25 Programs on singly linear linked list:

===

Linked list structure :

```
struct node {

        int no; // Data element
        struct node *next, // Address of next node
 };
```

===

Program 1.

Write a program which adds a new node in singly linear linked list at first position.

```
Function Prototype :
        int InsertFirst(
                struct node **Head, int no
                );
```

        Input linked list : |10|->|20|->|30|->|40|->|50|->|60|->|70|
        Input data element : 21
        Output linked list : |21|->|10|->|20|->|30|->|40|->|50|->|60|->|70|

Program 2.

Write a program which adds a new node in a singly linear linked list at last position.

Function Prototype :

```
        int InsertLast(
                struct node **Head, int no
        );
```

Input linked list : |10|->|20|->|30|->|40|->|50|->|60|->|70|

Input data element : 21

Output linked list : |10|->|20|->|30|->|40|->|50|->|60|->|70|->|21|

Program 3.

Write a program which adds a new node in singly linear linked list at specified position.

Function Prototype :

```
int InsertAtPosition(
        struct node **Head, int no, int pos
);
```

Input linked list : |10|->|20|->|30|->|40|->|50|->|60|->|70|

Input data element : 21

Input position : 4

Output linked list : |10|->|20|->|30|->|21|->|40|->|50|-

->|60|>|70|

Program 4.

Write a program which removes the first node from a singly linear linked list.

```
Function Prototype :
int DeleteFirst(
        struct node **Head
);

Input linked list : |10|->|20|->|30|->|40|->|50|->|60|->|70|
Output linked list : |20|->|30|->|40|->|50|->|60|->|70|
```

Program 5.

Write a program which removes the last node from a singly linear linked list.

```
Function Prototype :
int DeleteLast(
                struct node **Head
);

Input linked list : |10|->|20|->|30|->|40|->|50|->|60|->|70|
Output linked list : |10|->|20|->|30|->|40|->|50|->|60|
```

Program 6.

Write a program which removes nodes from singly linear linked lists which are at specified positions.

```
Function Prototype :
int DeleteAtPosition(
                struct node **Head, int pos
);
```

```
Input linked list : |10|->|20|->|30|->|40|->|50|->|60|->|70|
Input position : 4
Output linked list : |10|->|20|->|30|->|50|->|60|->|70|
```

Program 7.

Write a program which displays contents of singly linear linked lists.

```
Function Prototype :
int Display(
        struct node *Head
);
```

Program 8.

Write a program which searches for the first occurrence of a particular element from a singly linear linked list.
Function should return the position at which element is found.

```
Function Prototype :
```

```
int SearchFirstOcc(
        struct node *Head, int no
);
```

Input linked list : |10|->|20|->|30|->|40|->|50|->|30|->|70|
Input element : 30
Output : 3

Program 9.

Write a program which searches the last occurrence of a particular element from a singly linear linked list.
Function should return the position at which element is found.

Function Prototype :

```
int SearchLastOcc(
        struct node *Head , int no
);
```

Input linked list : |10|->|20|->|30|->|40|->|50|->|30|->|70|
Input element : 30
Output : 6

Program 10.

Write a program which searches the second last occurrence of a particular element from a singly linear linked list.
Function should return the position at which element is found.

Function Prototype :

```
int SearchSecLastOcc(
        struct node *Head , int no
        );
```

Input linked list : |10|->|20|->|30|->|40|->|30|->|30|->|70|
Input element : 30
Output : 5

Program 11.

Write a program which searches all occurrences of a particular element from a singly linear linked list. Function should return the number of occurance of that element.

```
Function Prototype :
        int SearchAll(
                struct node *Head , int no
         );
```

Input linked list : |10|->|20|->|30|->|40|->|30|->|30|->|70|
Input element : 30
Output : 3

Program 12.

Write a program which accepts two singly linear linked lists from user and concat source linked list after destination linked list.

```
Function Prototype :
        int ConcatList(
                struct node **Src , struct node **Dest
        );
```

Input source linked list : |30|->|30|->|70|
Input destination linked list : |10|->|20|->|30|->|40| Output destination linked list : |10|->|20|->|30|->|40|->|30|- >|30|>|70|

Program 13.

Write a program which accepts two singly linear linked lists from user and concat first N elements of source linked list after destination linked list.

Function Prototype :

```
int ConcatFirstN(
        struct node **Src ,struct node **Dest, int no
);
```

Input source linked list : |30|->|30|->|70|
Input destination linked list : |10|->|20|->|30|->|40|
Input number of elements : 2
Output destination linked list : |10|->|20|->|30|->|40|->|30|->|30|

Program 14.

Write a program which accepts two singly linear linked lists from user and concat last N elements of source linked list after destination linked list.

Function Prototype :

```
int ConcatLastN(
        struct node **Src ,struct node **Dest, int no
);
```

Input source linked list : |30|->|30|->|70|
Input destination linked list : |10|->|20|->|30|->|40|
Input number of elements : 2
Output destination linked list : |10|->|20|->|30|->|40|->|30|->|70|

Program 15.

Write a program which accepts two singly linear linked list from user and also accept range and concat elements of source singly linear linked list
from that range after a singly linear destination linked list.

Function Prototype :

```
int ConcatListRange(
        struct node **Src, struct node **Dest , int start , int end );
```

Input source linked list : |30|->|30|->|70|->|80|->|90|->|100|
Input destination linked list : |30|->|40|
Input starting range : 2
Input ending range : 5
Output destination linked list : |30|->|40|->|30|->|70|->|80|->|90|

Program 16.

Write a program which copies contents of source singly linear linked list to singly linear destination linked list.

Function Prototype :

```
int LLCopy(
        struct node **Src, struct node **Dest
);
```

Input source linked list : |30|->|30|->|70|->|80|->|90|->|100|

Input destination linked list : Empty (NULL)

Output destination linked list : |30|->|30|->|70|->|80|->|90|->|100|

Program 17.

Write a program which copies first N contents of singly linear source linked list to destination singly linear linked list.

Function Prototype :

```
int LLNCopy(
        struct node **Src , struct node **Dest, int no '
);
```

Input source linked list : |30|->|30|->|70|->|80|->|90|->|100|

Input destination linked list : Empty (NULL)

Input no : 4

Output destination linked list : |30|->|30|->|70|->|80|

Program 18.

Write a program which copies the last N contents of source singly linear linked list to destination singly linear linked list.

```
Function Prototype :
        int LLINCopy(
                 struct node **Src , struct node **Dest , int no
);
```

Input source linked list : |30|->|30|->|70|->|80|->|90|->|100|
Input destination linked list : Empty (NULL)
Input no : 4
Output destination linked list : |70|->|80|->|90|->|100|

Program 19.

Write a program which copies contents of source singly linear linked list to destination singly linear linked list which lies between the particular range which is accepted from the user.

```
Function Prototype :
        int LLCopyRange(
                    struct node **Src, struct node **Dest, int start, int end
        );
```

Input source linked list : |30|->|30|->|70|->|80|->|90|->|100|
Input destination linked list : Empty (NULL)
Input starting range : 2
Input starting range : 5
Output destination linked list : |30|->|70|->|80|->|90|

Program 20.

Write a program which copies alternate contents of source singly linear linked list to destination singly linear linked list.

```
Function Prototype :
        int LLCopyAlt(
                struct node **Src, struct node **Dest
        );
```

Input source linked list : |30|->|30|->|70|->|80|->|90|->|100|->|110|

Input destination linked list : Empty (NULL)

Output destination linked list : |30|->|70|->|90|->|110|

Program 21.

Write a program which copies contents of source singly linear linked list whose addition of digits is even number to destination singly linear linked list.

```
Function Prototype :
        int LLCopyEx(
                struct node **Src , struct node **Dest
        );
```

Input source linked list : |30|->|33|->|73|->|80|->|90|->|100|->|110|

Input destination linked list : Empty (NULL)

Output destination linked list : |33|->|77|->|110|

Program 22.

Write a program which copies contents of source singly linear linked list whose addition of digits is prime number to destination singly linear linked list.

```
Function Prototype :
        int LLCopyEx(
                struct node **Src , struct node **Dest
        );
```

Input source linked list : |30|->|32|->|73|->|80|->|70|->|110|->|112|

Input destination linked list : Empty (NULL)

Output destination linked list : |30|->|30|->|70|

Program 23.

Write a program which accept source singly linear linked list and destination singly linear linked list and check whether the source list is a sublist of the destination list. Function returns first position at which sub list found.

```
Function Prototype :
        int SubList(
                struct node **Src, struct node **Dest
        );
```

Input source linked list : |73|->|80|->|70|

```
        Input dest
list:|10|->|73|->|80|->|17|->|22|->|73|->|80|- >|70|-|21|
        Output : First Sub list found at position 6
```

Program 24.

Write a program which accept source singly linear linked list and destination singly linear linked list and check whether the source list is a sublist of the destination list. Function returns last position at which sub list found.

```
        Function Prototype :
                int SubList(
                        struct node **Src , struct node **Dest
                );
        Input source linked list : |73|->|80|->|70|
        Input dest
list:|10|->|73|->|80|->|70|->|22|->|73|->|80|- >|70|-|21|
        Output : Last Sub list found at position 6
```

Program 25.

Write a program which accepts source singly linear linked list from user and copy the contents into destination singly linear linked in ascending order.

```
        Function Prototype :
                int CopyAsc(
                    struct node **Src , struct node **Dest
                );
```

Input source linked list : |110|->|73|->|10|->|80|->|70|->|12|

Input destination linked list: Empty (NULL)

Output destination linked list: |10|->|12|->|70|->|73|->|80|->|110|