Bharatiya Vidya Bhavan's

Sardar Patel Institute of Technology

(Autonomous Institute Affiliated to University of Mumbai)

Name:YASH LALIT SHARMA

UID:2023300220

EXP NO. 4

DIV: D

BATCH: A or D1

AIM: Implement a given problem statement using Doublely Linked List.

PROBLEM STATEMENT:

Perform following Operations on doubly linked list

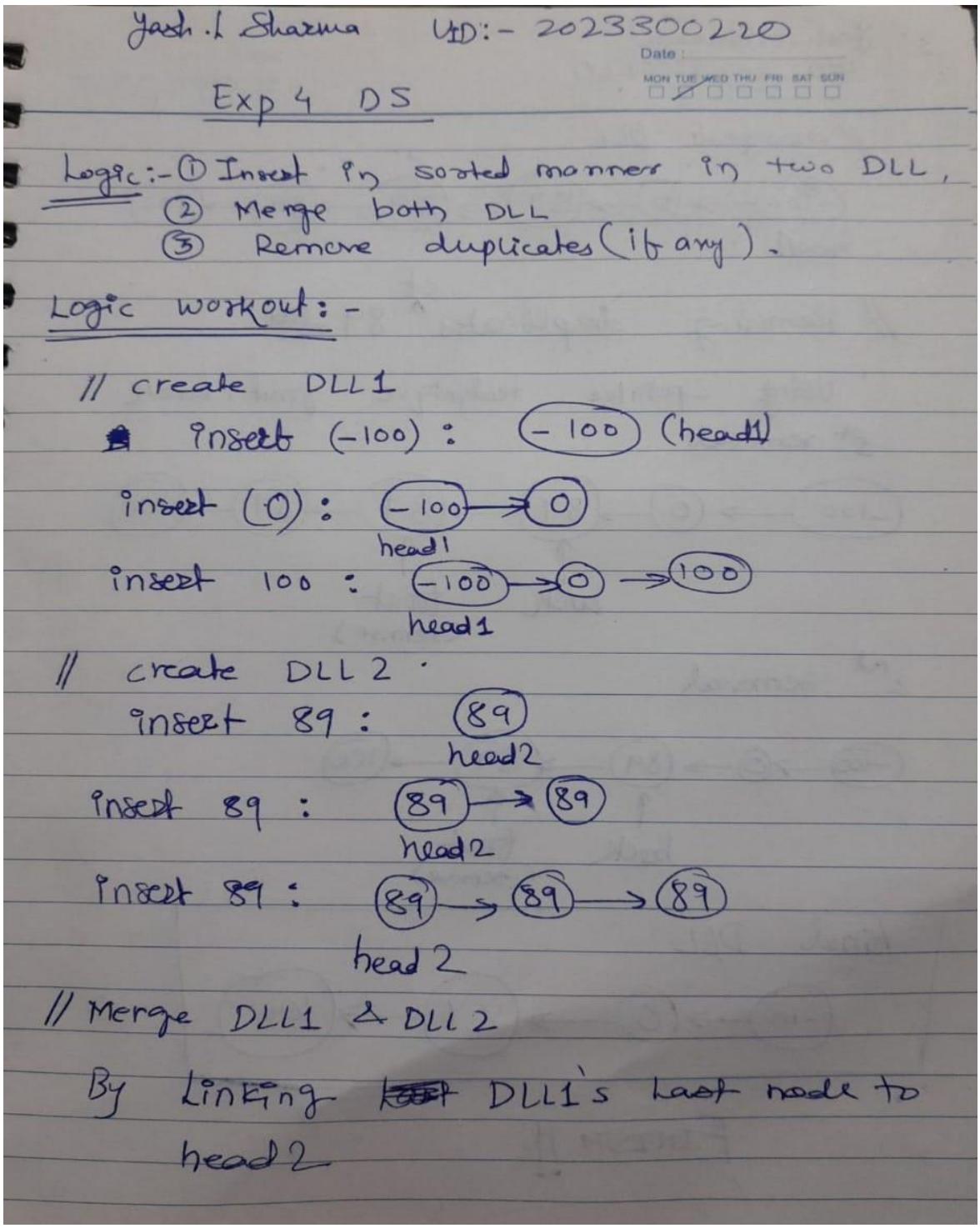
1-sort the list. Take input data in random order and insert it at appropriate place in the list

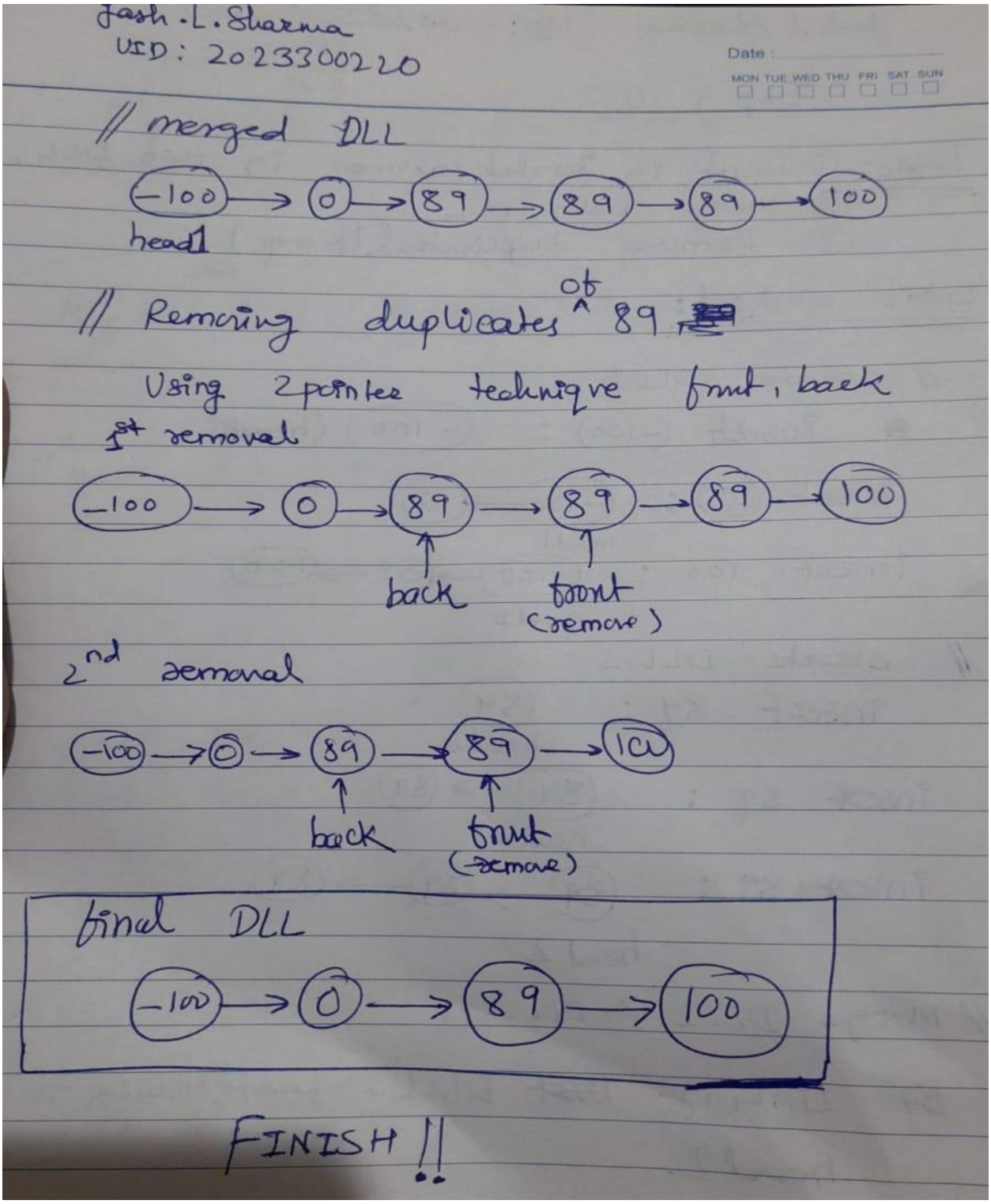
2-Remove the duplicates from the sorted doubly linked list

3-merge two doubly linked lists in sorted manner

Hand Written Solution:

Page 1




Yash. L. Sharma
UID: 2023300220
Date :
MON TUE WED THU FRI SAT SUN
// merged DLL
-100 → 0 → 89 → 89 → 89 → 100
head
// Removing duplicates of 89
Using 2 pointer technique front, back
1st removal
-100 → 0 → 89 → 89 → 89 → 100
back
front (remove)
2nd removal
-100 → 0 → 89 → 89 → 100
back
front (remove)
final DLL
-100 → 0 → 89 → 100
FINISH !!

Program

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

struct Node{
 int data;
 struct Node* next;
 struct Node* prev;

};

struct Node* createnode(int x)
{
  struct Node* newnode = (struct Node*)malloc(sizeof(struct Node));
   if (newnode == NULL) {
       printf("Memory allocation failed!\n");
       exit(1);
    }

   newnode->next=NULL;
   newnode->prev=NULL;

   newnode->data=x;



   return  newnode;
}
```

```
void insert(struct Node**head,int x)
{
 struct Node* newnode =createnode(x);
 if(*head == NULL)
 {
    *head= newnode;
    return;
 }
 // insert to  left of head
 if( (*head)->data  >=x)
 {
   newnode= createnode(x);
   newnode->next=*head;
   (*head)->prev=newnode;
   *head=newnode;
 }

 else{
   // x> head->data

  struct Node* curr = *head;

  while(curr->next !=NULL &&  x>=curr->next->data )
       curr=curr->next;

       if(curr->next == NULL)
       {
```

```
        newnode = createnode(x);
        curr->next=newnode;
        newnode->prev=curr;
        return;
       }

       if(curr->next->data>= x)
       {
        newnode = createnode(x);
        newnode->next=curr->next;
        curr->next->prev=newnode;
        curr->next=newnode;
        newnode->prev=curr;
        return;
       }
  }
}


void merge(struct Node** head1 ,struct Node** head2 )
{
   printf("linked list after merging:\n");
   // going  to last node of dll1
   struct Node* curr = *head2;
   while(curr!=NULL)
   {
      insert(&*head1,curr->data);
      curr=curr->next;
```

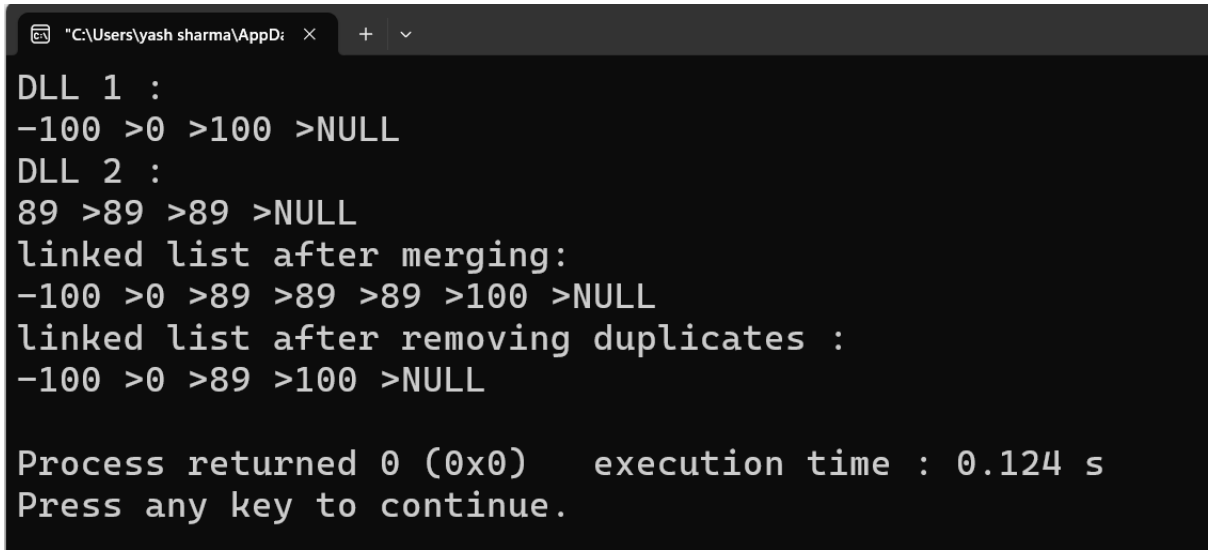
```
    }
}
void remove_duplicate(struct Node** head)
{
    printf("linked list after removing duplicates :\n");
    struct Node* front = *head;
    struct Node* rear = *head;

    while(rear !=NULL)
    {
       front=rear->next;
       while(front !=NULL)
       {

         if(front->data == rear->data)
         {
           struct Node* temp = front;
           front=front->next;
           if(temp->prev!=NULL)
           temp->prev->next=temp->next;
           if(temp->next !=NULL)
           temp->next->prev=temp->prev;
          if (rear->next == temp)
              rear->next = temp->next;
           //free(temp);
         }
         else
         front=front->next;
```

```
      }
     rear=rear->next;
   }
}

void display(struct Node** head)
{
   struct Node* curr = *head;
   //printf("Linked List : \n");
   while(curr!=NULL)
   {
      printf("%d >",curr->data);
      curr = curr->next;
   }
   printf("NULL \n");
}

int main()
{
  struct Node* head1 = NULL;
   insert(&head1,0);
   insert(&head1,-1);
   insert(&head1,5);
   insert(&head1,3);
   insert(&head1,7);
   insert(&head1,1);
   insert(&head1,9);
   printf("DLL 1 :\n");
```

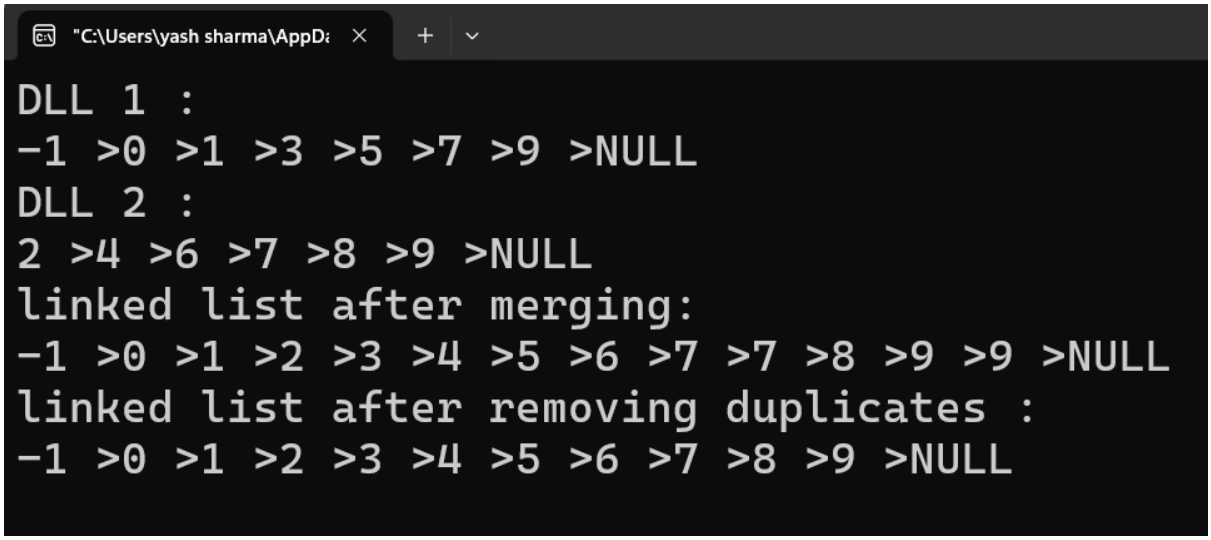
```
    display(&head1);


   struct Node* head2 = NULL;
    insert(&head2,9);
    insert(&head2,8);
    insert(&head2,7);
    insert(&head2,2);
    insert(&head2,4);
    insert(&head2,6);
    printf("DLL 2 :\n");
    display(&head2);
    merge(&head1,&head2);
    display(&head1);
    remove_duplicate(&head1);
    display(&head1);
    return 0;
}
```

Output:-

//TEST CASE 1



//TEST CASE 2



Conclusion: In this experiment we performed operations on Doubely Linkedlist like insertion in sorted manner ,merging and removing duplicates which honed our coding skill to next level .