Course Name: Data Structures and Algorithms Lab
Course code : MCSE501P
Faculty Name: SARAVANAN R – SCOPE

Lab Assessment – 1

List of programs:

1. Write a program in C using arrays to implement the following: Create two lists A and B to store data (receive the input from key board, A - to store int and B - to store char). Perform the following operations:
    a. Deletion of 1st, last, intermediate elements (get the input from the key board)
    b. Insertion at the beginning, end, intermediate locations (get the input from the key board)
    c. Splitting of list into two equal parts
2. Write a program in C to implement queue and operations on it (enqueue, dequeue, location of front and back pointers, display elements). Get the input from the key board for creating queue.
3. Write a program in C to implement circular queue and operations on it (enqueue, dequeue, location of front and back pointers, display elements). Get the input from the key board for creating queue.
4. Write a program in C to implement stack and operations on the it (push, pop, top element, display elements).
5. Write a program in C to evaluate the postfix form of an algebraic expression using stack. Get the input from the keyboard.

1. **Write a program in C using arrays to implement the following: Create two lists A and B to store data (receive the input from key board, A - to store int and B - to store char). Perform the following operations:**

a.Deletion of 1st, last, intermediate elements (get the input from the key board)

b. Insertion at the beginning, end, intermediate locations (get the input from the key board)

c. Splitting of list into two equal parts

Program code:

```
#include <stdio.h>

int actualIndexInt=0;
int actualIndexChar=0;

void menu(){
   printf("\nPress 1. Delete 1st element");
   printf("\nPress 2. Delete last element");
   printf("\nPress 3. Delete element of a given index");
   printf("\nPress 4. Insert element in the begining");
   printf("\nPress 5. Insert element in the end");
   printf("\nPress 6. Insert element at a given index");
   printf("\nPress 7. Split the list");
   printf("\nEnter your choice : ");
}

void printIntArray(int arr[],int n){
   for(int i=0;i<=n;i++)
      if(i!=n)
         printf("%d, ",arr[i]);
      else
         printf("%d ",arr[i]);
   printf("\n");
}

void printCharArray(char arr[],int n){
   for(int i=0;i<=n;i++)
      if(i!=n)
         printf("%c, ",arr[i]);
      else
         printf("%c ",arr[i]);
   printf("\n");
}

int main(){
   int n1;
   int n2;
   int temp;
   char tempc;
   printf("Enter size of integer array : ");
   scanf("%d",&n1);
   actualIndexInt=n1-1;
```

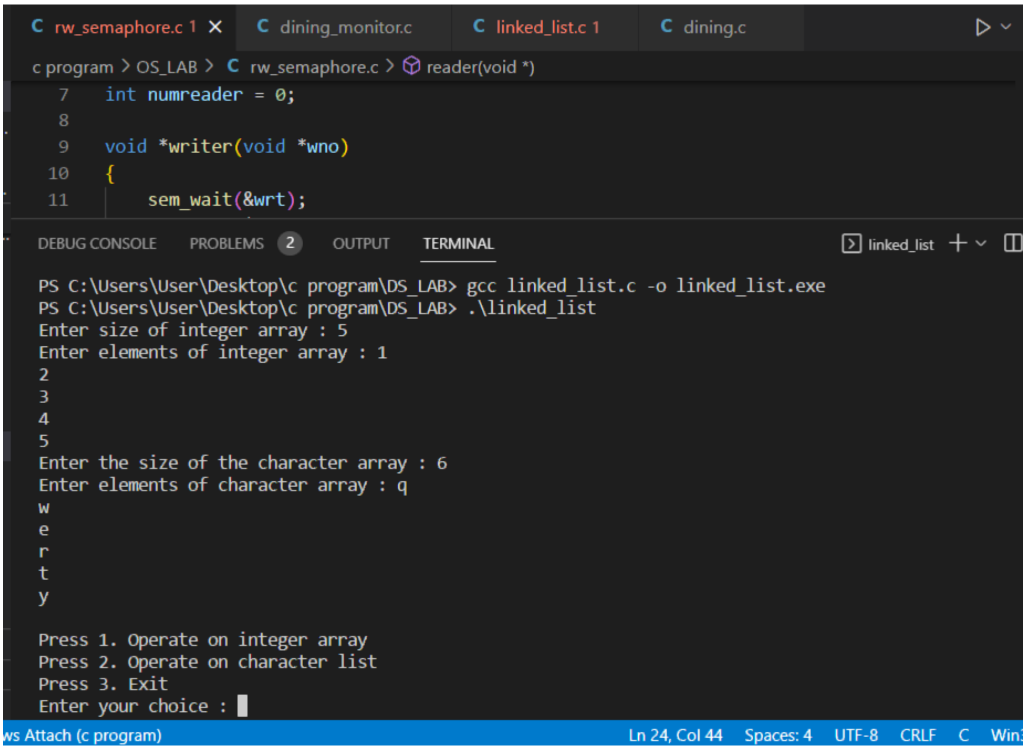
```
int intArr[(n1+5)];
int i;
printf("Enter elements of integer array : ");
for(i=0;i<n1;i++)
   scanf("%d",&intArr[i]);
printf("Enter the size of the character array : ");
scanf("%d",&n2);
actualIndexChar=n2-1;
char charArr[(n2+5)];
printf("Enter elements of character array : ");
for(i=0;i<n2;i++){
   fflush(stdin);
   scanf("%c",&charArr[i]);
}
int loop=1;
while(loop){
   printf("\nPress 1. Operate on integer array");
   printf("\nPress 2. Operate on character list");
   printf("\nPress 3. Exit");
   printf("\nEnter your choice : ");
   int ch1;
   scanf("%d",&ch1);
   switch(ch1){
      case 1: ;
         menu();
         int ch1;
         scanf("%d",&ch1);
         {
         switch(ch1){
            case 1: ;
               temp=intArr[0];
               for(i=1;i<=actualIndexInt;i++)
                  intArr[i-1]=intArr[i];
               printf("\n%d removed successfully from the list ",temp);
               actualIndexInt--;
               printIntArray(intArr,actualIndexInt);
            break;

            case 2: ;
               temp=intArr[actualIndexInt--];
               printf("\n%d removed successfully from the list ",temp);
               printIntArray(intArr,actualIndexInt);
            break;

            case 3: ;
               printf("\nEnter index of the element you want to delete");
               int ind;
               scanf("%d",&ind);
               if(ind<=actualIndexInt){
                  temp=intArr[ind];
                  for(i=ind+1;i<=actualIndexInt;i++)
                     intArr[i-1]=intArr[i];
                  printf("\n%d removed successfully from the list ",temp);
                  actualIndexInt--;
                  printIntArray(intArr,actualIndexInt);
               }
               else
                  printf("\nInvalid index");
            break;

            case 4: ;
               printf("\nEnter element to be inserted ");
               for(i=actualIndexInt;i>=0;i--)
                     intArr[i+1]=intArr[i];
               scanf("%d",&intArr[0]);
               actualIndexInt++;
               printf("\nElement addition successful ");
               printIntArray(intArr,actualIndexInt);
            break;

            case 5: ;
               printf("\nEnter element to be inserted");
               scanf("%d",&intArr[++actualIndexInt]);
               printf("\nElement addition succsful ");
               printIntArray(intArr,actualIndexInt);
            break;

            case 6: ;
               printf("\nEnter index in which element to be inserted");
               scanf("%d",&temp);
                  if(temp<=actualIndexInt){
                  for(i=actualIndexInt;i>=temp;i--)
                        intArr[i+1]=intArr[i];
                  printf("\nEnter element to be inserted");
                  scanf("%d",&intArr[temp] );
                  actualIndexInt++;
                  printf("\nElement added successfully\n");
                  printIntArray(intArr,actualIndexInt);
               }
               else
                  printf("\nInvalid index");
            break;

            case 7: ;
               int l1[actualIndexInt/2];
               int l1Len=0;
               int l2[actualIndexInt-(actualIndexInt/2)];
               int l2Len=0;
               for(int i=0;i<=actualIndexInt;i++){
                  if(i<=actualIndexInt/2)
                     l1[l1Len++]=intArr[i];
                  else
```

```
               l2[l2Len++]=intArr[i];
         }
         printf("\nThe splitted arrays are : \n");
         printIntArray(l1,l1Len-1);
         printIntArray(l2,l2Len-1);
   }
      break;

      /*default: ;
         printf("Wrong Input");    */
   }

   break;
case 2: ;
   menu();
   char ch2;
   fflush(stdin);
   scanf("%c",&ch2);
   {
   switch(ch2){
      case '1': ;
         tempc=charArr[0];
         for(i=1;i<=actualIndexChar;i++)
            charArr[i-1]=charArr[i];
         printf("\n%c removed successfully from the list ",tempc);
         actualIndexChar--;
         printCharArray(charArr,actualIndexChar);
      break;

      case '2': ;
         tempc=charArr[actualIndexChar--];
         printf("\n%c removed successfully from the list ",tempc);
         printCharArray(charArr,actualIndexChar);
      break;

      case '3': ;
         printf("\nEnter index of the element you want to delete ");
         int ind;
         scanf("%d",&ind);
         if(ind<=actualIndexChar){
            tempc=charArr[ind];
            for(i=ind+1;i<=actualIndexChar;i++)
               charArr[i-1]=charArr[i];
            printf("\n%c removed successfully from the list ",tempc);
            actualIndexChar--;
            printCharArray(charArr,actualIndexChar);
         }
         else
            printf("\nInvalid index");
      break;
```

```
case '4': ;
  printf("\nEnter element to be inserted ");
  for(i=actualIndexChar;i>=0;i--)
      charArr[i+1]=charArr[i];
  fflush(stdin);
  scanf("%c",&charArr[0]);
  actualIndexChar++;
  printf("\nElement addition successful ");
  printCharArray(charArr,actualIndexChar);
break;

case '5': ;
  printf("\nEnter element to be inserted ");
  fflush(stdin);
  scanf("%c",&charArr[++actualIndexChar]);
  printf("\nElement addition succsful ");
  printCharArray(charArr,actualIndexChar);
break;

case '6': ;
  printf("\nEnter index in which element to be inserted ");

  scanf("%d",&temp);
  if(temp<=actualIndexInt){
    for(i=actualIndexInt;i>=temp;i--)
        charArr[i+1]=charArr[i];
    printf("\nEnter element to be inserted ");
    fflush(stdin);
    scanf("%c",&charArr[temp]);
    actualIndexChar++;
    printf("\nElement added successfully ");
    printCharArray(charArr,actualIndexChar);
  }
  else
    printf("\nInvalid index");
break;

case '7': ;
  char lc1[(actualIndexChar/2)];
  int lc1Len=0;
  char lc2[(actualIndexChar-(actualIndexChar/2))];
  int lc2Len=0;
  for(int i=0;i<=actualIndexChar;i++){
    if(i<=actualIndexChar/2)
      lc1[lc1Len++]=charArr[i];
    else
      lc2[lc2Len++]=charArr[i];
  }
  printf("\nThe splitted arrays are : \n");
  printCharArray(lc1,lc1Len-1);
  printCharArray(lc2,lc2Len-1);
```

```
                  break;

                 /*/ default: ;
                     printf("Wrong Input"); */
               }
               }

               break;
            case 3: ;
               loop = 0;
            break;

            default: ;
               printf("Wrong input");
         }

      }
   }
```

Output:- Two lists IntArr and CharArr to store the input from key board, A - to store integer data and B - to store character data.

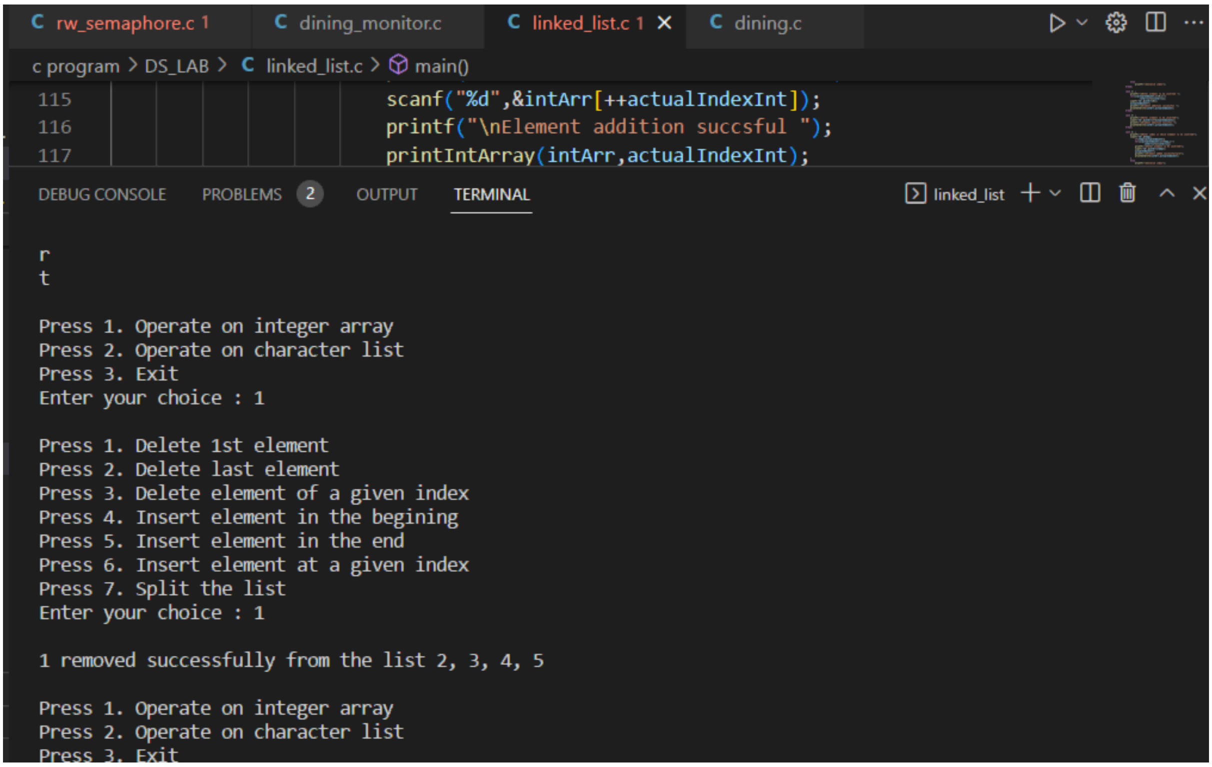


```
int numreader = 0;

void *writer(void *wno)
{
    sem_wait(&wrt);
```

```
PS C:\Users\User\Desktop\c program\DS_LAB> gcc linked_list.c -o linked_list.exe
PS C:\Users\User\Desktop\c program\DS_LAB> .\linked_list
Enter size of integer array : 5
Enter elements of integer array : 1
2
3
4
5
Enter the size of the character array : 6
Enter elements of character array : q
w
e
r
t
y

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice : █
```

Operation on Integer list :

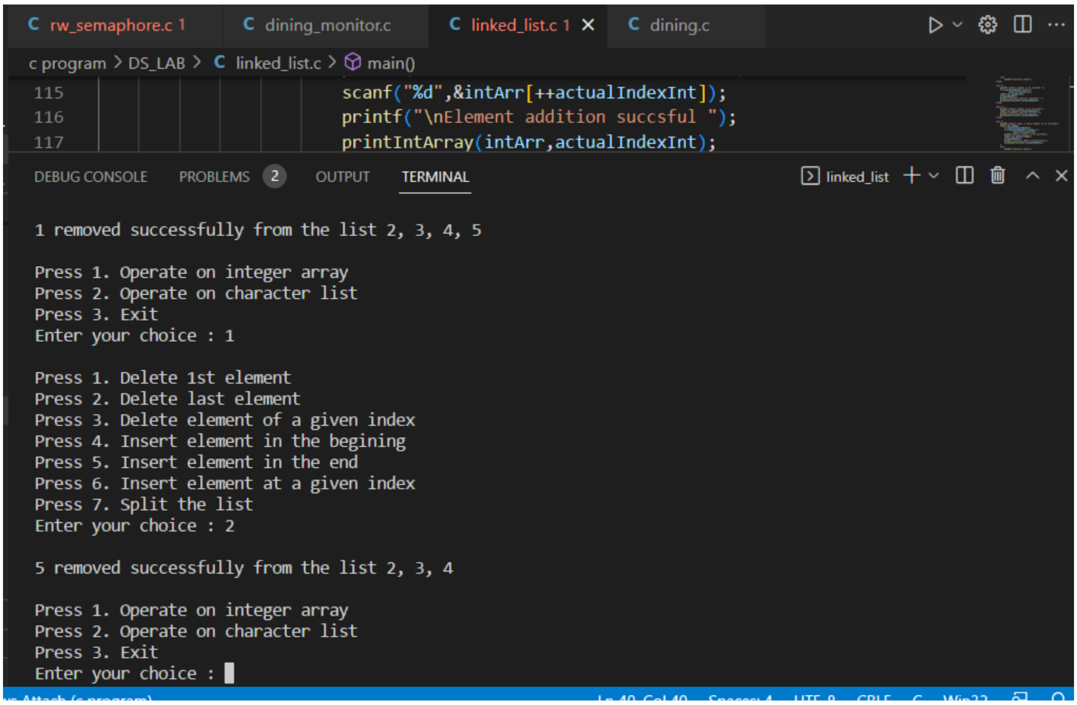
1. Deletion of First element :



```
scanf("%d",&intArr[++actualIndexInt]);
printf("\nElement addition succsful ");
printIntArray(intArr,actualIndexInt);
```

```
r
t

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 1

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 1

1 removed successfully from the list 2, 3, 4, 5

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
```

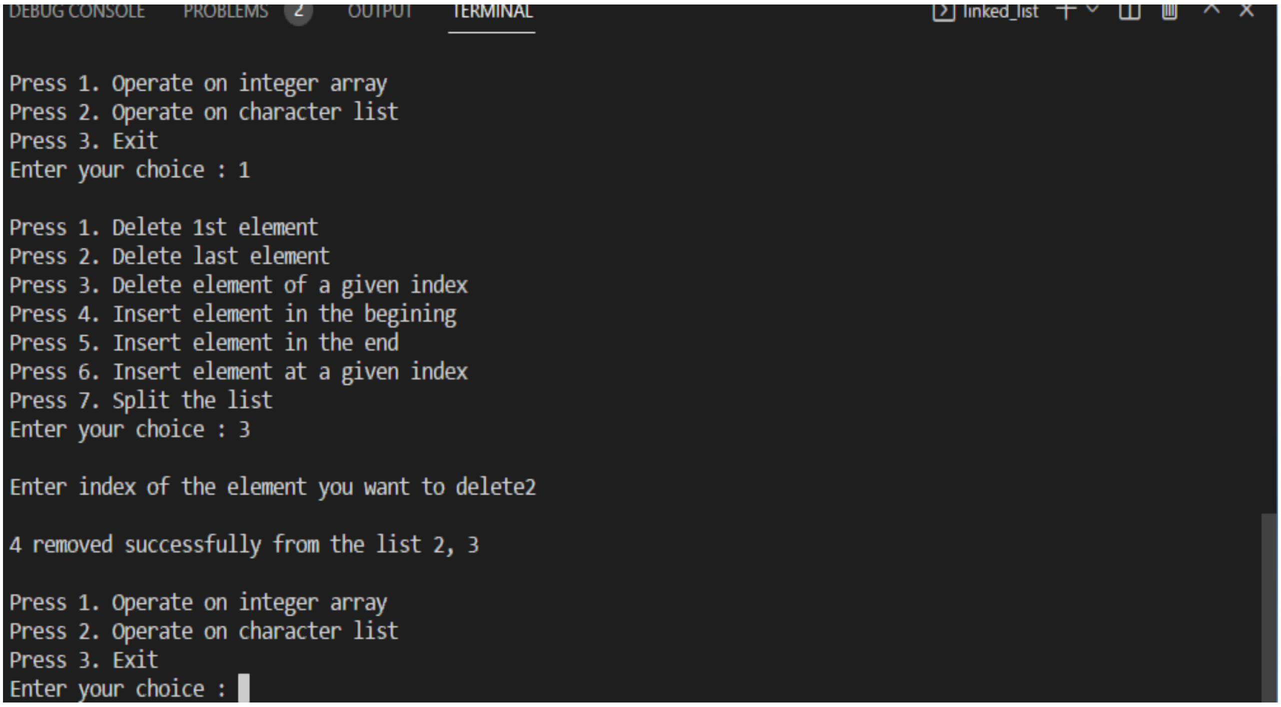
2. Deletion of last element :



```
scanf("%d",&intArr[++actualIndexInt]);
printf("\nElement addition succsful ");
printIntArray(intArr,actualIndexInt);
```

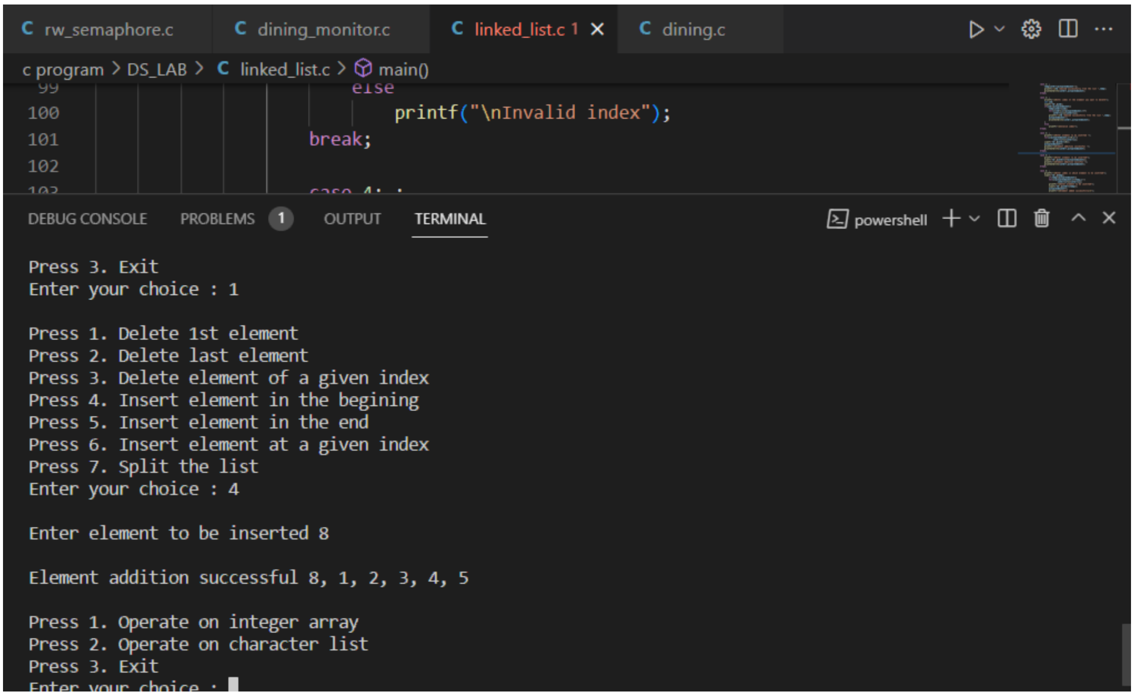
```
1 removed successfully from the list 2, 3, 4, 5

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 1

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 2

5 removed successfully from the list 2, 3, 4

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice :
```

3. Delete element of a given index :



```
Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 1

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 3

Enter index of the element you want to delete2

4 removed successfully from the list 2, 3

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice :
```

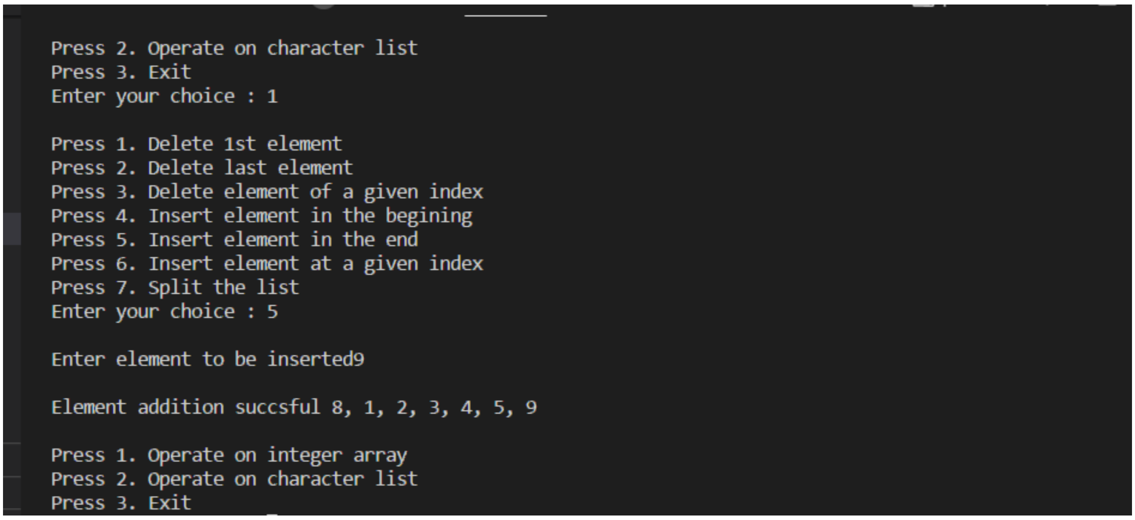
4. Insert at beginning



```
                else
                    printf("\nInvalid index");
            break;

            case 4: :
```

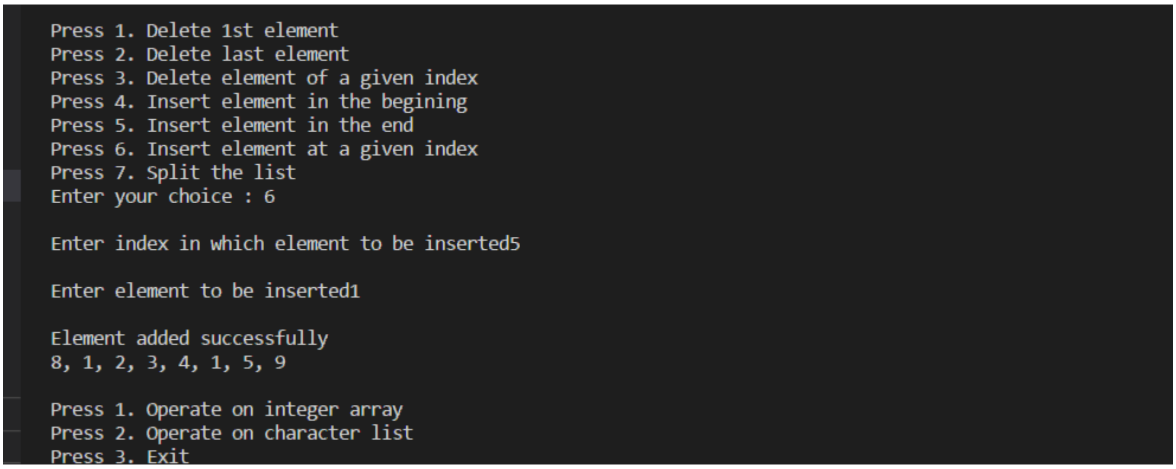
```
Press 3. Exit
Enter your choice : 1

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 4

Enter element to be inserted 8

Element addition successful 8, 1, 2, 3, 4, 5

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice :
```

5.Insert at end



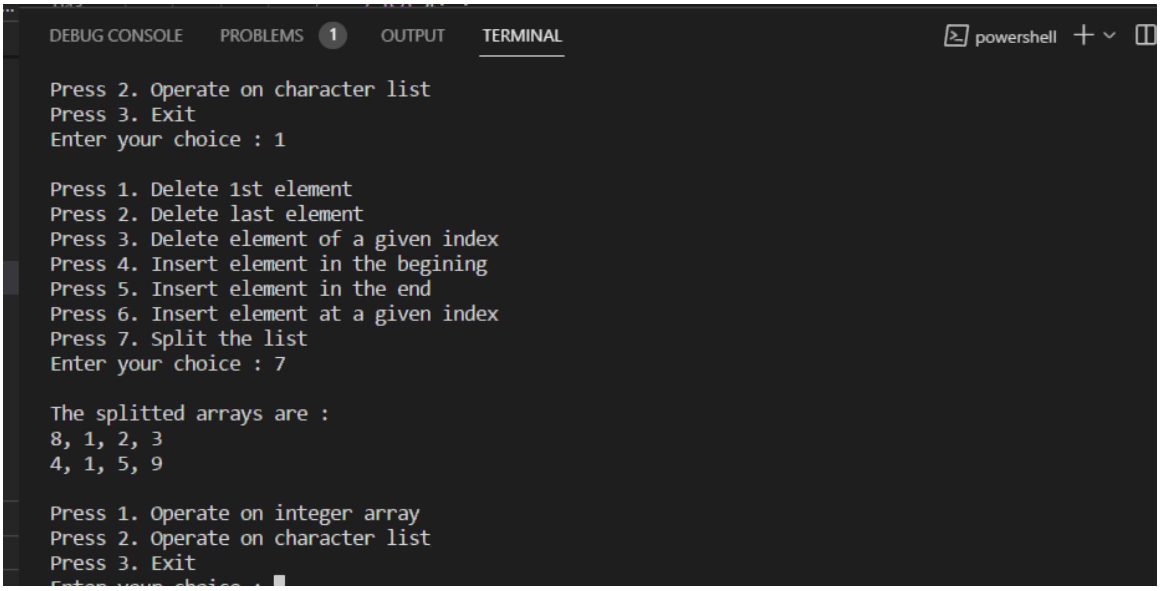
```
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 1

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 5

Enter element to be inserted9

Element addition succsful 8, 1, 2, 3, 4, 5, 9

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
```

6. Insert at given index



```
Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 6

Enter index in which element to be inserted5

Enter element to be inserted1

Element added successfully
8, 1, 2, 3, 4, 1, 5, 9

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
```

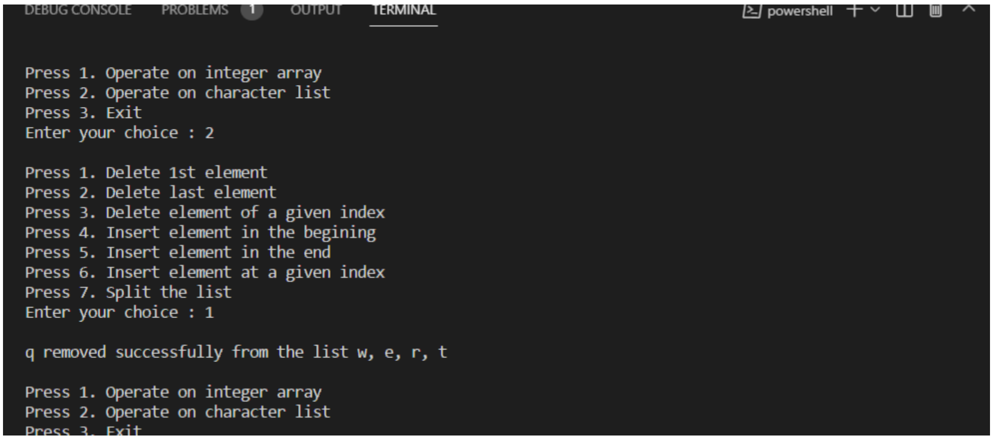
7. Split the list



```
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 1

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 7

The splitted arrays are :
8, 1, 2, 3
4, 1, 5, 9

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice :
```

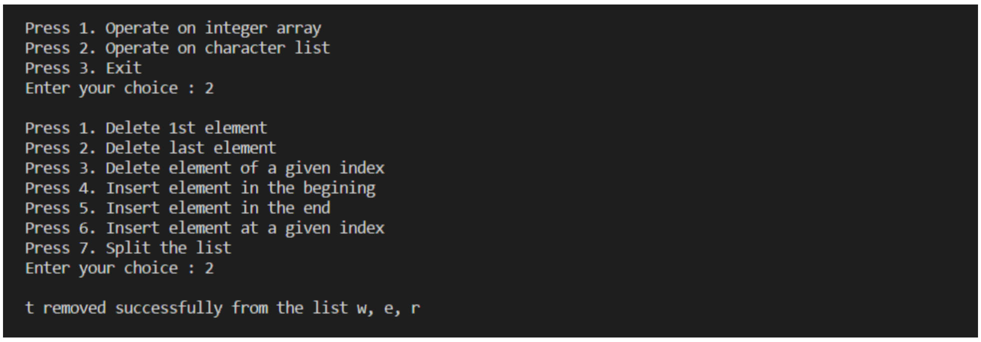
Operation on character list :

1. Deletion of First element :



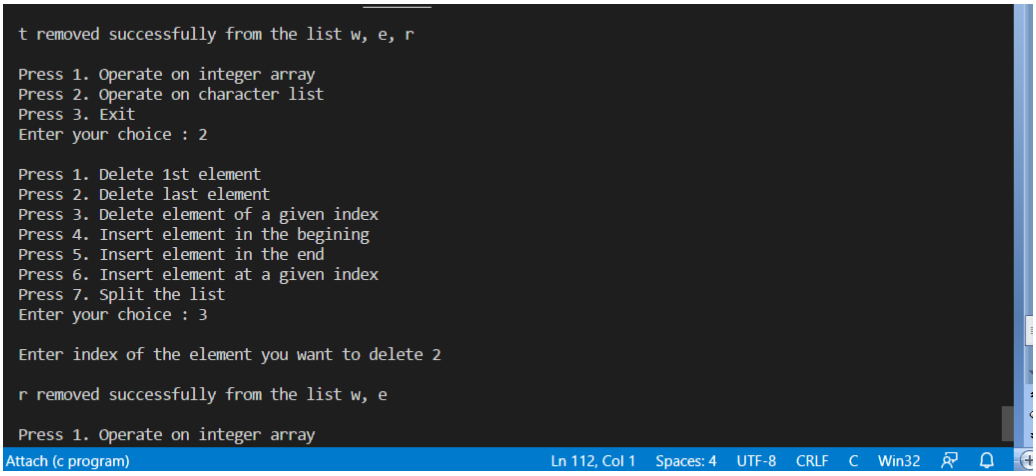
```
Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 2

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 1

q removed successfully from the list w, e, r, t

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
```

2. Deletion of last element :



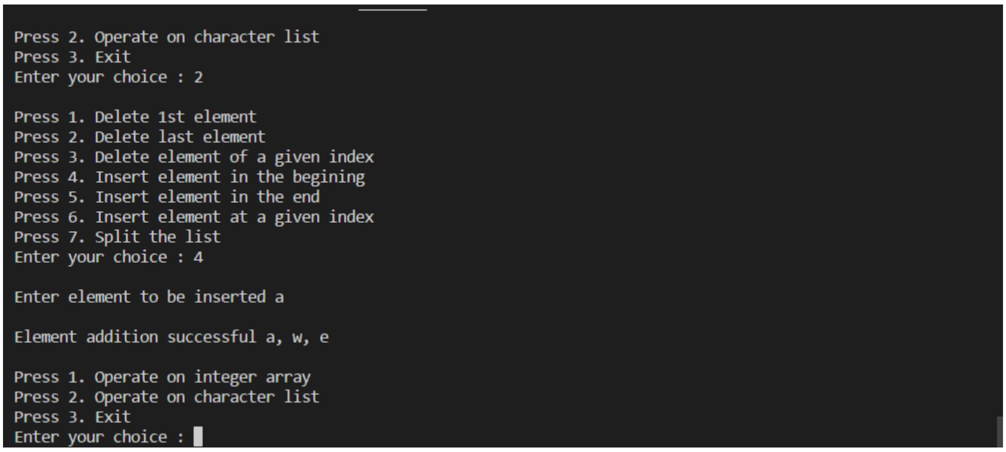
```
Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 2

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 2

t removed successfully from the list w, e, r
```

3. Delete element of a given index :



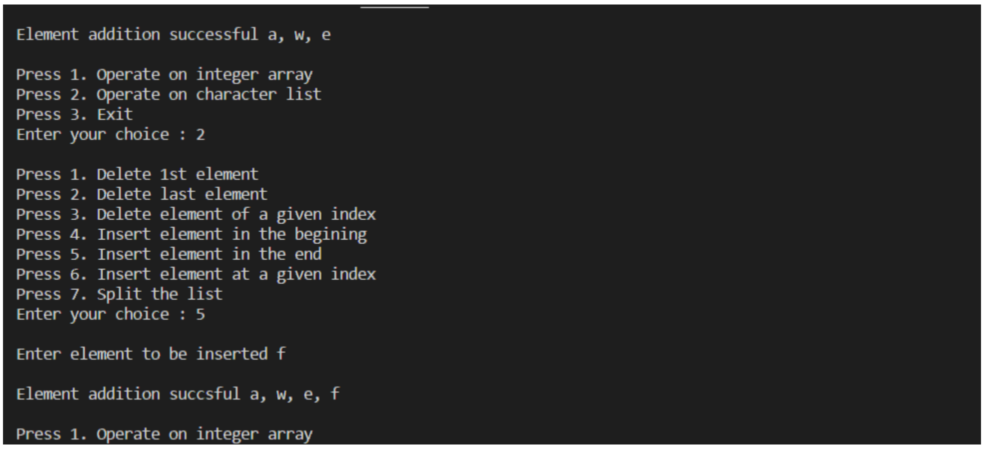
```
t removed successfully from the list w, e, r

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 2

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 3

Enter index of the element you want to delete 2

r removed successfully from the list w, e

Press 1. Operate on integer array
```

4. Insert at beginning



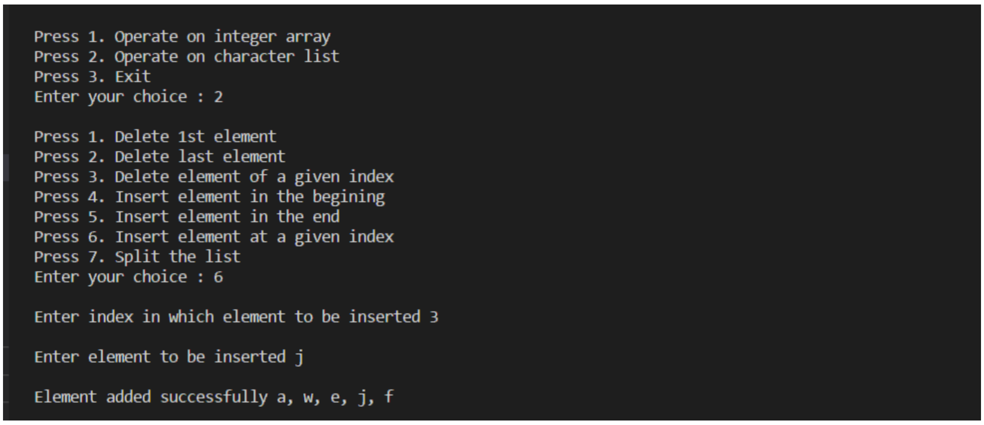
```
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 2

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 4

Enter element to be inserted a

Element addition successful a, w, e

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice :
```

5. Insert at end



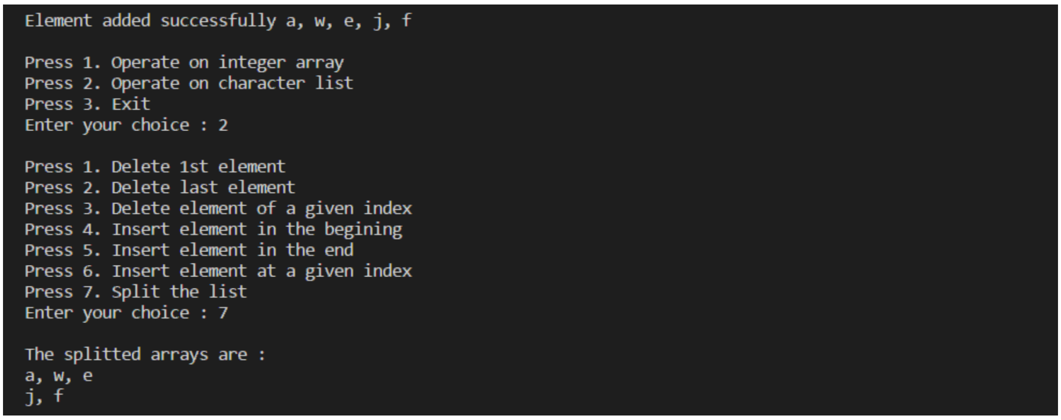
```
Element addition successful a, w, e

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 2

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 5

Enter element to be inserted f

Element addition succsful a, w, e, f

Press 1. Operate on integer array
```

6. Insert at given index



```
Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 2

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 6

Enter index in which element to be inserted 3

Enter element to be inserted j

Element added successfully a, w, e, j, f
```

7. Split the list



```
Element added successfully a, w, e, j, f

Press 1. Operate on integer array
Press 2. Operate on character list
Press 3. Exit
Enter your choice : 2

Press 1. Delete 1st element
Press 2. Delete last element
Press 3. Delete element of a given index
Press 4. Insert element in the begining
Press 5. Insert element in the end
Press 6. Insert element at a given index
Press 7. Split the list
Enter your choice : 7

The splitted arrays are :
a, w, e
j, f
```

2. **Write a program in C to implement queue and operations on it (enqueue, dequeue, location of front and back pointers, display elements). Get the input from the key board for creating queue.**

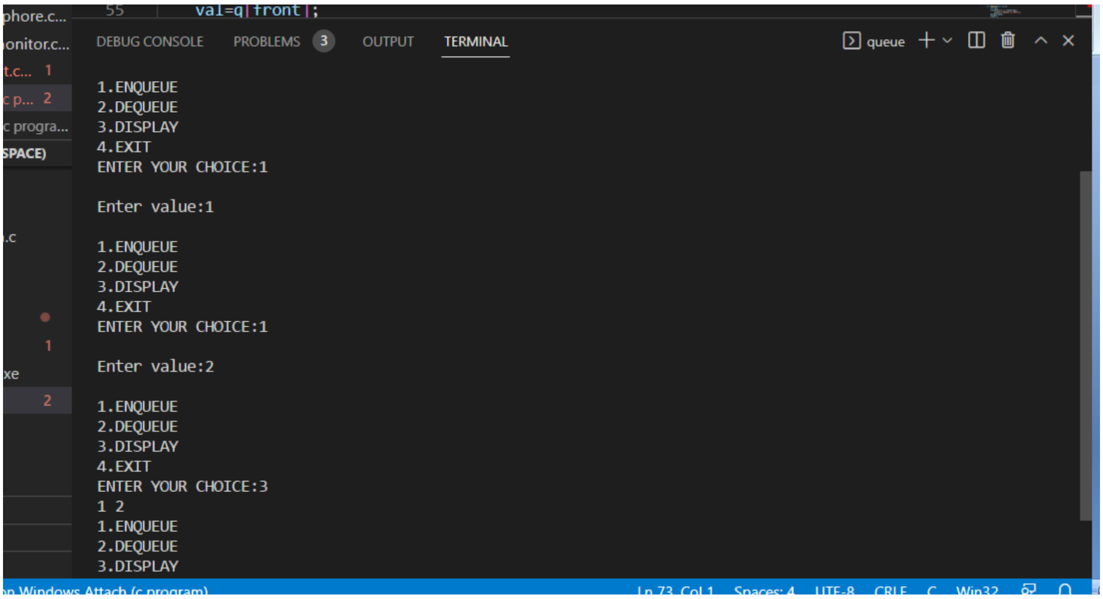
Code:

```
#include<stdio.h>
void enqueue(int[],int);
int dequeue(int[]);
void display(int[]);

#define SIZE 5
int q[SIZE],front=-1,rear=-1;
void main(){

  int val,ch;
  do{
    printf("\n1.ENQUEUE\n2.DEQUEUE\n3.DISPLAY\n4.EXIT\nENTER YOUR CHOICE:");
    scanf("%d",&ch);
    switch(ch){

      case 1:
        printf("\nEnter value:");
        scanf("%d",&val);
        enqueue(q,val);
      break;

      case 2:
        val=dequeue(q);
        printf("\nDELETED VALUE: %d",val);
      break;

      case 3:
        display(q);
      break;

      default:
```

```
                printf("\nInvalid input");

        }
    }while(ch!=4);
}

void enqueue(int q[], int val){
    if(rear==SIZE-1)
        printf("\nQUEUE IS FULL");
    else if(front==-1){
        front++;
        rear++;
    }
    else
        rear++;

    q[rear]=val;
}

int dequeue(int q[]){
    int val;
    if(front==-1 || front>rear)
        printf("\nQUEUE IS EMPTY");
    val=q[front];
    front++;
    if(front>rear){
        front=-1;
        rear=-1;
    }
    return val;
}

void display(int q[]){
    int i;
    if(front==-1 || front>rear)
        printf("\nQUEUE IS EMPTY");
    else{
        for(i=front;i<=rear;i++)
            printf("%d ",q[i]);
    }
}
```

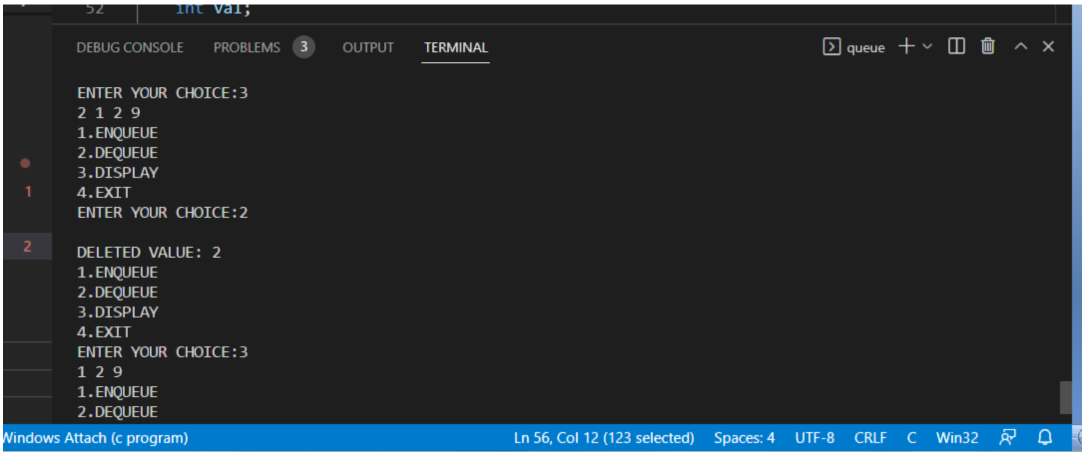
Output:

1. enqueue



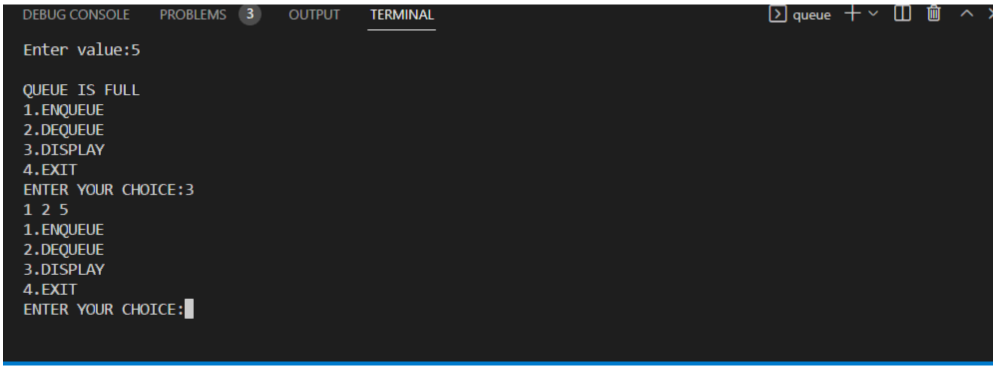
```
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:1

Enter value:1

1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:1

Enter value:2

1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:3
1 2
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
```

2. dequeue



```
ENTER YOUR CHOICE:3
2 1 2 9
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:2

DELETED VALUE: 2
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:3
1 2 9
1.ENQUEUE
2.DEQUEUE
```

3. display elements



```
Enter value:5

QUEUE IS FULL
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:3
1 2 5
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:
```

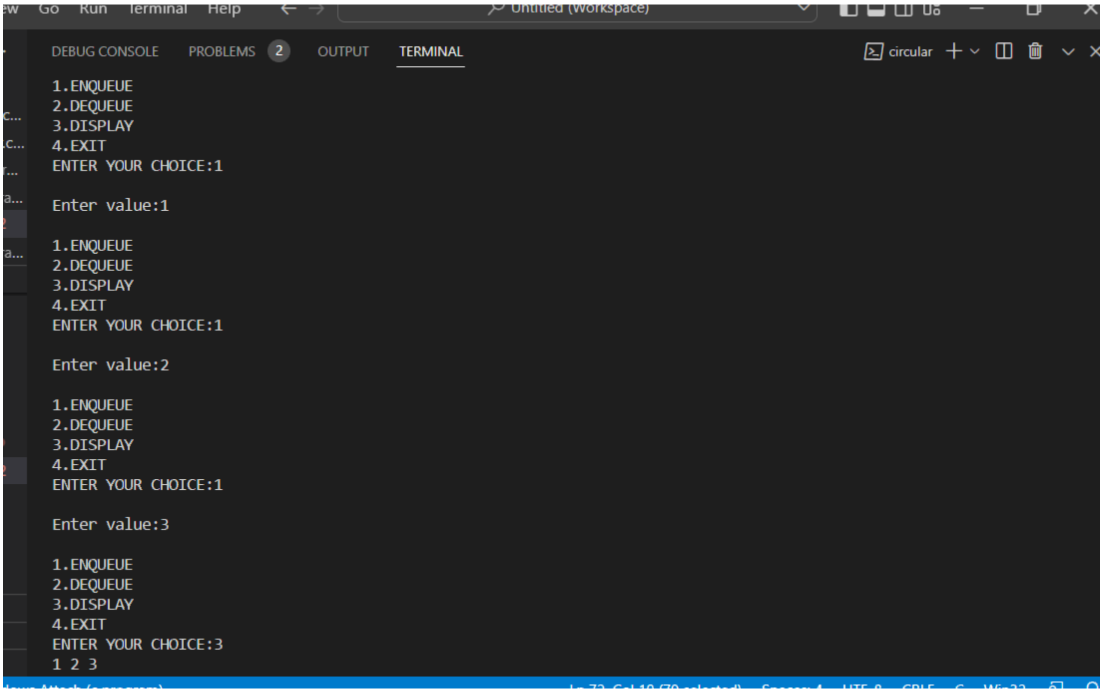
3. **Write a program in C to implement circular queue and operations on it (enqueue, dequeue, location of front and back pointers, display elements). Get the input from the key board for creating queue.**

Code:-

```
#include<stdio.h>
void enqueue(int[],int);
int dequeue(int[]);
void display(int[]);

#define SIZE 5
int q[SIZE],front=-1,rear=-1;
void main(){
   int val,ch;

   do{
      printf("\n1.ENQUEUE\n2.DEQUEUE\n3.DISPLAY\n4.EXIT\nENTER YOUR CHOICE:");
      scanf("%d",&ch);
      switch(ch){
         case 1:
            printf("\nEnter value:");
            scanf("%d",&val);
            enqueue(q,val);
         break;

         case 2:
         val=dequeue(q);
            printf("\nDELETED VALUE: %d",val);
         break;

         case 3:
            display(q);
         break;

         case 4:
         break;

         default:
            printf("Wrong input");
      }
   }while(ch!=4);
}

void enqueue(int q[], int val){

   if(front==-1 && rear==-1){
         front=rear=0;
         q[rear]=val;
   }
      else if ((rear+1)%SIZE==front ){
         printf("Queue overflown");
      }
      else{
        rear=(rear+1)%SIZE;
        q[rear]=val;
      }

   }

int dequeue(int q[]){
   int val;
   if(front==-1 || (front==-1 && rear==-1))
      printf("\nQUEUE IS EMPTY");
   else{
      val=q[front];
      if(front==rear)
         front=rear=-1;
      else
         front=(front+1)%SIZE;
      return val;
   }
}
void display(int q[]){
   int i;
   if(front==-1 || (front==-1 && rear==-1))
      printf("\nQUEUE IS EMPTY");
   else{
      for(i=front;i<=rear;i++)
         printf("%d ",q[i]);
   }
}
```

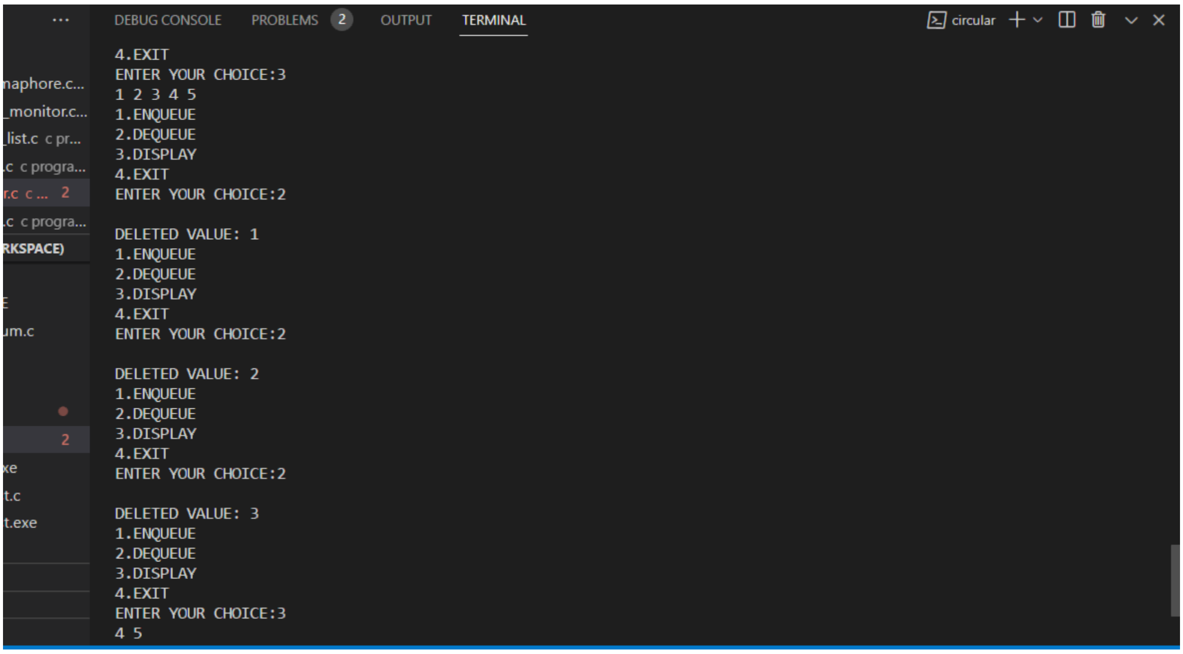
Output:-

1. enqueue



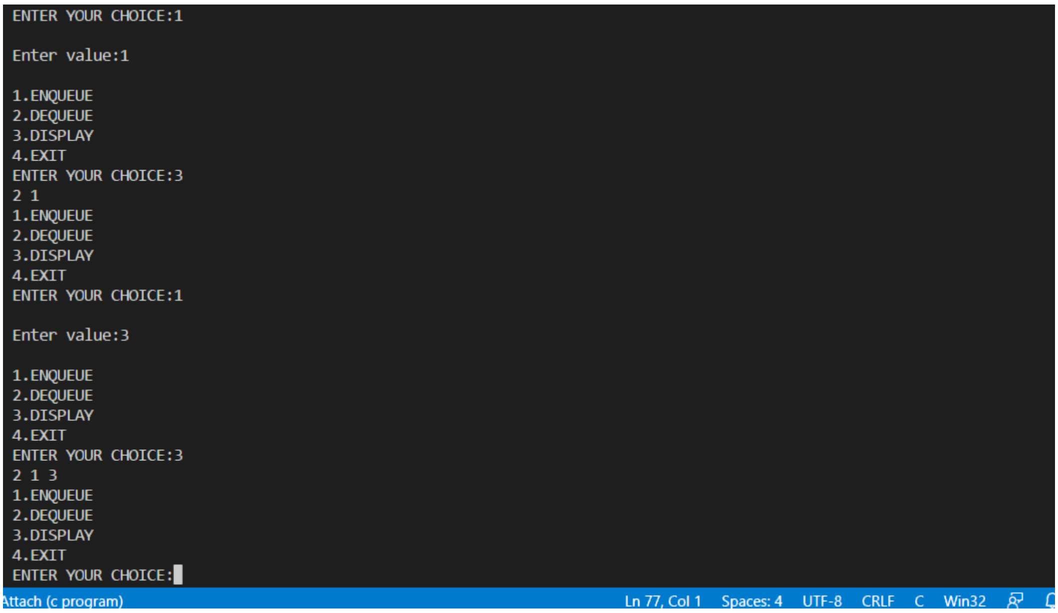
```
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:1

Enter value:1

1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:1

Enter value:2

1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:1

Enter value:3

1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:3
1 2 3
```

2. **dequeue**



```
4.EXIT
ENTER YOUR CHOICE:3
1 2 3 4 5
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:2

DELETED VALUE: 1
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:2

DELETED VALUE: 2
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:2

DELETED VALUE: 3
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:3
4 5
```

3. Display



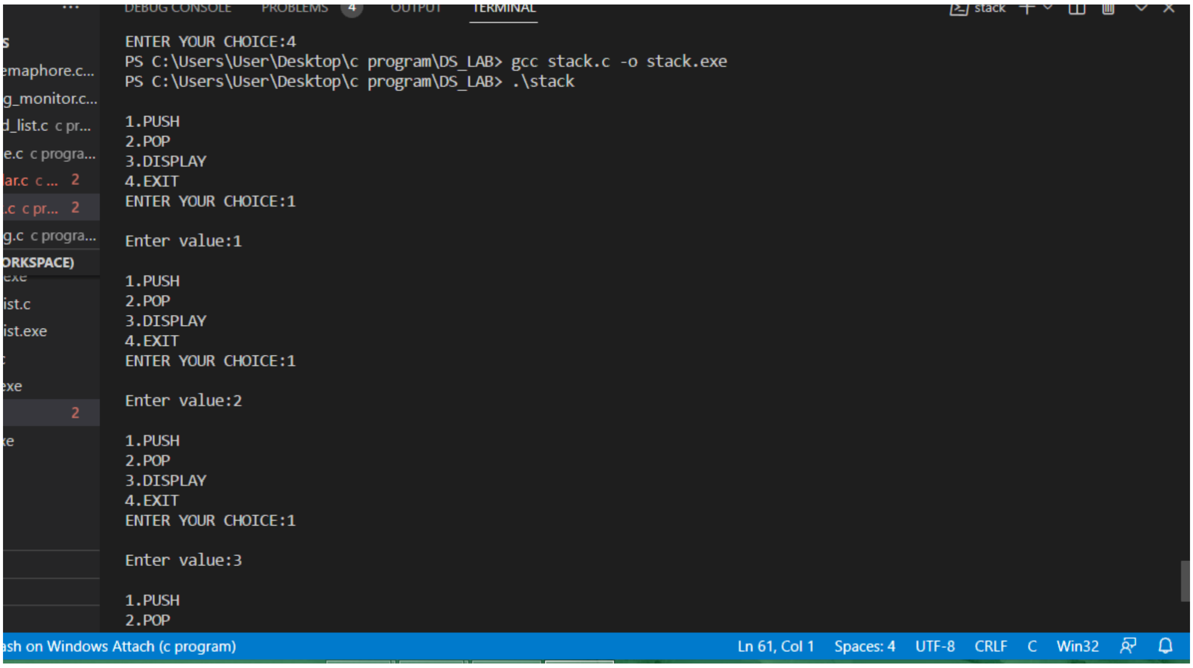
```
ENTER YOUR CHOICE:1

Enter value:1

1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:3
2 1
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:1

Enter value:3

1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:3
2 1 3
1.ENQUEUE
2.DEQUEUE
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:
```

4. Write a program in C to implement stack and operations on the it (push, pop, top element, display elements).

```
#include<stdio.h>
void push(int[],int);
int pop(int[]);
void display(int[]);

#define MAX 5
int s[MAX],top=-1;
void main(){
   int val,ch;
   do{
      printf("\n1.PUSH\n2.POP\n3.DISPLAY\n4.EXIT\nENTER YOUR CHOICE:");
      scanf("%d",&ch);
      switch(ch){
         case 1:
            printf("\nEnter value:");
            scanf("%d",&val);
            push(s,val);
         break;

         case 2:
            val=pop(s);
            printf("\nDELETED VALUE: %d",val);
         break;

         case 3:
                display(s);
            break;
        }
    }while(ch!=4);
}

void push(int s[], int val){
    if(top==MAX-1)
        printf("\nOVERFLOW");
    else{
        top++;
        s[top]=val;
    }
}

int pop(int s[]){
    int val;
    if(top==-1)
        printf("\nUNDERFLOW");
    else{
        val=s[top];
        top--;
    }
    return val;
}

void display(int s[]){
    int i;
    if(top==-1)
        printf("\nUNDERFLOW");
    else{
        for(i=top;i>=0;i--)
            printf("%d ",s[i]);
    }
}
```

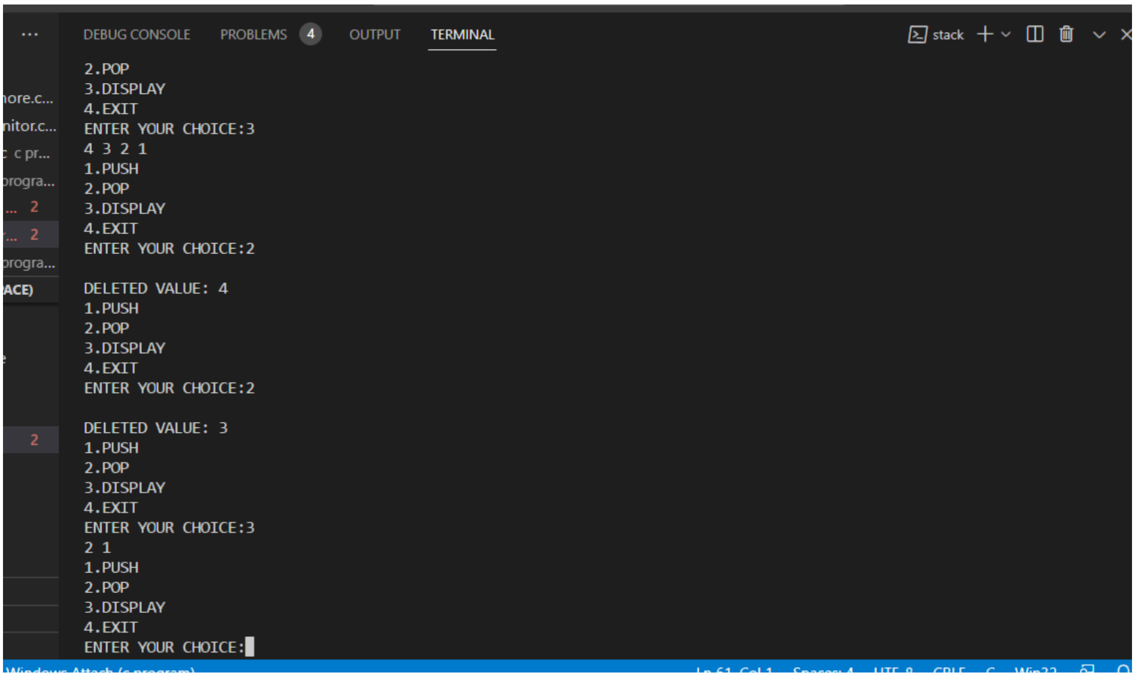
Output:-

1. push operation



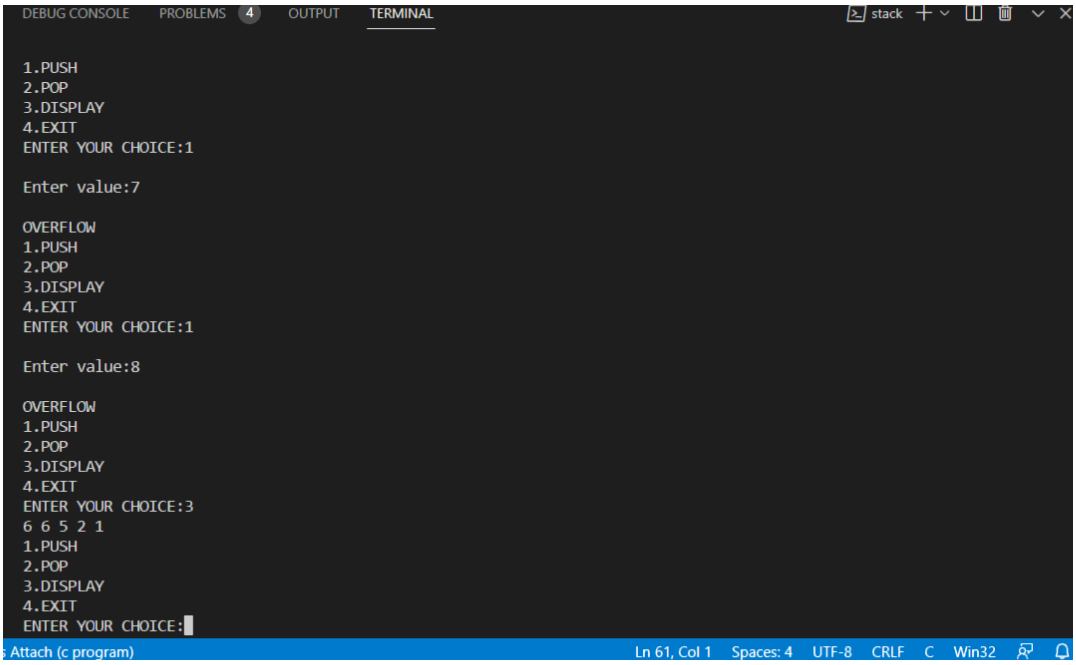
```
ENTER YOUR CHOICE:4
PS C:\Users\User\Desktop\c program\DS_LAB> gcc stack.c -o stack.exe
PS C:\Users\User\Desktop\c program\DS_LAB> .\stack

1.PUSH
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:1

Enter value:1

1.PUSH
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:1

Enter value:2

1.PUSH
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:1

Enter value:3

1.PUSH
2.POP
```

2. Pop Operations



```
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:3
4 3 2 1
1.PUSH
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:2

DELETED VALUE: 4
1.PUSH
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:2

DELETED VALUE: 3
1.PUSH
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:3
2 1
1.PUSH
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:
```

3. Display



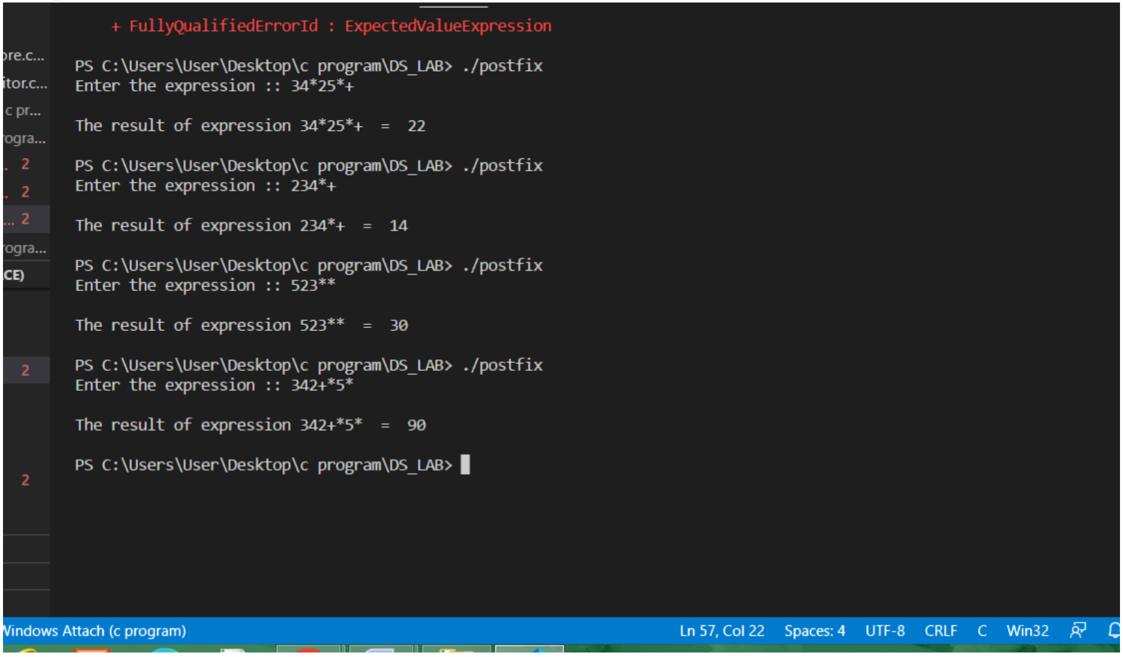
```
1.PUSH
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:1

Enter value:7

OVERFLOW
1.PUSH
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:1

Enter value:8

OVERFLOW
1.PUSH
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:3
6 6 5 2 1
1.PUSH
2.POP
3.DISPLAY
4.EXIT
ENTER YOUR CHOICE:
```

5. **Write a program in C to evaluate the postfix form of an algebraic expression using stack. Get the input from the keyboard.**

```
 #include<stdio.h>
int stack[20];
int top = -1;

void push(int x)
{
   stack[++top] = x;
}

int pop()
{
   return stack[top--];
}

int main()
{
   char exp[20];
   char *e;
   int n1,n2,n3,num;
   printf("Enter the expression :: ");
   scanf("%s",exp);
```

```
  e = exp;
  while(*e != '\0')
  {
    if(isdigit(*e))
    {
      num = *e - 48;
      push(num);
    }
    else
    {
      n1 = pop();
      n2 = pop();
      switch(*e)
      {
      case '+':
      {
        n3 = n1 + n2;
        break;
      }
      case '-':
      {
        n3 = n2 - n1;
        break;
      }
      case '*':
      {
        n3 = n1 * n2;
        break;
      }
      case '/':
      {
        n3 = n2 / n1;
        break;
      }
      }
      push(n3);
    }
    e++;
  }
  printf("\nThe result of expression %s  =  %d\n\n",exp,pop());
  return 0;
}
```

Output:-



```
    + FullyQualifiedErrorId : ExpectedValueExpression

PS C:\Users\User\Desktop\c program\DS_LAB> ./postfix
Enter the expression :: 34*25*+

The result of expression 34*25*+  =  22

PS C:\Users\User\Desktop\c program\DS_LAB> ./postfix
Enter the expression :: 234*+

The result of expression 234*+  =  14

PS C:\Users\User\Desktop\c program\DS_LAB> ./postfix
Enter the expression :: 523**

The result of expression 523**  =  30

PS C:\Users\User\Desktop\c program\DS_LAB> ./postfix
Enter the expression :: 342+*5*

The result of expression 342+*5*  =  90

PS C:\Users\User\Desktop\c program\DS_LAB> 
```