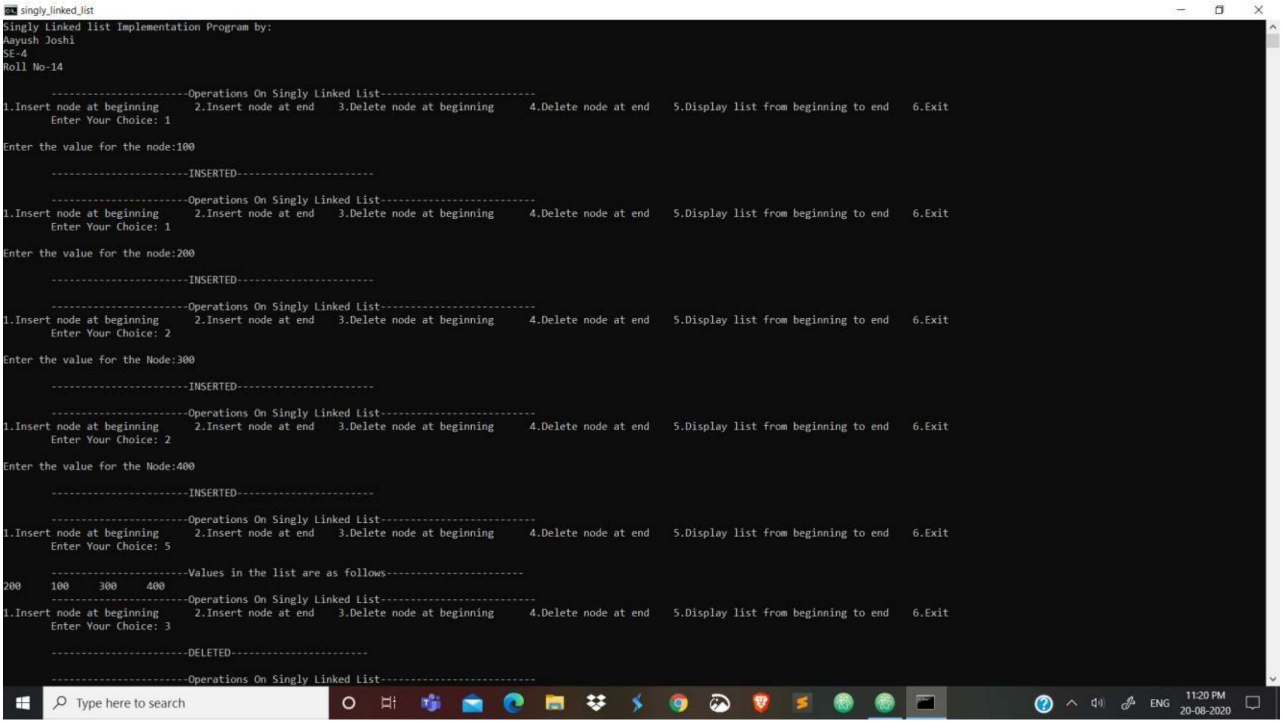
EXPERIMENT 2
Singly Linked List

Program:

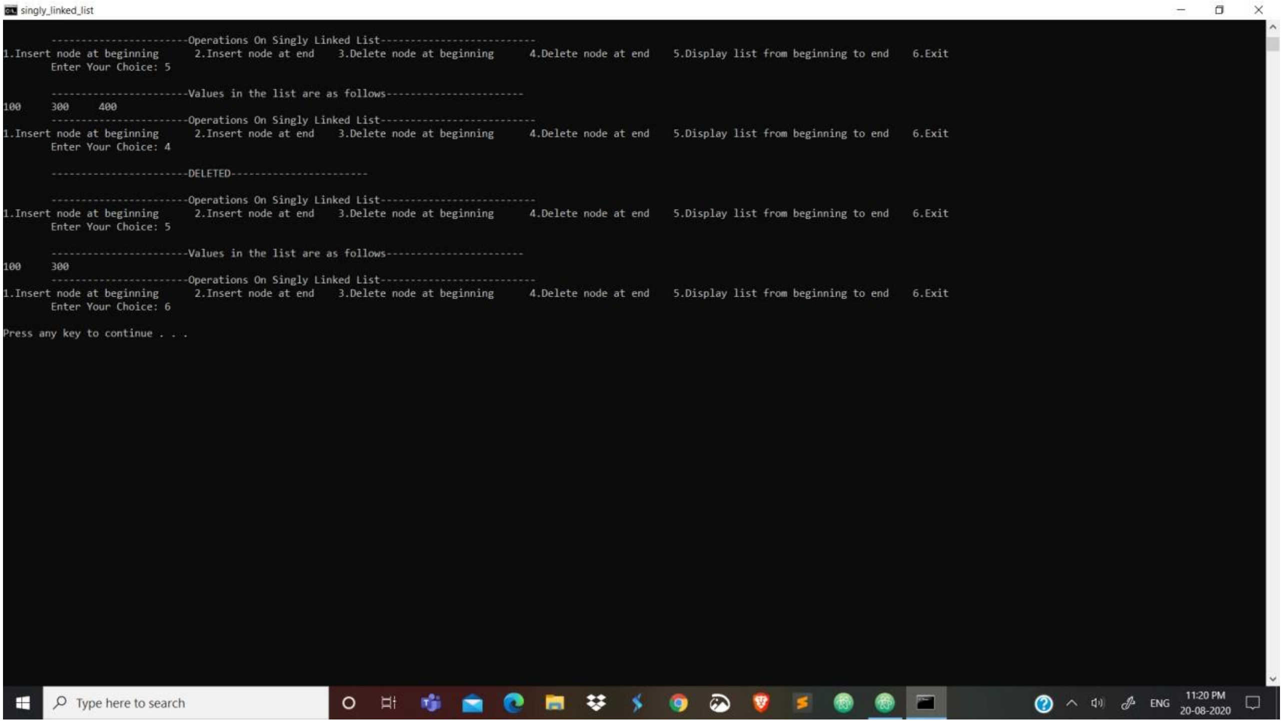
```
#include<stdlib.h>
#include<stdio.h>
#include<malloc.h>
//Defining isempty for checking the List
#define ISEMPTY printf("\nEMPTY LIST:");
//Declaring node
struct node
{   int value;
struct
node*next;
};

//Declaring the required functions for
performing operations struct node*
create_node(int); void
insert_node_begin(); void
insert_node_end(); void
delete_node_begin(); void
delete_node_end(); void display();
```

```
struct node
*newnode,*ptr,*prev,*temp;
struct node *start = NULL, *end
= NULL;


int main()
{
int
ch;
  printf("Singly Linked list Implementation Program by:\nAayush
Joshi\nSE-4\nRoll No-14\n");   while(2)
  {
    printf("\n\t-----------------------Operations On Singly Linked List------------
--------------");
    printf("\n1.Insert node at beginning\t2.Insert node at end\t3.Delete node at
beginning\t4.Delete node at end\t5.Display list from beginning to
end\t6.Exit");     printf("\n\tEnter Your Choice: ");     scanf("%d",&ch);

    switch(ch)
    {
      case
1:insert_node_begin();brea
k;       case
2:insert_node_end();break;
case
3:delete_node_begin();bre
ak;       case
4:delete_node_end();break
;       case
5:display();break;       case
6:exit(0);       default:
              printf("\n\tEnter correct choice (1-6)");break;
    }
  }
  return 0;
}

//Creating Node struct
node* create_node(int
val)
{
    newnode = (struct node
*)malloc(sizeof(struct node));     if
(newnode == NULL)
    {
        printf("\nMemory was not allocated");
        return 0;
  }
else
{
    newnode->value =
val;     newnode->next
= NULL;     return
newnode;
  }
}

//Inserting node at
beginning void
insert_node_begin()
{
int
val;

  printf("\nEnter the value for
the node:");   scanf("%d",
&val);   newnode =
create_node(val);   if (start
== end && start == NULL)
  {
        start = end =
newnode;         start-
>next = NULL;
end->next = NULL;
    }
else
    {
        temp =
start;         start
= newnode;
start->next =
temp;
    }
    printf("\n\t-----------------------INSERTED-----------------------\n");
}

//Insering   node   at
end                   void
insert_node_end()
{     int val;


    printf("\nEnter the value for
the Node:");     scanf("%d",
&val);     newnode =
create_node(val);     if (start
== end && end == NULL)
    {
        start = end =
newnode;         start-
>next = NULL;
end->next = NULL;
    }
else
    {
        end->next =
newnode;         end =
newnode;         end-
>next = NULL;
    }
    printf("\n\t----------------------INSERTED----------------------\n");
}
//Deleting Node at
beginning void
delete_node_begin()
{
  struct node*temp = start;
if(temp==NULL)
  {
    printf("\nLinked list is empty:");
  }
  else
  {
    start = start->next;
free(temp);
printf("\n\t---------------
--------DELETED------
----------------\n");
  }

}

//Deleting Node at
end void
delete_node_end()
{
  struct node *temp = start;
if(temp==NULL)
  {
    printf("\nLinked list is empty:");
  }
  else
  {
    if(temp->next == NULL)
    {
      start = NULL;
free(temp);
    }
else
    {
      while(temp->next->next != NULL)
      {
        temp = temp->next;
      }
      free(temp->next);       temp->next =
NULL;       printf("\n\t----------------------
DELETED----------------------\n");
    }
  }
}

//Displaying the list from
beginning to end void display()
{     if (start
== NULL)
    {
        ISEMPTY;         printf(":No
nodes in the list to display\n");
    }
else
    {
        printf("\n\t----------------------Values in the list are as
follows----------------------\n");         for (ptr = start;ptr !=
NULL;ptr = ptr->next)
        {
            printf("%d\t", ptr->value);
        }
    }
}
```

Output:



```
Singly Linked list Implementation Program by:
Aayush Joshi
SE-4
Roll No-14

        ----------------------Operations On Singly Linked List-------------------------
1.Insert node at beginning      2.Insert node at end    3.Delete node at beginning      4.Delete node at end    5.Display list from beginning to end    6.Exit
        Enter Your Choice: 1

Enter the value for the node:100

        ----------------------INSERTED----------------------

        ----------------------Operations On Singly Linked List-------------------------
1.Insert node at beginning      2.Insert node at end    3.Delete node at beginning      4.Delete node at end    5.Display list from beginning to end    6.Exit
        Enter Your Choice: 1

Enter the value for the node:200

        ----------------------INSERTED----------------------

        ----------------------Operations On Singly Linked List-------------------------
1.Insert node at beginning      2.Insert node at end    3.Delete node at beginning      4.Delete node at end    5.Display list from beginning to end    6.Exit
        Enter Your Choice: 2

Enter the value for the Node:300

        ----------------------INSERTED----------------------

        ----------------------Operations On Singly Linked List-------------------------
1.Insert node at beginning      2.Insert node at end    3.Delete node at beginning      4.Delete node at end    5.Display list from beginning to end    6.Exit
        Enter Your Choice: 2

Enter the value for the Node:400

        ----------------------INSERTED----------------------

        ----------------------Operations On Singly Linked List-------------------------
1.Insert node at beginning      2.Insert node at end    3.Delete node at beginning      4.Delete node at end    5.Display list from beginning to end    6.Exit
        Enter Your Choice: 5

        ----------------------Values in the list are as follows----------------------
200     100     300     400
        ----------------------Operations On Singly Linked List-------------------------
1.Insert node at beginning      2.Insert node at end    3.Delete node at beginning      4.Delete node at end    5.Display list from beginning to end    6.Exit
        Enter Your Choice: 3

        ----------------------DELETED----------------------

        ----------------------Operations On Singly Linked List-------------------------
```



```
        ----------------------Operations On Singly Linked List-------------------------
1.Insert node at beginning      2.Insert node at end    3.Delete node at beginning      4.Delete node at end    5.Display list from beginning to end    6.Exit
        Enter Your Choice: 5

        ----------------------Values in the list are as follows----------------------
100     300     400
        ----------------------Operations On Singly Linked List-------------------------
1.Insert node at beginning      2.Insert node at end    3.Delete node at beginning      4.Delete node at end    5.Display list from beginning to end    6.Exit
        Enter Your Choice: 4

        ----------------------DELETED----------------------

        ----------------------Operations On Singly Linked List-------------------------
1.Insert node at beginning      2.Insert node at end    3.Delete node at beginning      4.Delete node at end    5.Display list from beginning to end    6.Exit
        Enter Your Choice: 5

        ----------------------Values in the list are as follows----------------------
100     300
        ----------------------Operations On Singly Linked List-------------------------
1.Insert node at beginning      2.Insert node at end    3.Delete node at beginning      4.Delete node at end    5.Display list from beginning to end    6.Exit
        Enter Your Choice: 6

Press any key to continue . . .
```