

5.2.2024

Q. → Create a Double linked list

→ Insert a new node to the left of the node.

→ Delete the node based on specific position.

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

```
struct node {
```

```
    int data;
```

```
    struct node *next;
```

```
    struct node *prev;
```

```
};
```

```
struct node *head = 0, *newnode, *temp;
```

```
void insert ()
```

```
{
```

```
    int i, n;
```

```
    printf ("Enter the no. of elements: \n");
```

```
    scanf ("%d", &n);
```

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

```
{
```

```
        newnode = (struct node *) malloc (sizeof (struct node));
```

```
        printf ("Enter the %d element: \n", i+1);
```

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

```
        newnode->prev = 0;
```

```
        newnode->next = 0;
```

```
        if (head == 0)
```

```
{
```

```
            temp = head = newnode;
```

```
            newnode->prev = temp;
```

```

    }
    temp = newnode;
}
}

```

void display ()

```

{
    temp = head;
    while (temp != 0)
    {
        printf ("%d \n", temp->data);
        temp = temp->next;
    }
}

```

void insert_left ()

```

{
    int node, i = 1;
    printf ("Enter the node \n");
    scanf ("%d", &node);
    temp = head;
    if (node < 1)
    {
        printf ("invalid position \n");
    }
    else if (node == 1)
    {
        newnode = (struct node *) malloc (sizeof (struct node));
        printf ("Enter the data \n");
        scanf ("%d", &newnode->data);
        newnode->prev = 0;
        head->prev = newnode;
        newnode->next = head;
        head = newnode;
    }
}

```

else

{

newnode = (struct node *) malloc (size of (struct node));

printf ("Enter data: ");

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

while (i < node-1)

{

temp = temp->next;

i++;

}

newnode->prev = temp;

newnode->next = temp->next;

temp->next = newnode;

newnode->next->prev = newnode;

}

}

void delete - front()

{

int pos, i=1;

temp = head;

printf ("Enter position: ");

scanf ("%d", & pos);

while (i < pos)

{

temp = temp->next;

i++;

}

temp->prev->next = temp->next;

temp->next->prev = temp->prev;

free (temp);

}

```
void main()
```

```
{
```

```
    int choice, num;
```

```
    printf("Enter question /n 1. create /n 2. display /n 3. insert at  
    left /n 4. delete at position /n 5. -1 to end /n");
```

```
    while(1);
```

```
{
```

```
    printf("Enter question /n");
```

```
    scanf("%d", &choice);
```

```
    if(choice == -1)
```

```
{
```

```
        printf("Completed /n");
```

```
    }
```

```
    else
```

```
{
```

```
        switch(choice)
```

```
{
```

```
            case 1: create();
```

```
            break;
```

```
            case 2: display();
```

```
            break;
```

```
            case 3: insert-left();
```

```
            break;
```

```
            case 4: delete-pos();
```

```
            break;
```

```
            default: printf("Invalid Input /n");
```

```
        }
```

```
    }
```

```
}
```

```
}
```


Output

Enter operation

1. push
2. display
3. insert to left
4. delete from a position
5. -1 to stop

enter operation

1

Enter the no of elements: 3

Enter the 1 element: 2

Enter the 2 element: 3

Enter the 3 element: 4

Enter operation: 2

2

3

4

Enter operation: 3

Enter the node: 1

Enter data: 5

Enter operation: 2

5

2

3

4

Enter operation: 3

Enter the node: 3

Enter data: 6

Enter operation: 2

5

2

6

3

4

Enter operation: 4

Enter position: 3

Enter position: 2

5

2

3

4

Enter operation: 1

operation completed.

marks

5/5

enter operation

1.create

2.display

3.insert at left

4.delete at position

5.-1 to end

enter operation

1

Enter the no. of elements:

3

Enter the 1 element :

2

Enter the 2 element :

3

Enter the 3 element :

4

enter operation

2

2

3

4

enter operation

3

enter the node

1

enter data

5

enter operation

2

5



C:\Users\bmsce\Desktop\1bm X



```
2
3
4
enter operation
3
enter the node
3
enter data
6
enter operation
2
5
2
6
3
4
enter operation
4
enter position
3
enter operation
2
5
2
3
4
enter operation
-1
completed
```