
DST5.c
DST5.exe
DST6.c
DST6.exe
DST7.c
DST7.exe
DST9.c
DST9.exe
DST81.c
DST81.exe
DST82.c
DST82.exe
OUTLINE
TIMELINE
DST3.c
DST4.c
DST5.c
DST6.c
DST7.c
DST81.c
DST82.c
DST9.c
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
1. Insert at the beginning
2.Insert to the left of a node
3.Delete a specific element
4.Display
Enter your choice :1
Enter item to be inserted :1
1. Insert at the beginning
2.Insert to the left of a node
3.Delete a specific element
4.Display
Enter your choice :1
Enter item to be inserted :2
1. Insert at the beginning
2.Insert to the left of a node
3.Delete a specific element
4.Display
Enter your choice :2
Enter item to be inserted :3
Enter key element :2
1. Insert at the beginning
2.Insert to the left of a node
3.Delete a specific element
Ln 176, Col 1
Spaces: 4
UTF-8
CRLF
C
Go Live
Win32
Type here to search
26°C
ENG
18:14
25-01-2023

File Edit Selection View Go Run Terminal Help
DST9.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
DST5.c
DST5.exe
DST6.c
DST6.exe
DST7.c
DST7.exe
DST9.c
DST9.exe
DST81.c
DST81.exe
DST82.c
DST82.exe
DST3.c
DST4.c
DST5.c
DST6.c
DST7.c
DST81.c
DST82.c
DST9.c
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
Enter key element :2
1. Insert at the beginning
2.Insert to the left of a node
3.Delete a specific element
4.Display
Enter your choice :4
3 2 1
1. Insert at the beginning
2.Insert to the left of a node
3.Delete a specific element
4.Display
Enter your choice :3
Enter item to be deleted :2
1. Insert at the beginning
2.Insert to the left of a node
3.Delete a specific element
4.Display
Enter your choice :7

2.10 Experiment: 10

2.10.1 Question:

Write a program:

a) To construct a binary Search tree.
b) To traverse the tree using all the methods i.e., in-order, preorder and post order.
c) To display the elements in the tree.

2.10.2 Code:

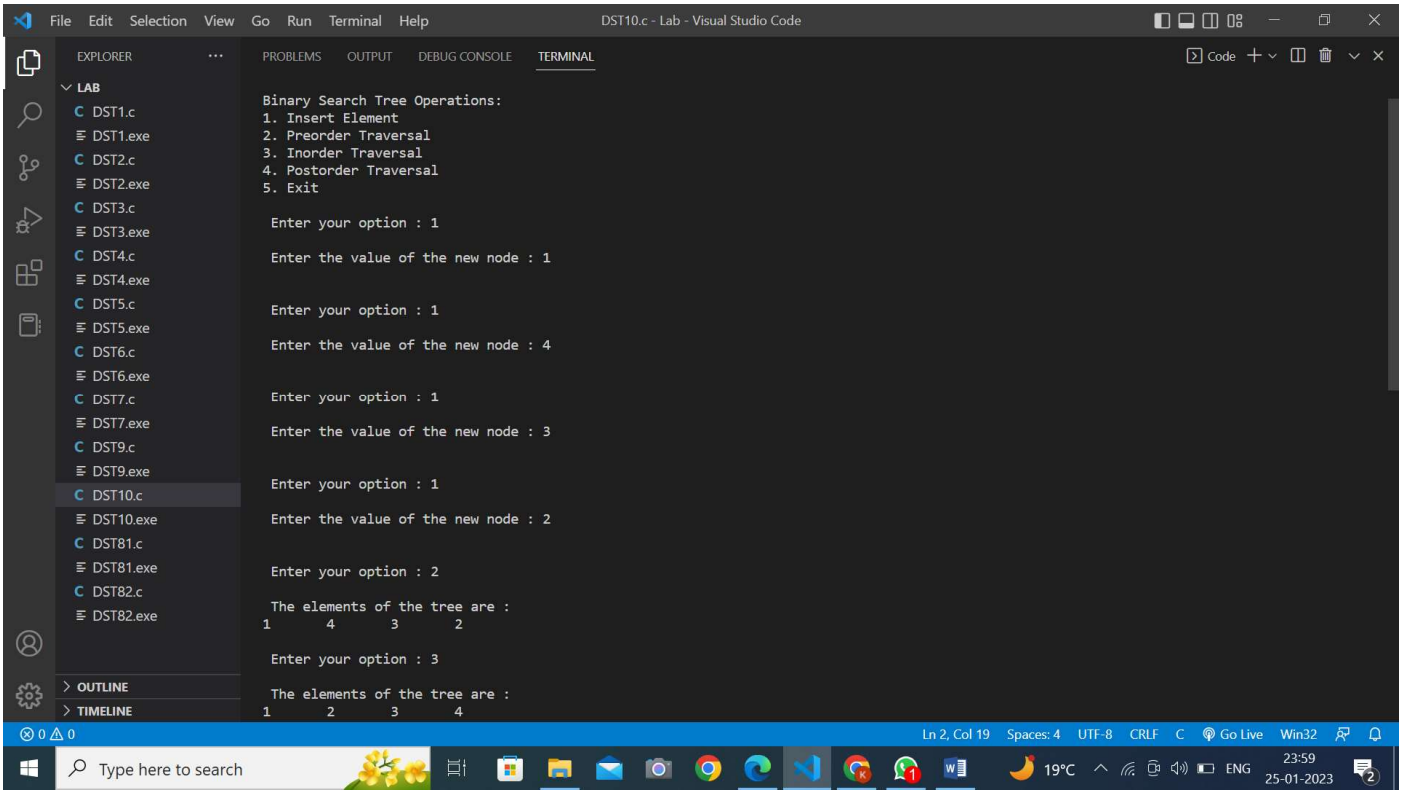
```
#include <stdio.h>
#include <conio.h>
#include <malloc.h>

struct node
{
   int data;
   struct node *left;
   struct node *right;
};

struct node *tree;

void create_tree(struct node *);
struct node *insertElement(struct node *, int);
void preorderTraversal(struct node *);
void inorderTraversal(struct node *);
void postorderTraversal(struct node *);

int main()
{
   int option, val;
   struct node *ptr;
   create_tree(tree);
                printf("Binary Search Tree Operations:\n");
                printf("1. Insert Element");
                printf("\n2. Preorder Traversal");
                printf("\n3. Inorder Traversal");
                printf("\n4. Postorder Traversal");
                printf("\n5. Exit");

   do
   {
                printf("\n\n Enter your option : ");
                scanf("%d", &option);
                switch(option)
 {
```

```
                        case 1:
                                printf("\n Enter the value of the new node : ");
                                scanf("%d", &val);
                                tree = insertElement(tree, val);
                                break;
                        case 2:
                                printf("\n The elements of the tree are : \n");
                                preorderTraversal(tree);
                                break;
                        case 3:
                                printf("\n The elements of the tree are : \n");
                                inorderTraversal(tree);
                                break;
                        case 4:
                                printf("\n The elements of the tree are : \n");
                                postorderTraversal(tree);
                                break;
        default: exit(0);
 }
}while(option!=14);
getch();
return 0;
}

void create_tree(struct node *tree)
{
   tree = NULL;
}

struct node *insertElement(struct node *tree, int val)
{
   struct node *ptr, *nodeptr, *parentptr;
   ptr = (struct node*)malloc(sizeof(struct node));
   ptr->data = val;
   ptr->left = NULL;
   ptr->right = NULL;

   if(tree==NULL)
   {
                tree=ptr;
                tree->left=NULL;
                tree->right=NULL;
   }

   else
   {
```

```
                parentptr=NULL;
                nodeptr=tree;
                while(nodeptr!=NULL)
    {
            parentptr=nodeptr;

            if(val<nodeptr->data)
               nodeptr=nodeptr->left;

            else
                    nodeptr = nodeptr->right;
    }
                if(val<parentptr->data)
                        parentptr->left = ptr;

                else
                        parentptr->right = ptr;
    }
return tree;
}

void preorderTraversal(struct node *tree)
{
   if(tree != NULL)
   {
      printf("%d\t", tree->data);
      preorderTraversal(tree->left);
      preorderTraversal(tree->right);
   }

}

void inorderTraversal(struct node *tree)
{
   if(tree != NULL)
   {
      inorderTraversal(tree->left);
      printf("%d\t", tree->data);
      inorderTraversal(tree->right);
   }
}

void postorderTraversal(struct node *tree)
{
   if(tree != NULL)
   {
```

```
        postorderTraversal(tree->left);
        postorderTraversal(tree->right);
        printf("%d\t", tree->data);
    }

}
```

2.10.3 Output:


File Edit Selection View Go Run Terminal Help
DST10.c - Lab - Visual Studio Code
EXPLORER
LAB
DST1.c
DST1.exe
DST2.c
DST2.exe
DST3.c
DST3.exe
DST4.c
DST4.exe
DST5.c
DST5.exe
DST6.c
DST6.exe
DST7.c
DST7.exe
DST9.c
DST9.exe
DST10.c
DST10.exe
DST81.c
DST81.exe
DST82.c
DST82.exe
OUTLINE
TIMELINE
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
Binary Search Tree Operations:
1. Insert Element
2. Preorder Traversal
3. Inorder Traversal
4. Postorder Traversal
5. Exit
Enter your option : 1
Enter the value of the new node : 1
Enter your option : 1
Enter the value of the new node : 4
Enter your option : 1
Enter the value of the new node : 3
Enter your option : 1
Enter the value of the new node : 2
Enter your option : 2
The elements of the tree are :
1 4 3 2
Enter your option : 3
The elements of the tree are :
1 2 3 4
Ln 2, Col 19 Spaces: 4 UTF-8 CRLF C Go Live Win32
Type here to search
19°C
ENG
23:59
25-01-2023