Data Structures

Experiment no. 6

Develop code to implement operations on Doubly Linked List

Q. WAP in C to implement Doubly Linked List operations.

Code:

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

struct Node *createNode(int);

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

struct Node *node,*list=NULL,*last=NULL,*temp=NULL,*p=NULL,*a;

struct Node *createNode(int info)
{
        struct Node *node=(struct Node*)malloc(sizeof(struct Node));
        node -> prev = NULL;
        node -> data = info;
        node -> next = NULL;
        return node;
}

void addAtBeginning(int info)
{
        node = createNode(info);
        if(list == NULL)
        {
                list = node;
                last = node;
        }
        else
        {
                list -> prev = node;
                node -> next = list;
                list = node;
```

```
        }
}

void addAtEnd(int info)
{
        node = createNode(info);
        if(list == NULL)
        {
                list = node;
                last = node;
        }
        else
        {
                last -> next = node;
                node -> prev = last;
                last = node;
        }
}

void addAtPosition(int info, int pos)
{
        node = createNode(info);
        temp = list;
        p = list;
        if(pos == 1)
        {
                addAtBeginning(info);
        }
        else
        {
          int i;
          for(i=0;i<pos-2;i++)
          {
                 temp = temp -> next;
                 p = temp;
          }
          temp = temp -> next;
          p -> next = node;
          node -> prev = p;
          temp -> prev = node;
          node -> next = temp;
        }
}

void deleteAtBeginning()
{
        if(list == NULL)
        {
```

```
                printf("\nList is empty");
        }
        else
        {
                temp = list;
                list = list -> next;
                list -> prev = NULL;
                free(temp);
                temp = NULL;
        }
}

void deleteAtEnd()
{
        if(list == NULL)
        {
                printf("\nList is empty");
        }
        else
        {
                temp = last;
                last = last -> prev;
                last -> next = NULL;
                free(temp);
                temp = NULL;
        }
}

void deleteAtPosition(int pos)
{
        if(list == NULL)
        {
                printf("\nList is empty");
        }
        else
        {
                int i;
                temp=list;
                p=list;
                for(i=0;i<pos-2;i++)
                {
                        temp = temp -> next;
                        p = temp;
                }
                temp = temp -> next;
                a = temp -> next;
                p -> next = a;
                a -> prev = p;
```

```
                free(temp);
        }
}


void display()
{
        temp = list;
        if(list == NULL)
        {
                printf("\nList is empty");
        }
        else
        {
                while(temp != NULL)
                {
                        printf("%d->",temp -> data);
                        temp = temp -> next;
                }
        }
}

void main()
{
        int info,ch,pos;
        clrscr();
        do{
                printf("\n1) Add no. at beginning\n2) Add no. at end\n3) Add no. at position\n4)
Delete at beginning\n5) Delete at end\n6) Delete at position\n7) Display\n8) EXIT\n");
                printf("\tEnter an option : ");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:
                                printf("\nEnter a no. : ");
                                scanf("%d",&info);
                                addAtBeginning(info);
                                break;

                        case 2:
                                printf("\nEnter a no. : ");
                                scanf("%d",&info);
                                addAtEnd(info);
                                break;

                        case 3:
                                printf("\nEnter a no. : ");
                                scanf("%d",&info);
```

```
                                printf("\nEnter position : ");
                                scanf("%d",&pos);
                                addAtPosition(info,pos);
                                break;

                        case 4:
                                deleteAtBeginning();
                                break;

                        case 5:
                                deleteAtEnd();
                                break;

                        case 6:
                                printf("\nEnter position : ");
                                scanf("%d",&pos);
                                deleteAtPosition(pos);
                                break;

                        case 7:
                                printf("\nElements in the List : \n\t");
                                display();
                                break;

                        case 8:
                                exit();
                                break;

                        default:
                                printf("\n\tInvalid Choice!\n");
                }
        }while(ch != 8);
        getch();
}
```

Output:

```
1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 7
```

```
Elements in the List :

List is empty
1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 1

Enter a no. : 100

1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 1

Enter a no. : 200

1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 1

Enter a no. : 300

1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 2

Enter a no. : 900
```

```
1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 2

Enter a no. : 500

1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 3

Enter a no. : 1000

Enter position : 4

1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 7

Elements in the List :
        300->200->100->1000->900->500->
1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 4

1) Add no. at beginning
2) Add no. at end
3) Add no. at position
```

```
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 5

1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 7

Elements in the List :
        200->100->1000->900->
1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 6

Enter position : 3

1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
        Enter an option : 7

Elements in the List :
        200->100->900->
1) Add no. at beginning
2) Add no. at end
3) Add no. at position
4) Delete at beginning
5) Delete at end
6) Delete at position
7) Display
8) EXIT
```

```
        Enter an option : 8
```