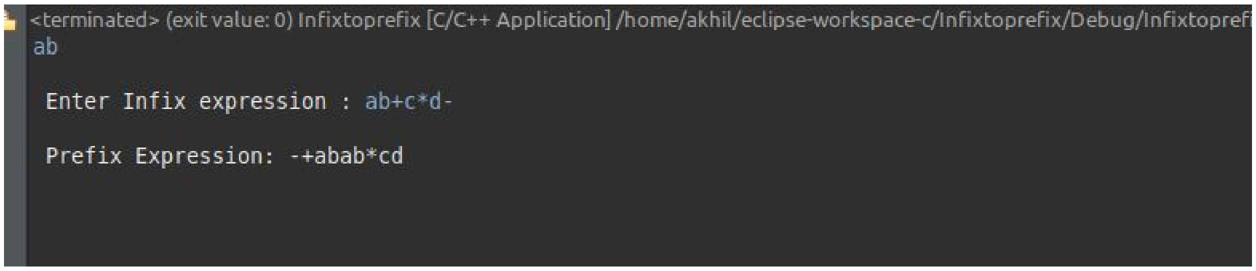
(1)--Write a program to convert an infix expression to a prefix expression using stacks. ?

Ans --}

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>
#define SIZE 100
char stack[SIZE];
int top = -1;
void push(char c);
char pop();
int isoperator(char symbol);
int precedence(char symbol);
void InfixToPrefix(char infix_exp[], char prefix_exp[]);
void main()
{
char infix[SIZE], prefix[SIZE];
printf("\n\n Enter Infix expression : ");
gets(infix);
InfixToPrefix(infix,prefix);
printf("\n Prefix Expression: ");
puts(prefix);
}

void InfixToPrefix(char infix_exp[], char prefix_exp[])
{
int i, j, k, pos, len;
char item, x, rev[SIZE];
pos=0;
len=strlen(infix_exp);
for(k=len-1;k>=0;k--)
{
rev[pos]=infix_exp[k];
pos++;
}
rev[pos]='\0';
strcpy(infix_exp,rev);
```

```
for(i=0; infix_exp[i]!='\0'; i++)
{
if(infix_exp[i] == ')')
infix_exp[i] = '(';
else if(infix_exp[i] == '(')
infix_exp[i] = ')';
}
push('(');
strcat(infix_exp,")");
i=0;
j=0;
item=infix_exp[i];
while(item != '\0')
{
if(item == '(')
{
push(item);
}
else if( isdigit(item) || isalpha(item))
{
prefix_exp[j] = item;
j++;
}
else if(isoperator(item) == 1)
{
x=pop();
while(isoperator(x) == 1 && precedence(x)>= precedence(item))
{
prefix_exp[j] = x;
j++;
x = pop();
}

push(x);
push(item);
}
else if(item == ')')
{
x = pop();
while(x != '(')
{
prefix_exp[j] = x;
j++;
x = pop();
}
}
else
{
printf("\nInvalid infix Expression.\n");
break;
}
i++;
item = infix_exp[i];
}
if(top > 0)
printf("\n Invalid infix Expression.");
prefix_exp[j] = '\0';
pos=0;
len=strlen(prefix_exp);
for(k=len-1;k>=0;k--)
{
rev[pos]=prefix_exp[k];
pos++;
}
rev[pos]='\0';
strcpy(prefix_exp,rev);
}
void push(char c)
{
if(top >= SIZE-1)
printf("\n Stack Overflow.");
else
{
top++;
stack[top] = c;
}
}
char pop()
{
char c;
c='\0';
if(top < 0)
printf("\n Stack Underflow.");
else
{
c = stack[top];
top--;
}
```

```
return c;
}

int isoperator(char symbol)
{
if(symbol == '^' || symbol == '*' || symbol == '/' || symbol == '+' || symbol == '-')
return 1;
else
return 0;
}
int precedence(char symbol)
{
if(symbol == '^')
return(5);
else if(symbol == '/')
return(4);
else if(symbol == '*')
return(3);
else if(symbol == '+')
return(2);
else if(symbol == '-')
return(1);
else
return(0);
}
```



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

 Enter Infix expression : ab+c*d-

 Prefix Expression: -+abab*cd
```

(2)--Implementation of Stack Using Linked List ?

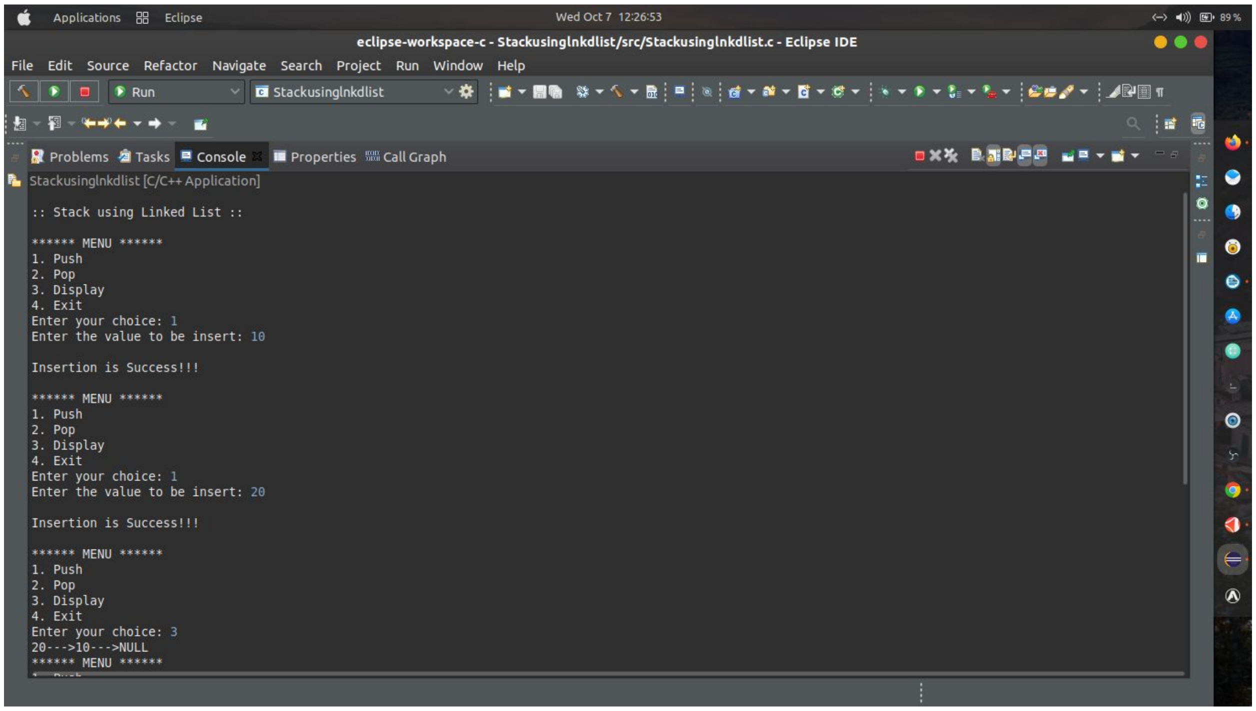
Ans --}

```
#include<stdio.h>
#include<curses.h>
struct Node
```

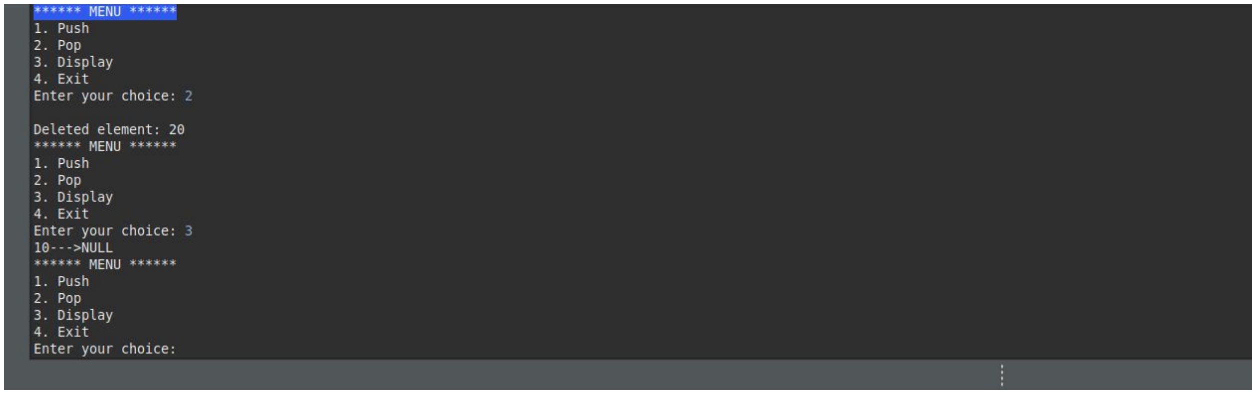
```
{
   int data;
   struct Node *next;
}*top = NULL;
void push(int);
void pop();
void display();
void main()
{
   int choice, value;
   printf("\n:: Stack using Linked List ::\n");
   while(1){
      printf("\n****** MENU ******\n");
      printf("1. Push\n2. Pop\n3. Display\n4. Exit\n");
      printf("Enter your choice: ");
      scanf("%d",&choice);
      switch(choice){
             case 1: printf("Enter the value to be insert: ");
                         scanf("%d", &value);
                         push(value);
                         break;
             case 2: pop(); break;
             case 3: display(); break;
             case 4: exit(0);
             default: printf("\nWrong selection!!! Please try again!!!\n");
      }
   }
}
void push(int value)
{
   struct Node *newNode;
   newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = value;
   if(top == NULL)
      newNode->next = NULL;
   else
      newNode->next = top;
   top = newNode;
   printf("\nInsertion is Success!!!\n");
}
void pop()
{
   if(top == NULL)
      printf("\nStack is Empty!!!\n");
```

```
  else{
    struct Node *temp = top;
    printf("\nDeleted element: %d", temp->data);
    top = temp->next;
    free(temp);
  }
}
void display()
{
  if(top == NULL)
    printf("\nStack is Empty!!!\n");
  else{
    struct Node *temp = top;
    while(temp->next != NULL){
         printf("%d--->",temp->data);
         temp = temp -> next;
    }
    printf("%d--->NULL",temp->data);
  }
}
```

OUTPUT--


Applications
Eclipse
Wed Oct 7 12:26:53
89 %
eclipse-workspace-c - Stackusinglnkdlist/src/Stackusinglnkdlist.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Run
Stackusinglnkdlist
Problems Tasks Console Properties Call Graph

```
Stackusinglnkdlist [C/C++ Application]

:: Stack using Linked List ::

****** MENU ******
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to be insert: 10

Insertion is Success!!!

****** MENU ******
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to be insert: 20

Insertion is Success!!!

****** MENU ******
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
20--->10--->NULL
****** MENU ******
```


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

Deleted element: 20
****** MENU ******
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
10--->NULL
****** MENU ******
1. Push
2. Pop
3. Display
4. Exit
Enter your choice:
```

(3)--Implementation of Queue Using Linked List ?

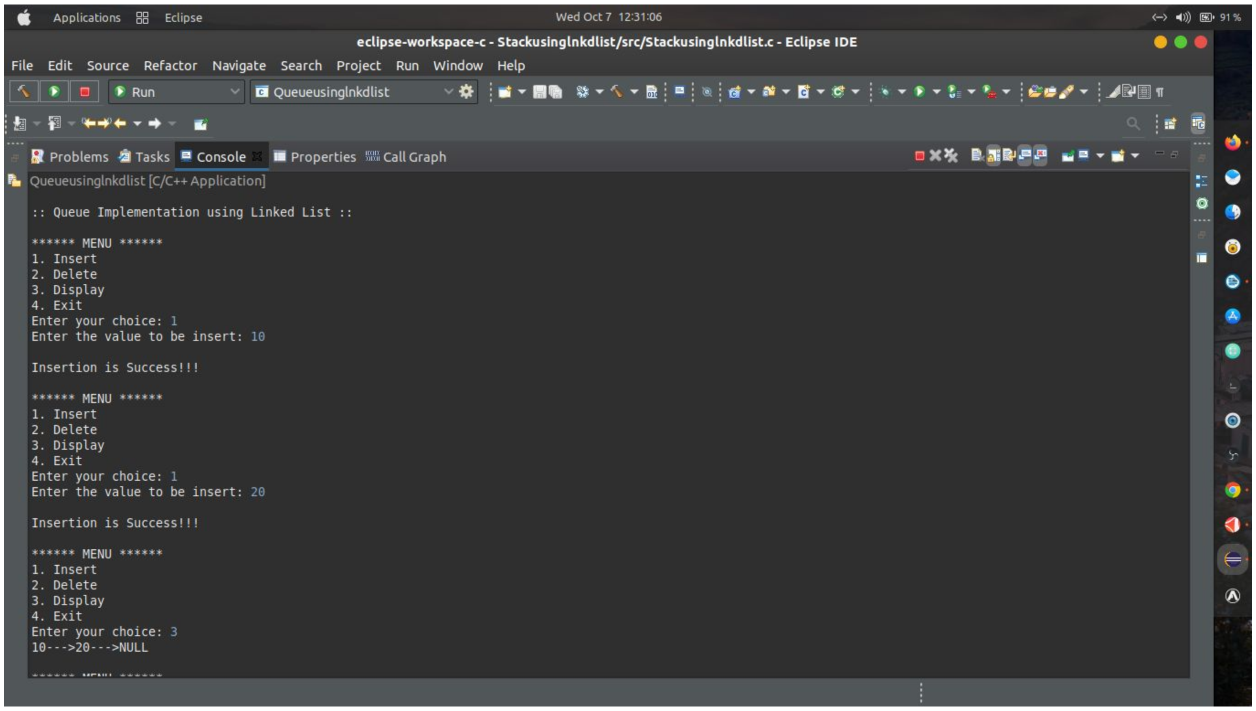
Ans--}

```
#include<stdio.h>
#include<curses.h>
struct Node
{
  int data;
  struct Node *next;
}*front = NULL,*rear = NULL;
void insert(int);
void delete();
void display();
void main()
{
  int choice, value;
  printf("\n:: Queue Implementation using Linked List ::\n");
  while(1){
    printf("\n****** MENU ******\n");
    printf("1. Insert\n2. Delete\n3. Display\n4. Exit\n");
    printf("Enter your choice: ");
    scanf("%d",&choice);
    switch(choice){
         case 1: printf("Enter the value to be insert: ");
                 scanf("%d", &value);
                 insert(value);
                 break;
         case 2: delete(); break;
         case 3: display(); break;
         case 4: exit(0);
         default: printf("\nWrong selection!!! Please try again!!!\n");
```

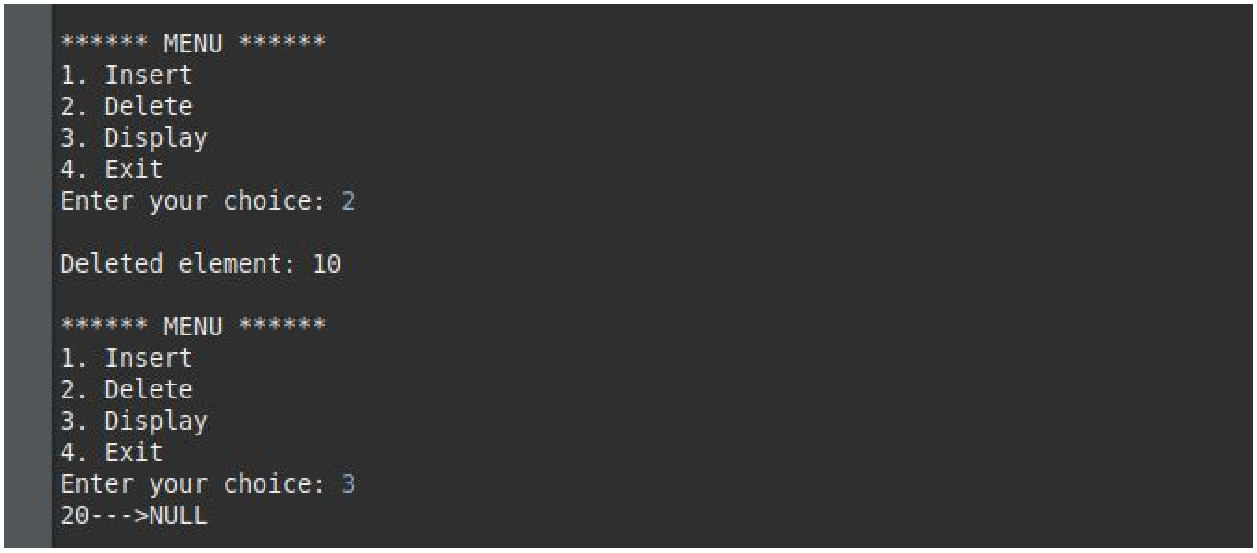
```
      }
   }
}
void insert(int value)
{
   struct Node *newNode;
   newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = value;
   newNode -> next = NULL;
   if(front == NULL)
      front = rear = newNode;
   else{
      rear -> next = newNode;
      rear = newNode;
   }
   printf("\nInsertion is Success!!!\n");
}
void delete()
{
   if(front == NULL)
      printf("\nQueue is Empty!!!\n");
   else{
      struct Node *temp = front;
      front = front -> next;
      printf("\nDeleted element: %d\n", temp->data);
      free(temp);
   }
}
void display()
{
   if(front == NULL)
      printf("\nQueue is Empty!!!\n");
   else{
      struct Node *temp = front;
      while(temp->next != NULL){
             printf("%d--->",temp->data);
             temp = temp -> next;
      }
      printf("%d--->NULL\n",temp->data);
   }
}

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

```
struct Node
{
  int data;
  struct Node *next;
}*front = NULL,*rear = NULL;
void insert(int);
void delete();
void display();
void main()
{
  int choice, value;
  printf("\n:: Queue Implementation using Linked List ::\n");
  while(1){
    printf("\n****** MENU ******\n");
    printf("1. Insert\n2. Delete\n3. Display\n4. Exit\n");
    printf("Enter your choice: ");
    scanf("%d",&choice);
    switch(choice){
         case 1: printf("Enter the value to be insert: ");
                 scanf("%d", &value);
                 insert(value);
                 break;
         case 2: delete(); break;
         case 3: display(); break;
         case 4: exit(0);
         default: printf("\nWrong selection!!! Please try again!!!\n");
    }
  }
}
void insert(int value)
{
  struct Node *newNode;
  newNode = (struct Node*)malloc(sizeof(struct Node));
  newNode->data = value;
  newNode -> next = NULL;
  if(front == NULL)
    front = rear = newNode;
  else{
    rear -> next = newNode;
    rear = newNode;
  }
  printf("\nInsertion is Success!!!\n");
}
void delete()
```

```
{
   if(front == NULL)
      printf("\nQueue is Empty!!!\n");
   else{
      struct Node *temp = front;
      front = front -> next;
      printf("\nDeleted element: %d\n", temp->data);
      free(temp);
   }
}
void display()
{
   if(front == NULL)
      printf("\nQueue is Empty!!!\n");
   else{
      struct Node *temp = front;
      while(temp->next != NULL){
            printf("%d--->",temp->data);
            temp = temp -> next;
      }
      printf("%d--->NULL\n",temp->data);
   }
```


Applications
Eclipse
Wed Oct 7 12:31:06
eclipse-workspace-c - Stackusinglnkdlist/src/Stackusinglnkdlist.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Run
Queueusinglnkdlist
Problems Tasks Console Properties Call Graph

```
Queueusinglnkdlist [C/C++ Application]

:: Queue Implementation using Linked List ::

****** MENU ******
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to be insert: 10

Insertion is Success!!!

****** MENU ******
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to be insert: 20

Insertion is Success!!!

****** MENU ******
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
10--->20--->NULL
```



```
****** MENU ******
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2

Deleted element: 10

****** MENU ******
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
20--->NULL
```