Data Structures

Experiment no. 5

Develop code to implement operations on Singly Linked List

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

Code:

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

struct Node* createNode(int);

struct Node
{
	int data;
	struct Node* next;
};
struct Node *node,*list=NULL,*last=NULL,*temp,*p,*q;

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

	return node;
}

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

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

void addAtPosition(int info, int position)
{
  int i;
  node=createNode(info);
  temp=list;
  if(position==1)
  {
                addAtFront(info);
  }
  else
  {
                for(i=1;i<position-1;i++)
                {
                        temp = temp -> next;
                }
                node -> next = temp ->next;
                temp -> next =node;
  }
}

void deletAtBeginning()
{
        temp=list;
        if(list==NULL)
        {
                printf("\n:List is empty");
        }
        else
        {
                list=list->next;
                free(temp);
        }
```

```
}

void deletAtEnd()
{
	if(list==NULL)
	{
		printf("\n:List is empty");
	}
	else if(list->next==NULL)
	{
		 list=NULL;
	}
	else
	{
		temp=list;
		p=list;
		while(temp->next!=NULL)
		{
			p=temp;
			temp=temp->next;
		}
		p->next=NULL;
		last=p;
		free(temp);
	}
}

void deletAtPosition(int pos)
{
	temp=list;
	p=list;
	int i;
	if(list==NULL)
	{
		printf("\nList is empty");
	}
	else if(pos==1)
	{
		deletAtBeginning();
	}
	else
	{
		for(i=1;i<pos-1;i++)
		{
			p=temp;
			temp=temp->next;
		}
		p->next=temp->next;
```

```
                free(temp);
        }
}

void search(int info)
{
        if(list==NULL)
        {
                printf("\nList is empty");
        }
        else
        {
                temp=list;
                while(temp != NULL)
                {
                        if(temp -> data==info)
                        {
                                printf("\nElement found");
                                return;
                        }
                        else
                        {
                                temp = temp -> next;
                        }
                }
                printf("\nElement not found");
        }
}

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

void main()
{
```

```
        int info,ch,position;
        clrscr();
        do
        {
                printf("\n1) Add element at front\n2) Add element at end\n3) Add element at a
position\n4) Delete element at front\n5) Delete element at end\n6) Delete element at
position\n7) Display\n8) Search\n9) EXIT\n");
                printf("\n\tEnter your choice : ");
                scanf("%d",&ch);
                switch(ch)
                {
                        case 1:
                                printf("\nEnter value to be added : ");
                                scanf("%d",&info);
                                addAtFront(info);
                                break;

                        case 2:
                                printf("\nEnter value to be added : ");
                                scanf("%d",&info);
                                addAtEnd(info);
                                break;

                        case 3:
                                printf("\nEnter element to be added : ");
                                scanf("%d",&info);
                                printf("\nAdd position :");
                                scanf("%d",&position);
                                addAtPosition(info,position);
                                break;

                        case 4:
                                deletAtBeginning();
                                break;

                        case 5:
                                deletAtEnd();
                                break;

                        case 6:
                                printf("\nEnter position to delete :");
                                scanf("%d",&position);
                                deletAtPosition(position);
                                break;

                        case 7:
                                printf("\nElements in the list : \n\t");
                                display();
```

```
                        break;

                case 8:
                        printf("Enter element to be searched : ");
                        scanf("%d",&info);
                        search(info);
                        break;

                case 9:
                        exit();
                        break;

                default:
                        printf("\nInvalid Choice!");
                }
        }while(ch!=9);
        getch();
}
```

Output:

```
1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 7

Elements in the list :
        List is empty

1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 1

Enter value to be added : 100
```

```
1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 1

Enter value to be added : 200

1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 2

Enter value to be added : 300

1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 7

Elements in the list :
        200     100     300
1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
```

```
9) EXIT

        Enter your choice : 3

Enter element to be added : 1000

Add position :2

1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 7

Elements in the list :
        200     1000    100     300
1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 4

1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 7

Elements in the list :
        1000    100     300
1) Add element at front
2) Add element at end
3) Add element at a position
```

```
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 5

1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 7

Elements in the list :
        1000    100
1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 1

Enter value to be added : 600

1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 2

Enter value to be added : 85
```

```
1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 6

Enter position to delete :3

1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 7

Elements in the list :
        600     100     85
1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 8
Enter element to be searched : 2

Element not found
1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT
```

```
        Enter your choice : 8
Enter element to be searched : 100

Element found
1) Add element at front
2) Add element at end
3) Add element at a position
4) Delete element at front
5) Delete element at end
6) Delete element at position
7) Display
8) Search
9) EXIT

        Enter your choice : 9
```