1BM19CS028 LAB-9.

Doubly linked list

① Inserting

```
void insertNode()
  {
    if (head == NULL)
      { head = newnode;
      newnode -> next = newnode -> prev = NULL;
       return;
    }
      Node * temp = head;
      for (temp; (temp -> next) != NULL; temp = temp -> next);
      temp->next = newnode;
        newnode -> prev = temp;
          newnode -> next = NULL;
    }
```

② Inserting to right of a node.

```
{ int ele;
  char choice;
  Printf ("\n Enter the ele whose right you want to insert : ");
  Scanf ("%d", ele);
Node *temp = head;

  { if (temp->next != NULL)
  { tempnew -> next = temp -> next;
      tempnew -> prev = temp;

   return;
  }.
```

```
else
{
    tempnew -> next = NULL;
    tempnew -> prev = temp;
    temp -> next = tempnew;
    return;
}
}
}
Print f(" In Element Not found : Press y to try again :");
Scanf (" %c", choice);
if (choice == 'y' || choice == 'Y')
    insertNodetoRight (tempnew);
    else
    free (temnew).
}.
```

(5) void displaylist ().

```
{
  if (head == NULL).
  {
    PrintF("\n Empty list !\n");
    return
  }
    Node * temp=head;
  Printf ("\n the list contains .");
    for ( temp; temp = NULL ;  temp= temp->next )
  Printf ("%d", temp->data);
    return;
}.
```

(6) deletion in Fast node

```
temp -> prev -> next = NULL;
free (temp);
if first Node:
        temp -> next -> prev = NULL;
        head = head -> next;
if in-between;
        temp -> prev -> next = temp -> next;
        temp -> next -> prev = temp -> prev;
        free (temp).
```