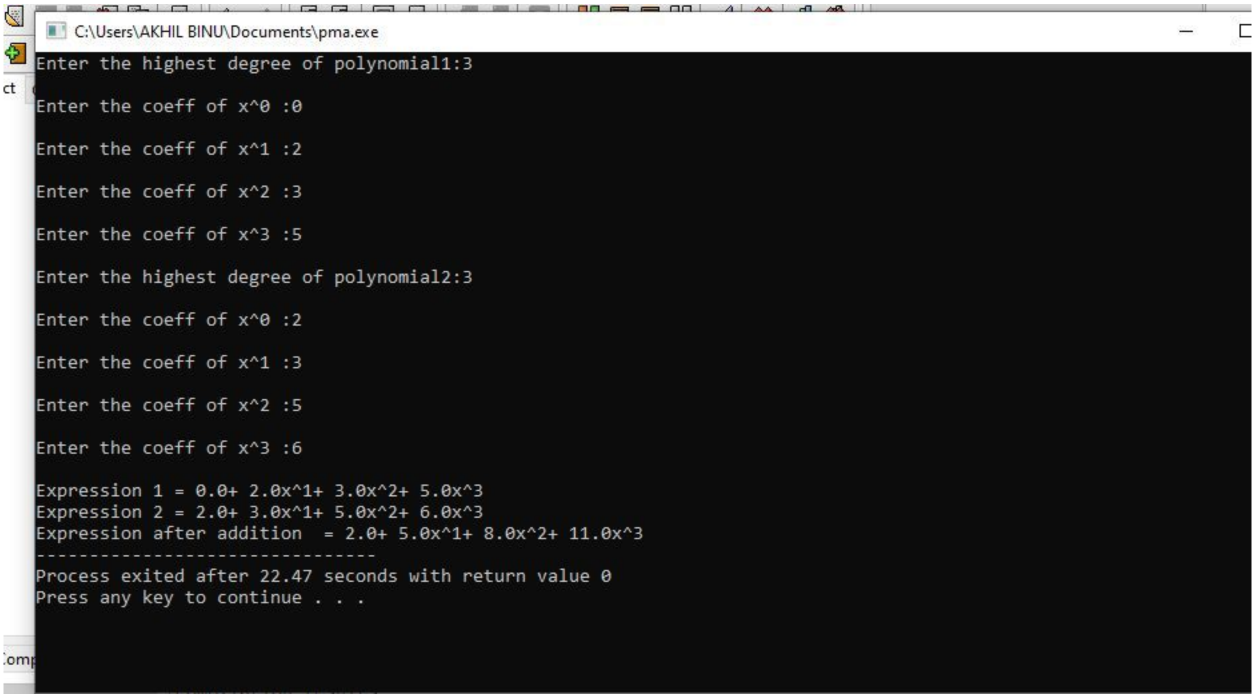
1) Write a program to read two polynomials and store them in an array. Calculate the sum of the two polynomials and display the first polynomial, second polynomial and the resultant Polynomial ?

Ans:

```
       .#include<stdio.h>
#include<math.h>
struct poly
{
float coeff;
int exp;
};
struct poly a[50],b[50],c[50],d[50];
int main()
{
int i;
int deg1,deg2;
int k=0,l=0,m=0;
printf("Enter the highest degree of polynomial1:");
scanf("%d",&deg1);
for(i=0;i<=deg1;i++)
{
printf("\nEnter the coeff of x^%d :",i);
scanf("%f",&a[i].coeff);
a[k++].exp = i;
}

printf("\nEnter the highest degree of polynomial2:");
scanf("%d",&deg2);
for(i=0;i<=deg2;i++)
{
printf("\nEnter the coeff of x^%d :",i);
scanf("%f",&b[i].coeff);
b[l++].exp = i;
}
printf("\nExpression 1 = %.1f",a[0].coeff);
for(i=1;i<=deg1;i++)
{
printf("+ %.1fx^%d",a[i].coeff,a[i].exp);
}
printf("\nExpression 2 = %.1f",b[0].coeff);
for(i=1;i<=deg2;i++)
{
printf("+ %.1fx^%d",b[i].coeff,b[i].exp);
}
if(deg1>deg2)
{
for(i=0;i<=deg2;i++)
{
c[m].coeff = a[i].coeff + b[i].coeff;
c[m].exp = a[i].exp;
m++;
}
for(i=deg2+1;i<=deg1;i++)
{
c[m].coeff = a[i].coeff;
c[m].exp = a[i].exp;
m++;
}
}
else
{
for(i=0;i<=deg1;i++)
{
c[m].coeff = a[i].coeff + b[i].coeff;
c[m].exp = a[i].exp;
m++;
}
for(i=deg1+1;i<=deg2;i++)
{

c[m].coeff = b[i].coeff;
c[m].exp = b[i].exp;
m++;
}
}
printf("\nExpression after addition = %.1f",c[0].coeff);
for(i=1;i<m;i++)
{
printf("+ %.1fx^%d",c[i].coeff,c[i].exp);
}
return 0;
}
```



```
Enter the highest degree of polynomial1:3

Enter the coeff of x^0 :0

Enter the coeff of x^1 :2

Enter the coeff of x^2 :3

Enter the coeff of x^3 :5

Enter the highest degree of polynomial2:3

Enter the coeff of x^0 :2

Enter the coeff of x^1 :3

Enter the coeff of x^2 :5

Enter the coeff of x^3 :6

Expression 1 = 0.0+ 2.0x^1+ 3.0x^2+ 5.0x^3
Expression 2 = 2.0+ 3.0x^1+ 5.0x^2+ 6.0x^3
Expression after addition  = 2.0+ 5.0x^1+ 8.0x^2+ 11.0x^3
--------------------------------
Process exited after 22.47 seconds with return value 0
Press any key to continue . . .
```

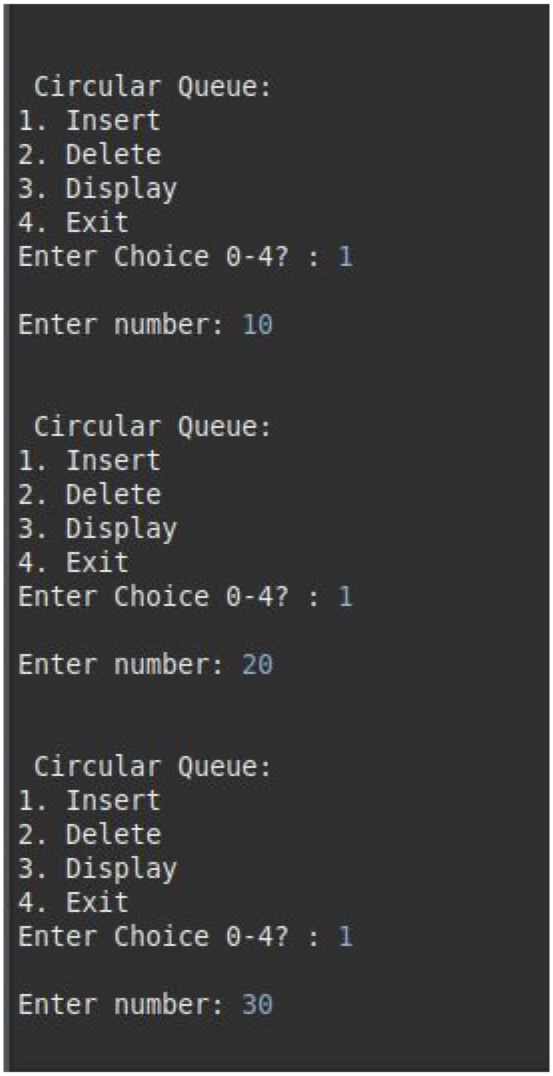
2) Implement a circular queue using arrays with the operations ?

Ans:

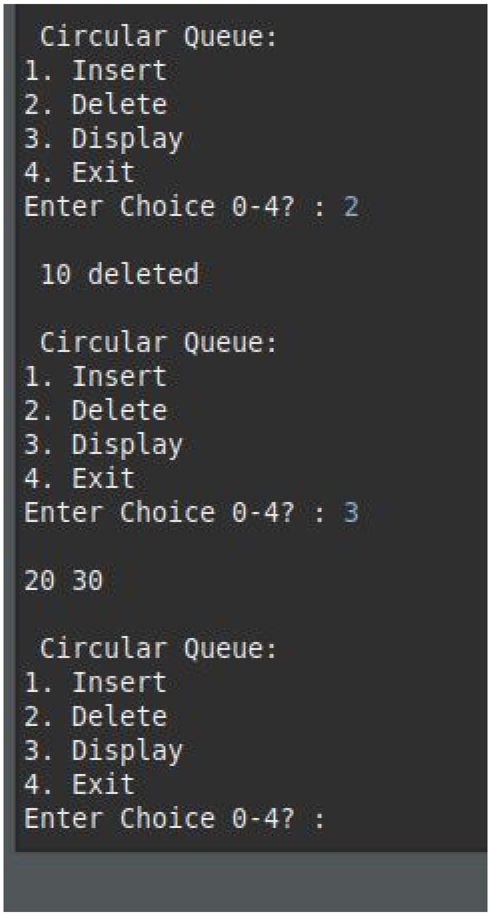
```
      #include <stdio.h>
#define size 5
void insertq(int[], int);
void deleteq(int[]);
void display(int[]);
int front = - 1;
int rear = - 1;
int main()
{
int n, ch;
int queue[size];
do
{
printf("\n\n Circular Queue:\n1. Insert \n2. Delete\n3. Display\n0. Exit");
printf("\nEnter Choice 0-3? : ");
scanf("%d", &ch);
switch (ch)
{
case 1:

printf("\nEnter number: ");
scanf("%d", &n);
```

```
insertq(queue, n);
break;
case 2:
deleteq(queue);
break;
case 3:
display(queue);
break;
}
}while (ch != 0);
}

void insertq(int queue[], int item)
{
if ((front == 0 && rear == size - 1) || (front == rear + 1))
{
printf("queue is full");
return;
}
else if (rear == - 1)
{
rear++;
front++;
}
else if (rear == size - 1 && front > 0)
{
rear = 0;
}
else
{
rear++;
}
queue[rear] = item;
}
void display(int queue[])
{
int i;
printf("\n");
if (front > rear)

{
for (i = front; i < size; i++)
{
printf("%d ", queue[i]);
```

```
}
for (i = 0; i <= rear; i++)
printf("%d ", queue[i]);
}
else
{
for (i = front; i <= rear; i++)
printf("%d ", queue[i]);
}
}
void deleteq(int queue[])
{
if (front == - 1)
{
printf("Queue is empty ");
}
else if (front == rear)
{
printf("\n %d deleted", queue[front]);
front = - 1;
rear = - 1;
}
else
{
printf("\n %d deleted", queue[front]);
front++;
}
}
```



```
 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
Enter Choice 0-4? : 1

Enter number: 10


 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
Enter Choice 0-4? : 1

Enter number: 20


 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
Enter Choice 0-4? : 1

Enter number: 30
```



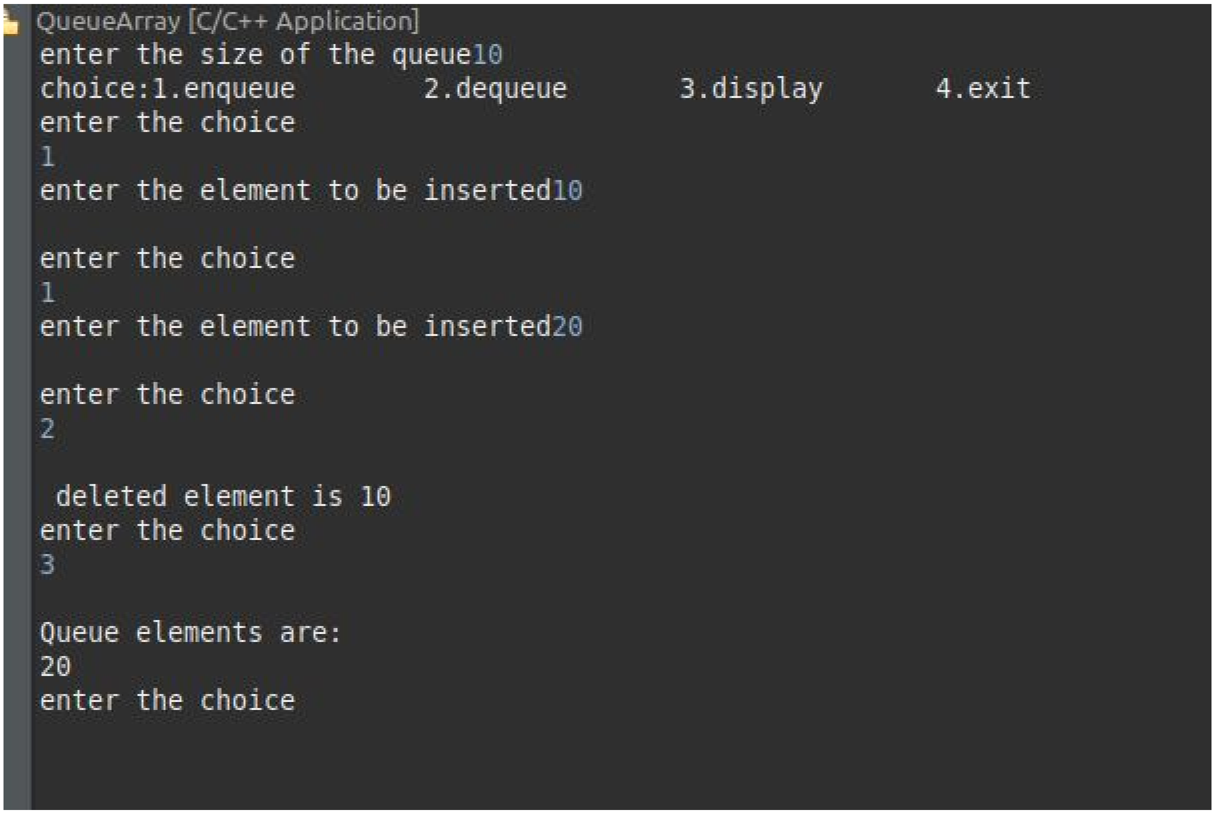
```
 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
Enter Choice 0-4? : 2

 10 deleted

 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
Enter Choice 0-4? : 3

20 30

 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
Enter Choice 0-4? :
```

3) Implement a Queue using arrays with the operations ?

Ans:

```
    #include<stdio.h>
int queue[100],maxsize,choice,front=-1,rear=-1,item,i;
void enqueue();
void dequeue();
void display();
int main()
{
printf("enter the size of the queue");
scanf("%d",&maxsize);
printf("choice:1.enqueue\t2.dequeue\t3.display\t4.exit");
do
{

printf("\nenter the choice\n");
scanf("%d",&choice);
if(choice==1)
{
enqueue();
}
else
if(choice==2)
{
dequeue();
}
else
if(choice==3)
{
display();
}
else
if(choice==4)
{
printf("\n\t EXIT POINT ");
}

}
while(choice!=4);
return 0;
}
void enqueue()
{
if(rear==maxsize-1)
{
printf("\nqueue is full\n");
}
else
if(front==-1)
{
front=0;

}

printf("enter the element to be inserted");
scanf("%d",&item);

rear=rear+1;
queue[rear]=item;

}
void dequeue()
```

```
{
if(front==-1 && rear==-1)
{
printf("the queue is empty");
return;
}
else
{
printf("\n deleted element is %d",queue[front]);
front++;
}
}
void display(){
if(rear == -1)
printf("\nQueue is Empty!!!");
else{
int i;
printf("\nQueue elements are:\n");
for(i=front; i<=rear; i++)
printf("%d\t",queue[i]);
}
}
```



```
enter the size of the queue10
choice:1.enqueue        2.dequeue       3.display       4.exit
enter the choice
1
enter the element to be inserted10

enter the choice
1
enter the element to be inserted20

enter the choice
2

 deleted element is 10
enter the choice
3

Queue elements are:
20
enter the choice
```

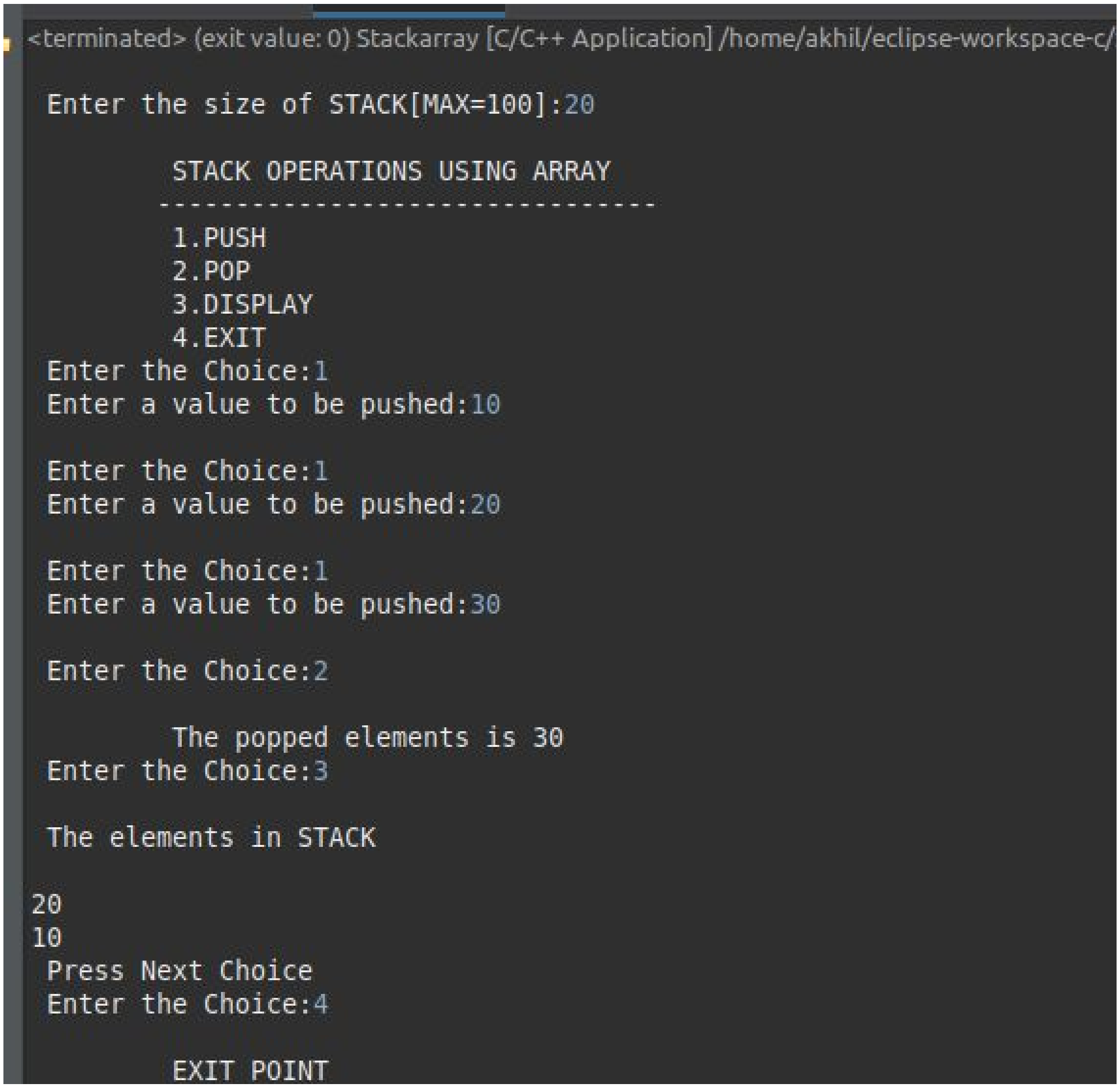
4) Implement a Stack using arrays with the operations ?

Ans :

```
        #include<stdio.h>
int stack[100],choice,n,top,x,i;
void push(void);
void pop(void);
void display(void);
int main()
{
//clrscr();
top=-1;
printf("\n Enter the size of STACK[MAX=100]:");
scanf("%d",&n);
printf("\n\t STACK OPERATIONS USING ARRAY");
printf("\n\t--------------------------------");
printf("\n\t 1.PUSH\n\t 2.POP\n\t 3.DISPLAY\n\t 4.EXIT");
do
{
printf("\n Enter the Choice:");
scanf("%d",&choice);
switch(choice)
{
case 1:
{
push();
break;
}
case 2:
{
pop();
break;
}
case 3:
{
display();
break;
}
case 4:
{
printf("\n\t EXIT POINT ");
break;
}
default:
```

```
{
printf ("\n\t Please Enter a Valid Choice(1/2/3/4)");
}
}
}
while(choice!=4);
return 0;
}
void push()
{
if(top>=n-1)
{
printf("\n\tSTACK is over flow");
}
else
{
printf(" Enter a value to be pushed:");
scanf("%d",&x);
top++;
stack[top]=x;
}
}
void pop()
{
if(top<=-1)
{
printf("\n\t Stack is under flow");
}
else
{
printf("\n\t The popped elements is %d",stack[top]);
top--;
}
}
void display()
{
if(top>=0)
{
printf("\n The elements in STACK \n");
for(i=top; i>=0; i--)
printf("\n%d",stack[i]);

printf("\n Press Next Choice");
```

}
else
{
printf("\n The STACK is empty");
}
}



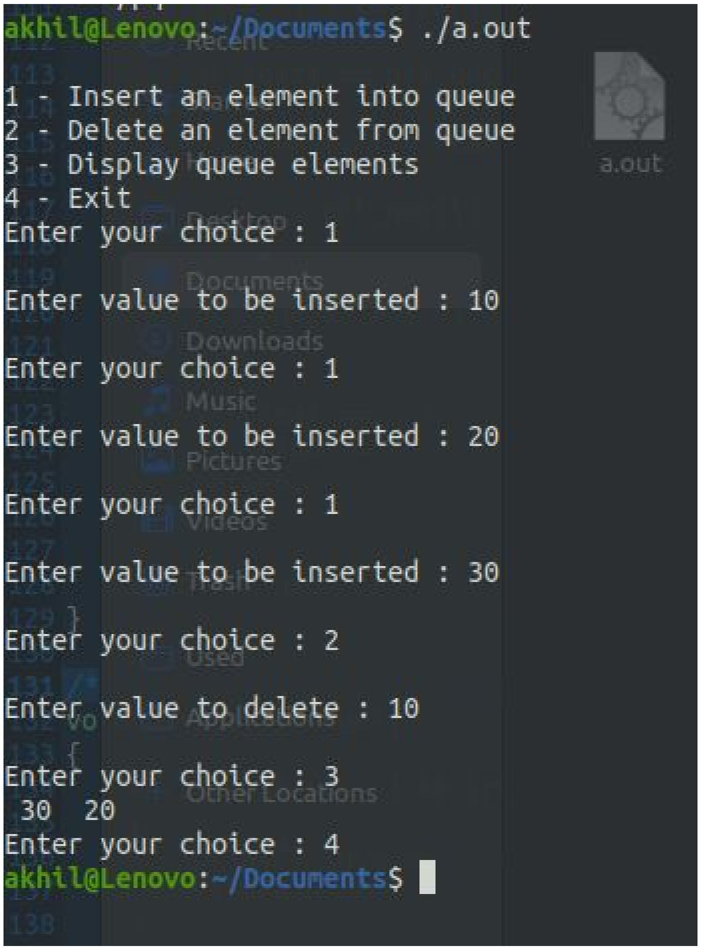
```
<terminated> (exit value: 0) Stackarray [C/C++ Application] /home/akhil/eclipse-workspace-c/

 Enter the size of STACK[MAX=100]:20

         STACK OPERATIONS USING ARRAY
        --------------------------------
         1.PUSH
         2.POP
         3.DISPLAY
         4.EXIT
 Enter the Choice:1
 Enter a value to be pushed:10

 Enter the Choice:1
 Enter a value to be pushed:20

 Enter the Choice:1
 Enter a value to be pushed:30

 Enter the Choice:2

         The popped elements is 30
 Enter the Choice:3

 The elements in STACK

20
10
 Press Next Choice
 Enter the Choice:4

         EXIT POINT
```

5) Implement a Priority Queue using arrays with the operations: ?

Ans :

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 5
void insert_by_priority(int);
void delete_by_priority(int);
void create();
void check(int);
void display_pqueue();
int pri_que[MAX];
int front, rear;
void main()
{
    int n, ch;
    printf("\n1 - Insert an element into queue");
    printf("\n2 - Delete an element from queue");
    printf("\n3 - Display queue elements");
    printf("\n4 - Exit");
    create();
    while (1)
    {
        printf("\nEnter your choice : ");
        scanf("%d", &ch);

        switch (ch)
        {
        case 1:
            printf("\nEnter value to be inserted : ");
            scanf("%d",&n);
            insert_by_priority(n);
            break;
        case 2:
            printf("\nEnter value to delete : ");
            scanf("%d",&n);
            delete_by_priority(n);
            break;
        case 3:
            display_pqueue();
            break;
        case 4:
            exit(0);
        default:
            printf("\nChoice is incorrect, Enter a correct choice");
        }
    }
}
void create()
{
    front = rear = -1;
}
```

```
void insert_by_priority(int data)
{
    if (rear >= MAX - 1)
    {
        printf("\nQueue overflow no more elements can be inserted");
        return;
    }
    if ((front == -1) && (rear == -1))
    {
        front++;
        rear++;
        pri_que[rear] = data;
        return;
    }
    else
        check(data);
    rear++;
}
void check(int data)
{
    int i,j;

    for (i = 0; i <= rear; i++)
    {
        if (data >= pri_que[i])
        {
            for (j = rear + 1; j > i; j--)
            {
                pri_que[j] = pri_que[j - 1];
            }
            pri_que[i] = data;
            return;
        }
    }
    pri_que[i] = data;
}
void delete_by_priority(int data)
{
    int i;

    if ((front==-1) && (rear==-1))
    {
        printf("\nQueue is empty no elements to delete");
        return;
    }

    for (i = 0; i <= rear; i++)
    {
        if (data == pri_que[i])
        {
            for (; i < rear; i++)
            {
                pri_que[i] = pri_que[i + 1];
            }

        pri_que[i] = -99;
        rear--;

        if (rear == -1)
            front = -1;
        return;
        }
    }
    printf("\n%d not found in queue to delete", data);
}

void display_pqueue()
{
    if ((front == -1) && (rear == -1))
    {
        printf("\nQueue is empty");
        return;
    }

    for (; front <= rear; front++)
    {
        printf(" %d ", pri_que[front]);
    }

    front = 0;
}
```



```
akhil@Lenovo:~/Documents$ ./a.out

1 - Insert an element into queue
2 - Delete an element from queue
3 - Display queue elements
4 - Exit
Enter your choice : 1

Enter value to be inserted : 10

Enter your choice : 1

Enter value to be inserted : 20

Enter your choice : 1

Enter value to be inserted : 30

Enter your choice : 2

Enter value to delete : 10

Enter your choice : 3
 30  20
Enter your choice : 4
akhil@Lenovo:~/Documents$ 
```

6) Implement a Double-Ended Queue (DEQUEUE) with the operations ?

Ans :

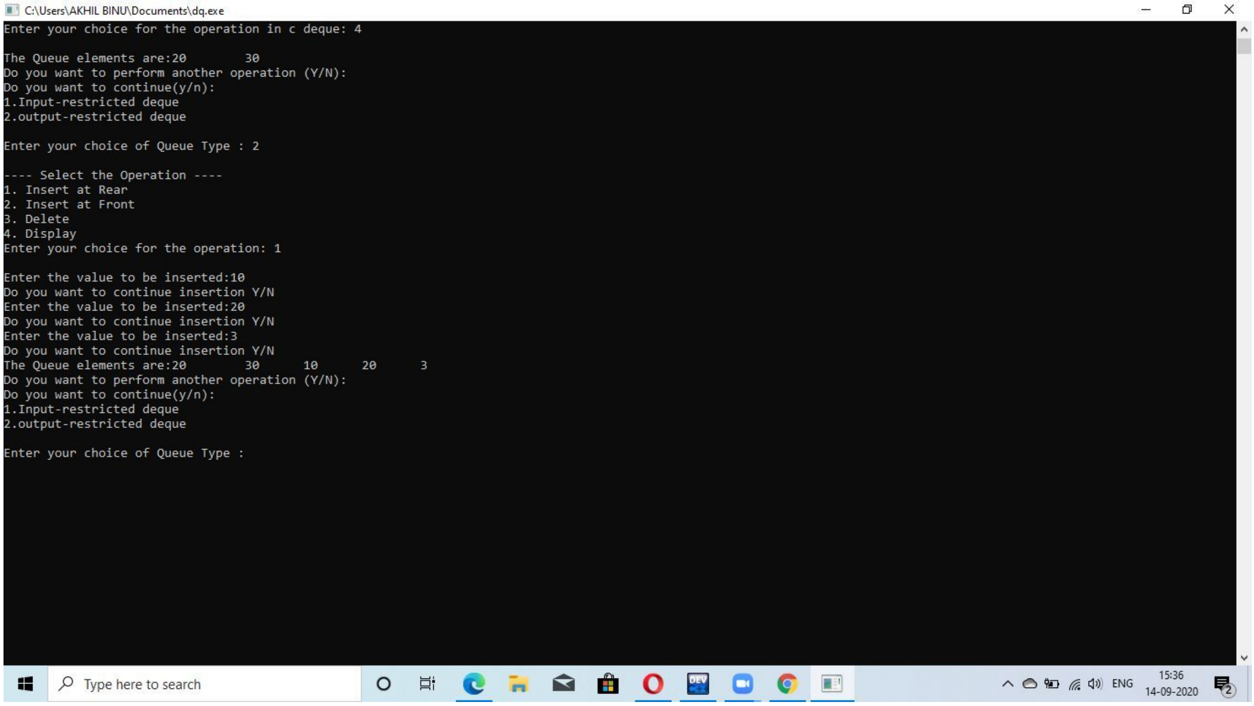
```
        #include<stdio.h>
#include<conio.h>
#define SIZE 100
void enQueue(int);
int deQueueFront();
int deQueueRear();
void enQueueRear(int);
void enQueueFront(int);
void display();
int queue[SIZE];
int rear = 0, front = 0;
int main()
{
char ch;
int choice1, choice2, value;
```

```
printf("\n******* Type of Double Ended Queue *******\n");
do
{
printf("\n1.Input-restricted deque \n");

printf("2.output-restricted deque \n");
printf("\nEnter your choice of Queue Type : ");
scanf("%d",&choice1);
switch(choice1)
{
case 1:
printf("\nSelect the Operation\n");
printf("1.Insert\n2.Delete from Rear\n3.Delete from Front\n4. Display");
do
{
printf("\nEnter your choice for the operation in c deque: ");
scanf("%d",&choice2);
switch(choice2)
{
case 1: enQueueRear(value);
display();
break;
case 2: value = deQueueRear();
printf("\nThe value deleted is %d",value);
display();
break;
case 3: value=deQueueFront();
printf("\nThe value deleted is %d",value);
display();
break;
case 4: display();
break;
default:printf("Wrong choice");
}
printf("\nDo you want to perform another operation (Y/N): ");
ch=getch();
}while(ch=='y'||ch=='Y');
getch();
break;
case 2 :
printf("\n---- Select the Operation ----\n");
printf("1. Insert at Rear\n2. Insert at Front\n3. Delete\n4. Display");
do
{
```

```
printf("\nEnter your choice for the operation: ");
scanf("%d",&choice2);
switch(choice2)

{
case 1: enQueueRear(value);
display();
break;
case 2: enQueueFront(value);
display();
break;
case 3: value = deQueueFront();
printf("\nThe value deleted is %d",value);
display();
break;
case 4: display();
break;
default:printf("Wrong choice");
}
printf("\nDo you want to perform another operation (Y/N): ");
ch=getch();
} while(ch=='y'||ch=='Y');
getch();
break ;
}
printf("\nDo you want to continue(y/n):");
ch=getch();
}while(ch=='y'||ch=='Y');
}
void enQueueRear(int value)
{
char ch;
if(front == SIZE/2)
{
printf("\nQueue is full!!! Insertion is not possible!!! ");
return;
}
do
{
printf("\nEnter the value to be inserted:");
scanf("%d",&value);
queue[front] = value;
front++;
printf("Do you want to continue insertion Y/N");
```

```
ch=getch();
}while(ch=='y');

}
void enQueueFront(int value)
{
char ch;
if(front==SIZE/2)
{
printf("\nQueue is full!!! Insertion is not possible!!!");
return;
}
do
{
printf("\nEnter the value to be inserted:");
scanf("%d",&value);
rear--;
queue[rear] = value;
printf("Do you want to continue insertion Y/N");
ch = getch();
}
while(ch == 'y');
}
int deQueueRear()
{
int deleted;
if(front == rear)
{
printf("\nQueue is Empty!!! Deletion is not possible!!!");
return 0;
}
front--;
deleted = queue[front+1];
return deleted;
}
int deQueueFront()
{
int deleted;
if(front == rear)
{
printf("\nQueue is Empty!!! Deletion is not possible!!!");
return 0;
}
rear++;
```

```
deleted = queue[rear-1];

return deleted;
}
void display()
{
int i;
if(front == rear)
printf("\nQueue is Empty!!! Deletion is not possible!!!");
else{
printf("\nThe Queue elements are:");
for(i=rear; i < front; i++)
{
printf("%d\t ",queue[i]);
}
}
}
```



```
Enter your choice for the operation in c deque: 4

The Queue elements are:20       30
Do you want to perform another operation (Y/N):
Do you want to continue(y/n):
1.Input-restricted deque
2.output-restricted deque

Enter your choice of Queue Type : 2

---- Select the Operation ----
1. Insert at Rear
2. Insert at Front
3. Delete
4. Display
Enter your choice for the operation: 1

Enter the value to be inserted:10
Do you want to continue insertion Y/N
Enter the value to be inserted:20
Do you want to continue insertion Y/N
Enter the value to be inserted:3
Do you want to continue insertion Y/N
The Queue elements are:20       30      10      20      3
Do you want to perform another operation (Y/N):
Do you want to continue(y/n):
1.Input-restricted deque
2.output-restricted deque

Enter your choice of Queue Type :
```

7) Using stack convert an infix expression to a postfix expression and evaluate the postfix Expression ?

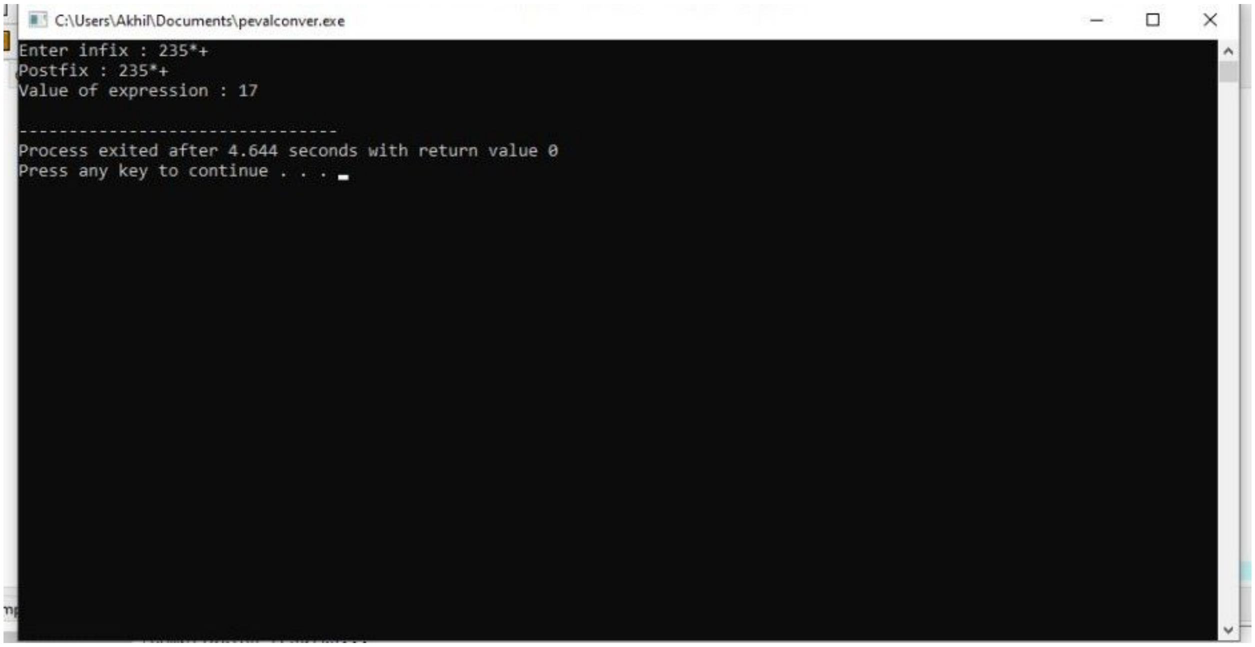
Ans :

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

```
#define BLANK ' '
#define TAB '\t'
#define MAX 50
void push(long int symbol);
long int pop();
void infix_to_postfix();
long int eval_post();
int priority(char symbol);
int isEmpty();
int white_space(char );
char infix[MAX], postfix[MAX];
long int stack[MAX];
int top;
int main()
{
        long int value;
        top=-1;
        printf("Enter infix : ");
        gets(infix);
        infix_to_postfix();
        printf("Postfix : %s\n",postfix);
        value=eval_post();
        printf("Value of expression : %ld\n",value);
        return 0;
}
void infix_to_postfix()
{
        unsigned int i,p=0;
        char next;
        char symbol;
        for(i=0;i<strlen(infix);i++)
        {
                symbol=infix[i];
                if(!white_space(symbol))
                {
                        switch(symbol)
                        {
                        case '(':
                                push(symbol);
                                break;
                        case ')':
                                while((next=pop())!='(')
                                        postfix[p++] = next;
                                break;
```

```
            case '+':
            case '-':
            case '*':
            case '/':
            case '%':
            case '^':
                while( !isEmpty( ) &&  priority(stack[top])>= priority(symbol) )
                    postfix[p++]=pop();
                push(symbol);
                break;
            default:
              postfix[p++]=symbol;
            }
        }
    }
    while(!isEmpty( ))
        postfix[p++]=pop();
    postfix[p]='\0';
}
int priority(char symbol)
{
    switch(symbol)
    {
    case '(':
        return 0;
    case '+':
    case '-':
        return 1;
    case '*':
    case '/':
    case '%':
        return 2;
    case '^':
        return 3;
    default :
        return 0;
    }
}
void push(long int symbol)
{
    if(top>MAX)
    {
        printf("Stack overflow\n");
        exit(1);
        }
        stack[++top]=symbol;
}
long int pop()
{
        if( isEmpty() )
        {
                printf("Stack underflow\n");
                exit(1);
        }
        return (stack[top--]);
}
int isEmpty()
{
        if(top==-1)
                return 1;
        else
                return 0;
}
int white_space(char symbol)
{
        if( symbol == BLANK || symbol == TAB )
                return 1;
        else
                return 0;
}
long int eval_post()
{
        long int a,b,temp,result;
        unsigned int i;

        for(i=0;i<strlen(postfix);i++)
        {
                if(postfix[i]<='9' && postfix[i]>='0')
                        push(postfix[i]-'0');
                else
                {
                        a=pop();
                        b=pop();
                        switch(postfix[i])
                        {
                        case '+':
                                temp=b+a; break;
                        case '-':
```

```
                temp=b-a;break;
            case '*':
                temp=b*a;break;
            case '/':
                temp=b/a;break;
            case '%':
                temp=b%a;break;
            case '^':
                temp=pow(b,a);
            }
            push(temp);
        }
    }
    result=pop();
    return result;
}
```



```
Enter infix : 235*+
Postfix : 235*+
Value of expression : 17

--------------------------------
Process exited after 4.644 seconds with return value 0
Press any key to continue . . .
```

8) Write a menu driven program for performing the following operations on a Linked List ?

Ans :

```
#include<stdlib.h>
#include <stdio.h>
void create();
void display();
void insert_begin();
void insert_end();
void insert_pos();
void delete_begin();
void delete_end();
void delete_pos();
struct node
{
int info;
struct node *next;
};
struct node *start=NULL;
int main()
{
int choice;
while(1){
printf("\n
MENU
");
printf("\n 1.Display ");
printf("\n 2.Insert at the beginning ");
printf("\n 3.Insert at the end ");
printf("\n 4.Insert at specified position
");
printf("\n 5.Delete from beginning
");
printf("\n 6.Delete from the end
");
printf("\n 7.Delete from specified position ");
printf("\n 8.Exit
\n");
printf("\n--------------------------------------\n");
printf("Enter your choice:\t");scanf("%d",&choice);
switch(choice)
{
case 1:
display();
break;
case 2:
insert_begin();
break;
case 3:
insert_end();
break;
case 4:
```

```
insert_pos();
break;
case 5:
delete_begin();
break;
case 6:
delete_end();
break;
case 7:
delete_pos();
break;
case 8:
exit(0);
break;
default:
printf("\n Wrong Choice:\n");
break;
}
}
return 0;
}
void create()
{
struct node *temp,*ptr;
temp=(struct node *)malloc(sizeof(struct node));
if(temp==NULL)
{
printf("\nOut of Memory Space:\n");exit(0);
}
printf("\nEnter the data value for the node:\t");
scanf("%d",&temp->info);
temp->next=NULL;
if(start==NULL)
{
start=temp;
}
else
{
ptr=start;
while(ptr->next!=NULL)
{
ptr=ptr->next;
}
ptr->next=temp;
```

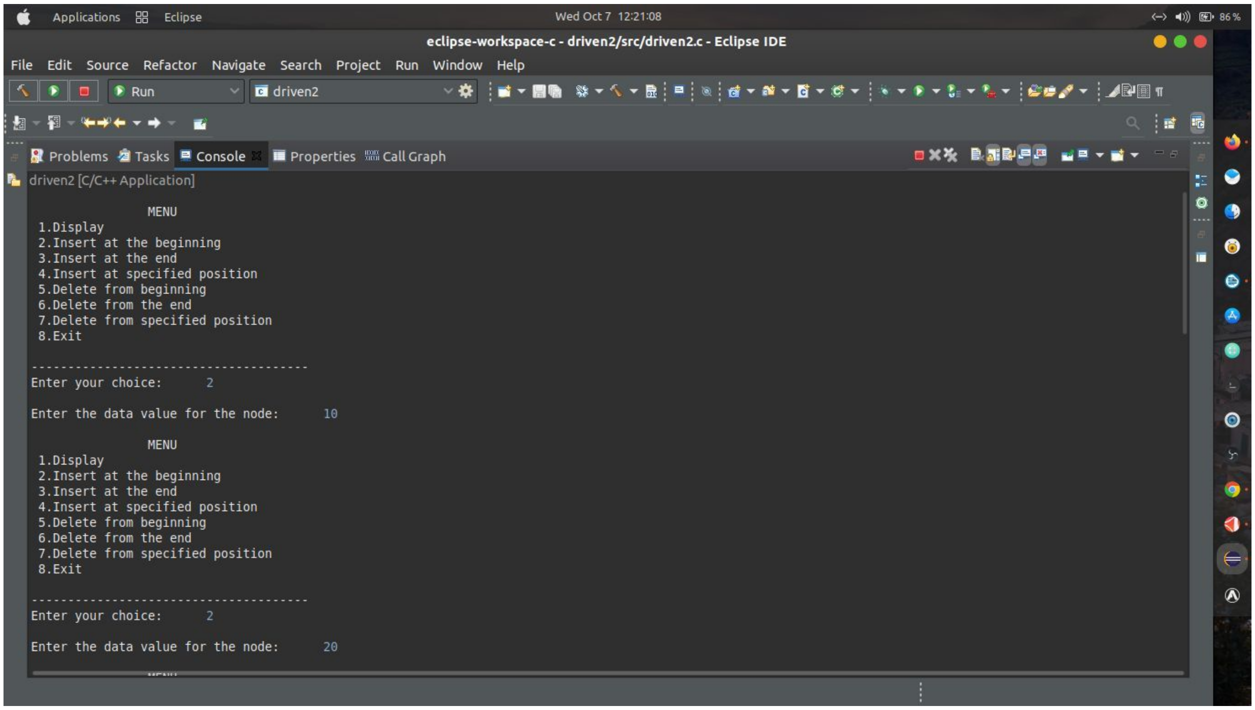
```
}
}
void display()
{
struct node *ptr;
if(start==NULL)
{
printf("\nList is empty:\n");
return;
}
else
{
ptr=start;
printf("\n The List elements are:\n");
while(ptr!=NULL)
{
printf("%d \t",ptr->info );
ptr=ptr->next ;
}
}
}
void insert_begin()
{
struct node *temp;
temp=(struct node *)malloc(sizeof(struct node));
if(temp==NULL)
{printf("\nOut of Memory Space:\n");
return;
}
printf("\nEnter the data value for the node:\t" );
scanf("%d",&temp->info);
temp->next =NULL;
if(start==NULL)
{
start=temp;
}
else
{
temp->next=start;
start=temp;
}
}
void insert_end()
{
```

```
struct node *temp,*ptr;
temp=(struct node *)malloc(sizeof(struct node));
if(temp==NULL)
{
printf("\nOut of Memory Space:\n");
return;
}
printf("\nEnter the data value for the node:\t" );
scanf("%d",&temp->info );
temp->next =NULL;
if(start==NULL)
{
start=temp;
}
else
{
ptr=start;
while(ptr->next !=NULL)
{
ptr=ptr->next ;
}
ptr->next =temp;
}
}
void insert_pos()
{struct node *ptr,*temp;
int i,pos;
temp=(struct node *)malloc(sizeof(struct node));
if(temp==NULL)
{
printf("\nOut of Memory Space:\n");
return;
}
printf("\nEnter the position for the new node to be inserted:\t");
scanf("%d",&pos);
printf("\nEnter the data value of the node:\t");
scanf("%d",&temp->info) ;
temp->next=NULL;
if(pos==0)
{
temp->next=start;
start=temp;
}
else
```

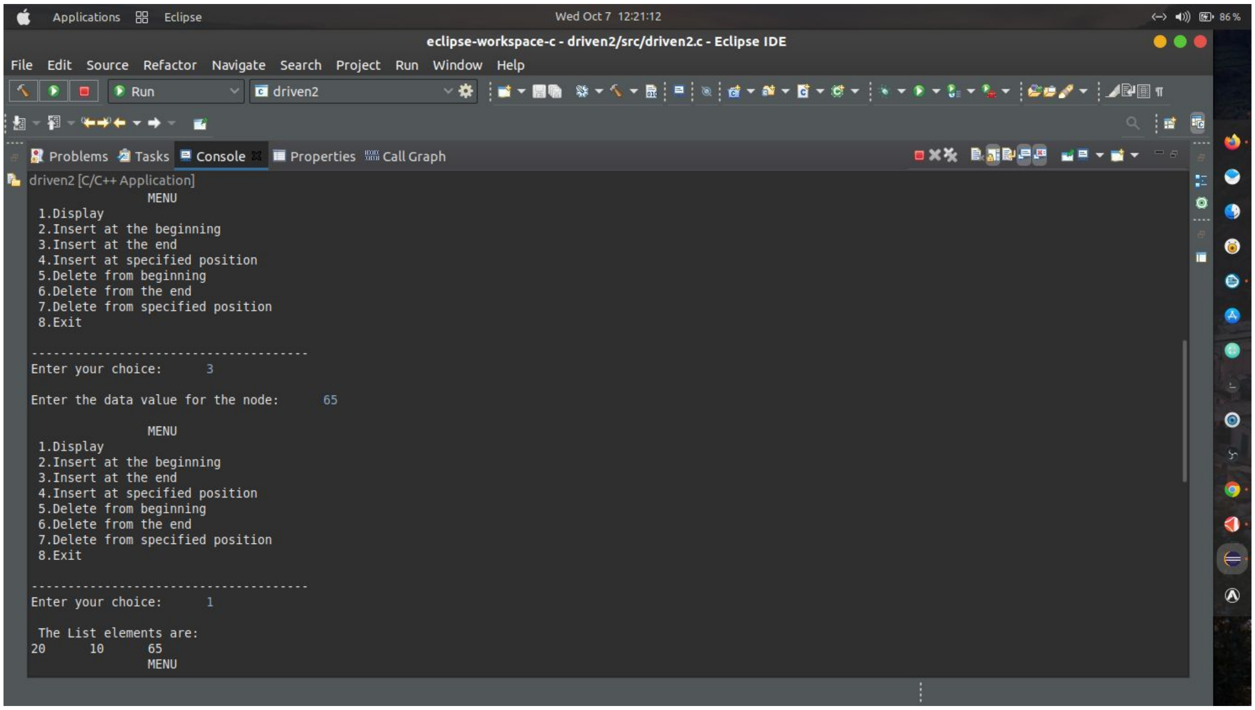
```
{
for(i=0,ptr=start;i<pos-1;i++) { ptr=ptr->next;
if(ptr==NULL)
{
printf("\n Position not found:[Handle with care]n");
return;
}
}
temp->next =ptr->next ;
ptr->next=temp;
}
}
void delete_begin()
{
struct node *ptr;
if(ptr==NULL)
{
printf("\n List is Empty :");
return;
}
else
{
ptr=start;
start=start->next ;printf("\n The deleted element is :%d \t",ptr->info);
free(ptr);
}
}
void delete_end()
{
struct node *temp,*ptr;
if(start==NULL)
{
printf("\n List is Empty:");
exit(0);
}
else if(start->next ==NULL)
{
ptr=start;
start=NULL;
printf("\n The deleted element is:%d \t",ptr->info);
free(ptr);
}
else
{
```

```
ptr=start;
while(ptr->next!=NULL)
{
temp=ptr;
ptr=ptr->next;
}
temp->next=NULL;
printf("\n The deleted element is:%d \t",ptr->info);
free(ptr);
}
}
void delete_pos()
{
int i,pos;
struct node *temp,*ptr;
if(start==NULL)
{
printf("\n The List is Empty:\n");
exit(0);
}
else
{
printf("\n Enter the position of the node to be deleted:\t");scanf("%d",&pos);
if(pos==0)
{
ptr=start;
start=start->next ;
printf("\n The deleted element is:%d \t",ptr->info );
free(ptr);
}
else
{
ptr=start;
for(i=0;i<pos;i++) { temp=ptr; ptr=ptr->next ;
if(ptr==NULL)
{
printf("\n Position not Found:\n");
return;
}
}
temp->next =ptr->next ;
printf("\n The deleted element is:%d \t",ptr->info );
free(ptr);
}
```

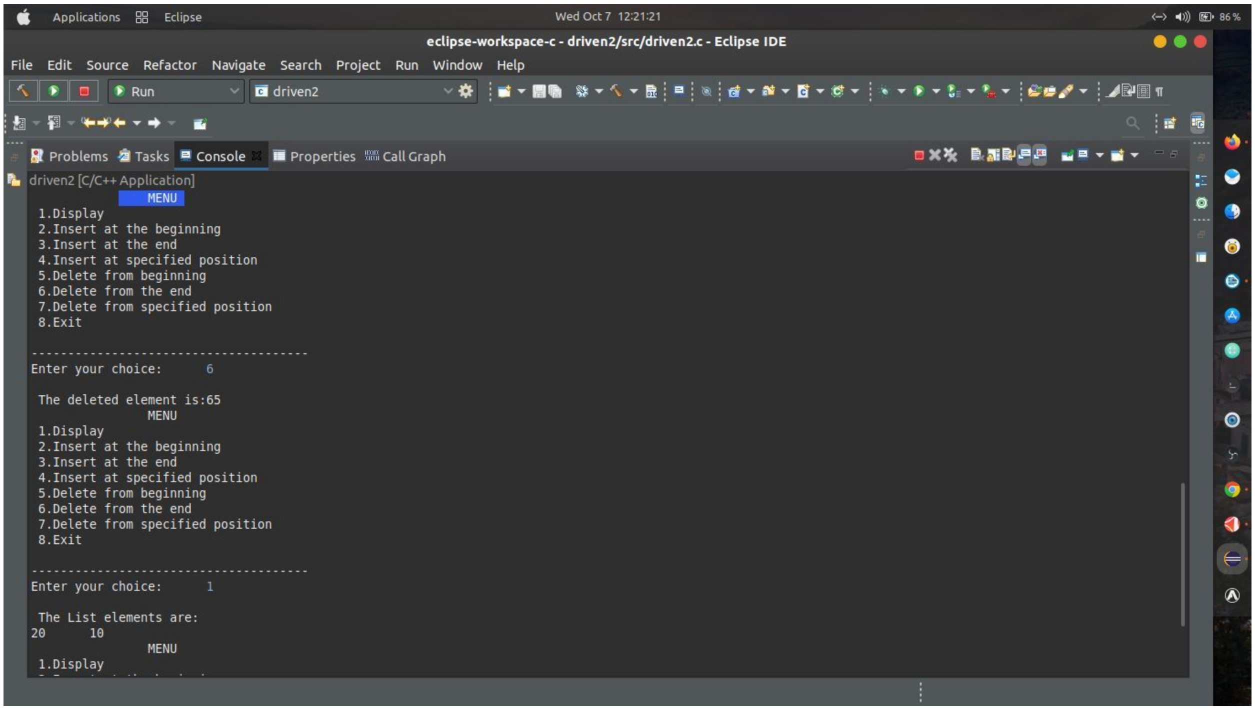
```
}
}
```


Applications
Eclipse
Wed Oct 7 12:21:08
86 %
eclipse-workspace-c - driven2/src/driven2.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Run
driven2
Problems
Tasks
Console
Properties
Call Graph

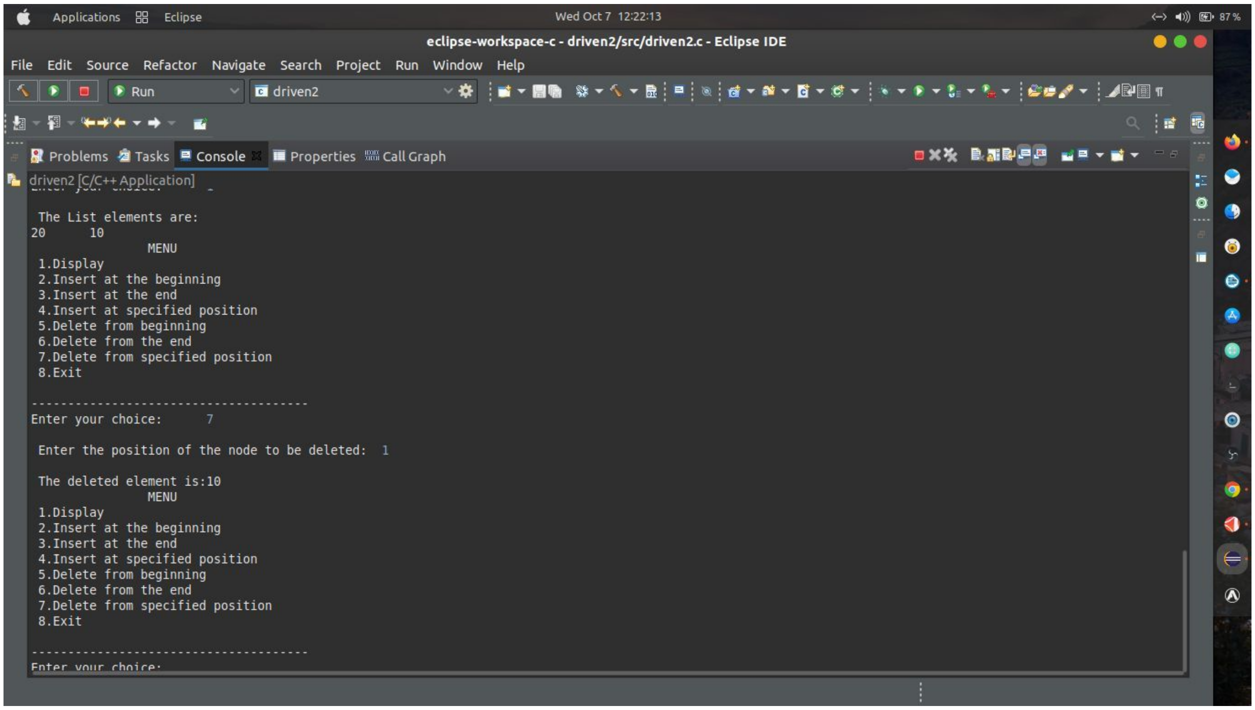
```
driven2 [C/C++ Application]

                MENU
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit

--------------------------------------
Enter your choice:      2

Enter the data value for the node:      10

                MENU
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit

--------------------------------------
Enter your choice:      2

Enter the data value for the node:      20
```


Applications
Eclipse
Wed Oct 7 12:21:12
86 %
eclipse-workspace-c - driven2/src/driven2.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Run
driven2
Problems
Tasks
Console
Properties
Call Graph

```
driven2 [C/C++ Application]
                MENU
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit

--------------------------------------
Enter your choice:      3

Enter the data value for the node:      65

                MENU
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit

--------------------------------------
Enter your choice:      1

 The List elements are:
20      10      65
                MENU
```


Applications
Eclipse
Wed Oct 7 12:21:21
86 %
eclipse-workspace-c - driven2/src/driven2.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Run
driven2
Problems
Tasks
Console
Properties
Call Graph

```
driven2 [C/C++ Application]
               MENU
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit

--------------------------------------
Enter your choice:      6

 The deleted element is:65
               MENU
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit

--------------------------------------
Enter your choice:      1

 The List elements are:
20      10
               MENU
 1.Display
```


Applications
Eclipse
Wed Oct 7 12:22:13
87 %
eclipse-workspace-c - driven2/src/driven2.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Run
driven2
Problems
Tasks
Console
Properties
Call Graph

```
driven2 [C/C++ Application]

 The List elements are:
20      10
               MENU
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit

--------------------------------------
Enter your choice:      7

 Enter the position of the node to be deleted:  1

 The deleted element is:10
               MENU
 1.Display
 2.Insert at the beginning
 3.Insert at the end
 4.Insert at specified position
 5.Delete from beginning
 6.Delete from the end
 7.Delete from specified position
 8.Exit

--------------------------------------
Enter your choice:
```

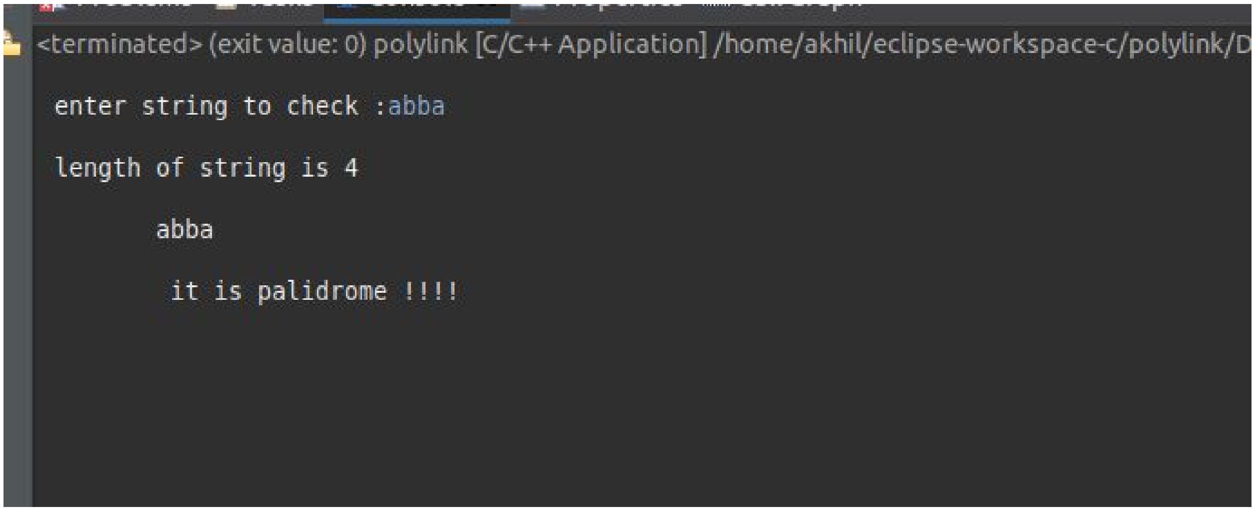
9) Create a Doubly Linked List from a string taking each character from the string. Check if the given string is palindrome in an efficient method ?

Ans :

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

```
#include<string.h>
 int create(char);
 int check(int);
 struct node
 {
      char c;
      struct node *fro;
      struct node *back;
 }*head=NULL,*tell=NULL;
 typedef struct node node;
 main()
 {
      char arr[20];
      int m,i;
      printf("\n enter string to check :");
      scanf("%s",&arr);
      m=strlen(arr);
      printf("\n length of string is %d",m);
      for(i=0;i<m;i++)
      {
           create(arr[i]);
      }
      display();
      check(m);
 }
 create(char ch)
 {
      node *temp,*r;
      temp=head;
      r=(node*)malloc(sizeof(node));
      r->c=ch;
      if(head == NULL)
      {
           head=r;
           tell=r;
           head->fro=NULL;
           head->back=NULL;
      }
      else
      {
           tell->fro=r;
           r->back=tell;
           tell=tell->fro;
           tell->fro=NULL;
```

```
    }
}
display()
{
    node *temp;
    temp=head;
    printf("\n\n\t");
    while(temp != NULL)
    {
        printf("%c",temp->c);
        temp=temp->fro;
    }
}
check(int m)
{
    node *tf,*tt;
    tf=head;
    tt=tell;
    while(m/2 > 0)
    {
        if(tf->c == tt->c)
        {
            tf=tf->fro;
            tt=tt->back;
        }
        else
        {
        printf("\n\n\t not a palidrome :???");
        exit(0);
      }
      m--;
    }
    printf("\n\n\t it is palidrome !!!!");
}
```



```
<terminated> (exit value: 0) polylink [C/C++ Application] /home/akhil/eclipse-workspace-c/polylink/D

 enter string to check :abba

 length of string is 4

        abba

         it is palidrome !!!!
```