

Lab - 4

1) Write a program to implement singly linked list with the following operations:

- a) Create a linked list
- b) Insertion of a node at first position, at any position and at end of the list.
- c) Display contents of the list

ans) #include <stdio.h>

#include <stdlib.h>

struct node

{

int data;

struct node * next;

};

struct node * start = NULL;

struct node * new_node, * temp;

Void createlist()

{

int i, n;

printf("Enter no. of elements : ");

scanf("%d", &n);

for(i = 0; i < n; i++)

{

new_node = (struct node *) malloc

sizeof(struct node));

printf("Enter the element : ");

scanf("%d", &new_node->data);

new_node->next = NULL;

if (start == NULL)

{

```

        temp = start > new-node;
    }
else
{
    temp->next = new-node;
    temp = new-node;
}
}

void insert_beg()
{
    new-node = (struct node *) malloc (sizeof
        (struct node));
    printf ("Enter new element : ");
    scanf ("%d", &new-node->data);
    new-node->next = start;
    start = new-node;
}

void insert_end()
{
    new-node = (struct node *) malloc (sizeof
        (struct node));
    printf ("Enter new element : ");
    scanf ("%d", &new-node->data);
    new-node->next = NULL;
    temp = start;
    while (temp->next != NULL)
    {
        temp = temp->next;
    }
    temp->next = new-node;
}

```

```

void insert - position()
{
    int pos, i = 0;
    new - node = (struct node*) malloc (sizeof ( struct
node));
    printf ("Enter the position : ");
    scanf ("%d", & pos);

    if (pos < 0)
    {
        printf ("Invalid");
    }
    else
    {
        temp = start;
        for (i = 0; i < pos; i++)
        {
            Temp = Temp -> next;
        }
        printf ("Enter new element : ");
        scanf ("%d", & new - node -> data);
        new - node -> next = temp -> next;
        temp -> next = new - node;
    }
}

void display()
{
    temp = start;
    printf ("Elements are : ");
    while (temp != NULL)
    {
        printf ("%d", temp -> data);
        temp = temp -> next;
    }
}

```

```
void main()
{
    int choice;
    while(1)
    {
        printf("menu : n 1. Create linked list,
               2. Insert at beginning 3. Insert
               end 4. Insert at any position,
               5. display 6. Exit");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1: createlist();
                      break;
            case 2: insert_beg();
                      break;
            case 3: insert_end();
                      break;
            case 4: insert_pos();
                      break;
            case 5: display();
                      break;
            case 6: exit(0);
                      break;
            default: printf("invalid input");
        }
    }
}
```

exit
29/1/24

OUTPUT:

Menu

Create linked list
end

1

Enter no. of elements : 3

Enter the element : 2

Enter the element : 4

Enter the element : 6

menu

1. Create linked list 2. Insert at beginning 3. Insert at end
4. Insert at any position 5. Display 6. Exit

2

Enter new element : 10

menu

1. Create linked list 2. Insert at beginning 3. Insert at end
4. Insert at any position 5. Display 6. Exit

3

Enter new element : 20

menu

1. Create linked list 2. Insert at beginning 3. Insert at end
4. Insert at any position 5. Display 6. Exit

4

Enter the position : 2

Enter new element : 30

menu

1. Create linked list 2. Insert at beginning 3. Insert at end
4. Insert at any position 5. Display 6. Exit

5

Elements are : 10 2 30 4 6 20

menu

1. Create linked list 2. Insert at beginning 3. Insert at end
4. Insert at any position 5. Display 6. Exit

6

Menu:

- 1.create linked list
- 2.Insert at beginnnning
- 3.Insert at end
- 4.insert at any position
- 5.Display
- 6.Exit

1

enter the number of elements:3

Enter the element:2

Enter the element:4

Enter the element:6

Menu:

- 1.create linked list
- 2.Insert at beginnnning
- 3.Insert at end
- 4.insert at any position
- 5.Display
- 6.Exit

2

Enter the new element:

5

Menu:

- 1.create linked list
- 2.Insert at beginnnning
- 3.Insert at end
- 4.insert at any position
- 5.Display
- 6.Exit

3

Enter the new element:

9

Menu:

- 1.create linked list
- 2.Insert at beginnning
- 3.Insert at end
- 4.insert at any position
- 5.Display
- 6.Exit

4

enter the position:

1

Enter the new element:

0

Menu:

- 1.create linked list
- 2.Insert at beginnning
- 3.Insert at end
- 4.insert at any position
- 5.Display
- 6.Exit

5

The elements are:

5 0 2 4 6 9

Menu:

- 1.create linked list
- 2.Insert at beginnning
- 3.Insert at end
- 4.insert at any position
- 5.Display
- 6.Exit

6

- >1 Write a program to implement singly linked list with the following operations
- a) Create a linked list
 - b) Selection of first element, specified and last element.
 - c) Display the contents of linked list

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

struct node
{
    int data;
    struct node * next;
};

struct node * start = NULL;
struct node * new_node;
struct node * temp;

void createlist()
{
    int i, n;
    printf("Enter the no. of elements:");
    scanf("%d", &n);
    for (i = 0; i < n; i++)
    {
        new_node = (struct node *) malloc
                    ( sizeof ( struct node ) );
        printf("Enter the %d element", i);
        scanf("%d", &new_node->data);
        new_node->next = NULL;
        if (start == NULL)
        {
            temp = start = new_node;
        }
        else
        {
            temp->next = new_node;
            temp = new_node;
        }
    }
}
```

else

{

 temp → next = new - node;
 temp = new - node;

}

}

void delete - beg()

{

 temp = start;
 if (start == NULL)

{

 printf ("The list is empty");

}

else

{

 start = temp → next;

 free (temp);

}

}

void delete - end()

{

 struct node * ptr;

 temp = start;

 while (temp → next != NULL)

{

 ptr = temp;

 temp = temp → next;

}

 if (temp == start)

{

 start = NULL;

}

 else

```
    ptr -> next = NULL ;  
}  
free (temp) ;  
}  
void delete_position()  
{  
    struct node * ptr ;  
    int pos, i ;  
    temp = start ;  
    printf (" Enter the position : ") ;  
    scanf ("%d", & pos) ;  
    for (i = 1; i < pos; i++)  
    {  
        temp = temp -> next ;  
    }
```

```
    ptr = temp -> next ;  
    temp -> next = ptr -> next ;  
    free (ptr) ;
```

```
}
```

```
void display()
```

```
{  
    temp = start ;  
    printf (" The elements are : ") ;  
    while (temp != NULL)  
    {  
        printf ("%d\t", temp -> data) ;  
        temp = temp -> next ;  
    }
```

```
}
```

```
void main()
```

```
{  
    int choice ;
```

```
    while (1)  
    {
```

```

printf ("Menu\n 1. Create linked list\n 2. Delete at beginning\n 3. Delete at end\n 4. Delete at any position\n 5. display\n 6. exit");
scanf ("%d", &choice);
switch (choice)
{
    case 1: createlist();
    break;
    case 2: delete - begin();
    break;
    case 3: delete - end();
    break;
    case 4: delete - position();
    break;
    case 5: display();
    break;
    case 6: exit(0);
    break;
    default: printf ("invalid input");
}
}

```

OUTPUT:

Menu

1. Create linked list
2. Delete at beginning
3. Delete at end
4. Delete at any position
5. display
6. Exit

Enter the no. of elements: 5

Enter the 1 element: 2

Enter the 2 element: 4

Enter the 3 element: 6

Enter the 4 element: 8

Enter the 5 element: 10

menu

1. Create linked list
2. Delete at beginning
3. Delete at end
4. Delete at any position
5. Display
6. Exit

2

menu

1. Create linked list
2. Delete at beginning
3. Delete at end
4. Delete at any position
5. Display
6. Exit

3

menu

1. Create linked list
2. Delete at beginning
3. Delete at end
4. Delete at any position
5. Display
6. Exit

5

The elements are : 4 6 8

menu

1. Create linked list
2. Delete at beginning
3. Delete at end
4. Delete at any position
5. Display
6. Exit

4

Enter the position : 1

menu

1. Create linked list
2. Delete at beginning
3. Delete at end
4. Delete at any position
5. Display
6. Exit

5

The elements are : 2 8

menu

1. Create linked list
2. Delete at beginning
3. Delete at end
4. Delete at any position
5. Display
6. Exit

6

Menu:

- 1.create linked list
- 2.Delete at beginnning
- 3.Delete at end
- 4.Delete at any position
- 5.Display
- 6.Exit

1

Enter the number of elements:

5

Enter the element:2

Enter the element:4

Enter the element:6

Enter the element:8

Enter the element:10

Menu:

- 1.create linked list
- 2.Delete at beginnning
- 3.Delete at end
- 4.Delete at any position
- 5.Display
- 6.Exit

2

Menu:

- 1.create linked list
- 2.Delete at beginnning
- 3.Delete at end
- 4.Delete at any position
- 5.Display
- 6.Exit

3

Menu:

- 1.create linked list
- 2.Delete at beginnning
- 3.Delete at end
- 4.Delete at any position
- 5.Display
- 6.Exit

5

The elements are: 4 6 8

Menu:

- 1.create linked list
- 2.Delete at beginnning
- 3.Delete at end
- 4.Delete at any position
- 5.Display
- 6.Exit

4

Enter the position:1

Menu:

- 1.create linked list
- 2.Delete at beginnning
- 3.Delete at end
- 4.Delete at any position
- 5.Display
- 6.Exit

5

The elements are: 4 8

Menu:

- 1.create linked list
- 2.Delete at beginnning
- 3.Delete at end
- 4.Delete at any position
- 5.Display
- 6.Exit

6