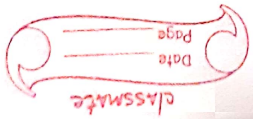
Week 8 Lab program:- Linked list:-

```
#include<stdio.h>
#include<stdlib.h>
#include<process.h>
struct node
{
    int info;
    struct node *link;
};
typedef struct node *NODE;
NODE getnode()
{
    NODE x;
    x = (NODE) malloc (sizeof(struct node));
    if (x==NULL)
    {
        printf("Memory is full\n");
        exit(0);
    }
    return x;
}
void freenode(NODE x)
{
    free(x);
}
NODE insert_front(NODE first, int item)
{
    NODE temp;
    temp = getnode();
    temp->info = item;
    temp->link = NULL;
    if (first == NULL)
        return temp;
    temp->link = first;
    first = temp;
    return first;
}
```

```
NODE delete_front(NODE first)
{
    NODE temp;
    if(first==NULL)
    {
        printf("List is empty, cannot delete item\n");
        return first;
    }
    temp = first;
    temp = temp -> link;
    printf("Item deleted at the front end is: %d\n", first -> info);
    free(first);
    return temp;
}

NODE insert_rear(NODE first, int item)
{
    NODE temp, cur;
    temp = getnode();
    temp -> info = item;
    temp -> link = NULL;
    if(first == NULL)
        return temp;
    cur = first;
    while(cur -> link != NULL)
        cur = cur -> link;
    cur -> link = temp;
    return first;
}
```

```
NODE delete_rear(NODE first)
{
    NODE cur, prev;
    if (first = NULL)
    {
      printf("The list is empty, cannot Delete Item\n");
      return first;
    }
   if ( first -> link == NULL)
   {
    printf("Item Deleted is : %d", first -> info);
    free(first);
    return NULL;
   }
   prev = NULL;
   cur = first;
   while(cur -> link != NULL)
   {
     prev = cur;
     cur = cur -> link;
} }
   printf("Item Deleted at the rear-end is : %d", cur -> info);
   free (cur);
   prev -> link = NULL;
   return first;
} }
{ NODE insert_pos (int item, int pos, NODE first)
  {
   NODE temp;
   NODE prev, cur;
    int count;
   temp = getnode();
   temp -> info = item;
   temp -> link = NULL;
```



```
if (first == NULL && pos == 1)
    return temp;
if ( first == NULL)
{
    printf(" Invalid position\n");
    return first;
}
if (pos == 1)
{
  temp -> link = first;
  return temp;
}
count = 1;
prev = NULL;
cur = first;
while ( cur != NULL && count != pos)
{
    prev = cur;
    cur = cur -> link;
    count ++;
}
if ( count == pos)
{
  prev -> link = temp;
  temp -> link = cur;
  return first;
}
printf (" IP\n");
 return first;
}
```

```
NODE order_list(int item, NODE first)
{
    NODE temp, prev, cur;
    temp = getnode( );
    temp -> info = item;
    temp -> link = NULL;
    if (first == NULL)
        return temp;
    if (item < first -> info)
    {
        temp -> link = first;
        return temp;
    }
    prev = NULL;
    cur = first;
    while (cur != NULL && item > cur -> info)
    {
        prev = cur;
        cur = cur -> link;
    }
    prev -> link = temp;
    temp -> link = cur;
    return first;
}

NODE delete_info(int key, NODE first)
{
    NODE prev, cur;
    if ( first == NULL)
    {
        printf("List is empty \n");
        return NULL;
    }
```

```
if (Key == first -> info)
{
    cur = first;
    first = first -> link;
    freenode (cur);
    return first;
}
prev = NULL;
cur = first;
while ( cur != NULL)
{
    if ( key == cur-> info) break;
    prev = cur;
    cur = cur -> link;
}
if (cur == NULL)
{
    printf("Search is unsuccessfull \n");
    return first;
}
prev -> link = cur-> link;
printf(" Key deleted is %d", cur-> info);
free node (cur);
return first;
}
void display (NODE first)
{
    NODE temp;
    if ( first == NULL)
        printf(" Lis is empty, cannot Display item \n")
    printf("\n x x  x x  x x  x x x \n");
    for ( temp = first ; temp != NULL; temp = temp -> link)
    {
        printf(" %d \n" temp -> info);
    }
    printf("\n x x x x x x x  x x x x x x x \n");
}
```

```
void main()
{
    int item, choice, pos, key;
    NODE first = NULL;
    for(;;)
    {
        printf("\n 1: Insert_front \n 2: Delete_front \n 3: Insert_rear \n");
        printf("\n 4: Delete_rear \n 5: Insert_pos \n 6: Order_list ");
        printf("\n 7: Delete into_ordered_list \n 8: Display \n 9: Exit");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1: printf("Enter the item at front-end: ");
                    scanf("%d", &item);
                    first = insert_front(first, item);
                    break;
            case 2: first = delete_front(first);
                    break;
            case 3: printf("Enter the item at rear-end: ");
                    scanf("%d", &item);
                    first = insert_rear(first, item);
                    break;
            case 4: first = delete_rear(first);
                    break;
            case 5: printf("Enter the position: ");
                    scanf("%d", &pos);
                    first = insert_pos(item, pos, first);
                    break;
            case 6: printf("Enter the item: ");
                    scanf("%d", &item);
                    first = order_list(item, first);
                    break;
```

```
case 7: printf("Enter the key to be deleted: ");
        scanf("%d", &key);
        first = delete_info(key, first);
        break;
case 8: display(first);
        break;
default: exit(0);
         break;
    }
  }
}
```

```
1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:insert_pos
6:display_list
7:Exit
Enter the choice: 1
Enter the item at front-end: 10

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:insert_pos
6:display_list
7:Exit
Enter the choice: 3
Enter the item at rear-end: 20

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:insert_pos
6:display_list
7:Exit
Enter the choice: 1
Enter the item at front-end: 30

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:insert_pos
6:display_list
7:Exit
Enter the choice: 6

**********************************************************************
30
10
20

**********************************************************************
```

```
1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:insert_pos
6:display_list
7:Exit
Enter the choice: 2
Item Deleted at the front-end is: 30

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:insert_pos
6:display_list
7:Exit
Enter the choice: 4
Item Deleted at the rear-end is : 20
1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:insert_pos
6:display_list
7:Exit
Enter the choice: 6

**************************************************************************
10

**************************************************************************

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:insert_pos
6:display_list
7:Exit
Enter the choice: 7
PS D:\DS 3rd Sem Notes\DS Lab\Week 8>
```

```
1:PUSH
2:POP
3:insert_pos
4:display_list
5:Exit
Enter the choice: 2
Item Deleted at the front-end is: 30

1:PUSH
2:POP
3:insert_pos
4:display_list
5:Exit
Enter the choice: 2
Item Deleted at the front-end is: 20

1:PUSH
2:POP
3:insert_pos
4:display_list
5:Exit
Enter the choice: 2
Item Deleted at the front-end is: 10

1:PUSH
2:POP
3:insert_pos
4:display_list
5:Exit
Enter the choice: 2
List is empty, Cannot Delete item
```

```
1:PUSH
2:POP
3:insert_pos
4:display_list
5:Exit
Enter the choice: 1
Enter the item at front-end: 10

1:PUSH
2:POP
3:insert_pos
4:display_list
5:Exit
Enter the choice: 1
Enter the item at front-end: 20

1:PUSH
2:POP
3:insert_pos
4:display_list
5:Exit
Enter the choice: 1
Enter the item at front-end: 30

1:PUSH
2:POP
3:insert_pos
4:display_list
5:Exit
Enter the choice: 4

**********************************************************************
30
20
10

**********************************************************************

1:PUSH
2:POP
3:insert_pos
4:display_list
5:Exit
Enter the choice: 2
Item Deleted at the front-end is: 30
```

```
1:Insert_rear
2:Delete_front
3:display_Queue
4:Exit
Enter the choice: 1
Enter the item at rear-end: 12

1:Insert_rear
2:Delete_front
3:display_Queue
4:Exit
Enter the choice: 1
Enter the item at rear-end: 13

1:Insert_rear
2:Delete_front
3:display_Queue
4:Exit
Enter the choice: 1
Enter the item at rear-end: 14

1:Insert_rear
2:Delete_front
3:display_Queue
4:Exit
Enter the choice: 3

**********************************************************************
12
13
14

**********
```

```
1:Insert_rear
2:Delete_front
3:display_Queue
4:Exit
Enter the choice: 2
Item Deleted at the front-end is: 12

1:Insert_rear
2:Delete_front
3:display_Queue
4:Exit
Enter the choice: 2
Item Deleted at the front-end is: 13

1:Insert_rear
2:Delete_front
3:display_Queue
4:Exit
Enter the choice: 2
Item Deleted at the front-end is: 14

1:Insert_rear
2:Delete_front
3:display_Queue
4:Exit
Enter the choice: 2
List is empty, Cannot Delete item

1:Insert_rear
2:Delete_front
3:display_Queue
4:Exit
Enter the choice: 4
PS D:\DS 3rd Sem Notes\DS Lab\Week 8>
```

```
1:Insert in Order List
2:Delete Item
3:display
4:Exit
Enter the choice: 1
Enter the item to be inserted in ordered list: 13

1:Insert in Order List
2:Delete Item
3:display
4:Exit
Enter the choice: 1
Enter the item to be inserted in ordered list: 15

1:Insert in Order List
2:Delete Item
3:display
4:Exit
Enter the choice: 1
Enter the item to be inserted in ordered list: 14

1:Insert in Order List
2:Delete Item
3:display
4:Exit
Enter the choice: 1
Enter the item to be inserted in ordered list: 12

1:Insert in Order List
2:Delete Item
3:display
4:Exit
Enter the choice: 3

**********************************************************************
12
13
14
15

**********************************************************************
```

```
1:Insert in Order List
2:Delete Item
3:display
4:Exit
Enter the choice: 2
Enter the key to be Deleted: 14
key deleted is 14
1:Insert in Order List
2:Delete Item
3:display
4:Exit
Enter the choice: 2
Enter the key to be Deleted: 12

1:Insert in Order List
2:Delete Item
3:display
4:Exit
Enter the choice: 3

**********************************************************************
13
15

**********************************************************************

1:Insert in Order List
2:Delete Item
3:display
4:Exit
Enter the choice: 4
PS D:\DS 3rd Sem Notes\DS Lab\Week 8>
```