1.Write a program which accept singly linear linked list and swap consecutive elements from that linked list.

Function Prototype : int Swap(PPNODE Src);

Input linked list : |10|->|20|->|30|->|40|->|50|->|60|->|70|->|80|->|90|

Output linked list : |20|->|10|->|40|->|30|->|60|->|50|->|80|->|70|->|90|

2.Write a program which accept source and destination linked list from user and we have to insert source linked list in destination linked list at given position.

Function Prototype : int InsertAt(PPNODE Src, PPNODE dest, int pos);

Input src linked list : |10|->|20|->|30|->|40|

Input dest linked list: |50|->|60|->|70|->|80|

Input position : 3

Output dest linked list: |50|->|60|->|10|->|20|->|30|->|40|->|70|->|80|

3.Write a program which accepts two singly linear linked list from user and check whether they are intersecting or not.

Function Prototype : BOOL ListIntersect( PNODE First, PNODE Second);

4.Write a program which accept singly linear linked list from and check whether linked list is palindrome or not.

Function Prototype : BOOL ListPallindrom(PNODE First );

Input linked list : |10|->|20|->|30|->|40|->|30|->|20|->|10|

Output : The given linked list is palindrome.

5.Write a program which accepts two singly linear linked list from user and compare both linked list. Function Prototype : BOOL ListCmp( PNODE First, PNODE Second);

Input linked list 1 : |60|->|70|->|50|->|40|->|30|->|10|->|20|

Input linked list 2 : |60|->|70|->|50|->|40|->|30|->|10|->|20|

Output linked list : Both the linked list are equal.

6.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(PPNODE Src, PPNODE 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|

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

Function Prototype : int SubList(PPNODE Src, PPNODE 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

8.Write a program which accept source singly linear linked list from user and copy the contents int o destination singly linear linked in ascending order.

Function Prototype : int CopyAsc(PPNODE Src,PPNODE 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|

9. Write a program which accept source singly linear linked list from user and reverse that linear linked. Function Prototype : int ListReverse(PPNODE Src);

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

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

10.Write a program which accept singly linear linked list from user and check whether list contains loop or not.

Function Prototype : int LoopChk(PNODE Src);

To check working of this function create loop in the linked list

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

Function Prototype : int ConcatList( PPNODE Src, PPNNODE 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|

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

Function Prototype : int ConFirstN( PPNODE Src, PPNODE 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|

13.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 singly linear destination linked list.

Function Prototype : int ConcatListRange( PPNODE Src, PPNODE 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|

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

Function Prototype : int LLCopy(PPNODE Src, PPNODE 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|

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

Function Prototype : int LLNCopy(PPNODE Src, PPNODE 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|

16. Write a program which displays all elements which are perfect from singly linear linked list.

Function Prototype :int DisplayPerfect( PNODE Head);

Input linked list : |11|->|28|->|17|->|41|->|6|->|89|

Output : 6 28

17. Write a program which displays all elements which are prime from singly linear linked list.

Function Prototype :int DisplayPrime( PNODE Head);

Input linked list : |11|->|20|->|17|->|41|->|22|->|89|

Output : 11 17 41 89

18. Write a program which returns addition of all even element from singly linear linked list.

Function Prototype :int AdditionEven( PNODE Head);

Input linked list : |11|->|20|->|32|->|41|

Output : 52

19. Write a program which return second maximum element from singly linear linked list.

Function Prototype :int SecMaximum( PNODE Head);

Input linked list : |110|->|230|->|320|->|240|

Output : 240

20. Write a program which display addition of digits of element from singly linear linked list.

Function Prototype :int SumDigit( PNODE Head);

Input linked list : |110|->|230|->|20|->|240|->|640| Output : 2 5 2 6 10

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

Function Prototype :int SearchFirstOcc( PNODE Head , int no );

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

Input element : 30

Output : 3

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

Function Prototype :int SearchLastOcc( PNODE Head, int no );

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

Input element : 30

Output : 6

23. Write a program which returns addition of all element from singly linear linked list.

Function Prototype :int Addition( PNODE Head);

Input linked list : |10|->|20|->|30|->|40|

Output : 100

24. Write a program which return largest element from singly linear linked list.

Function Prototype :int Maximum( PNODE Head);

Input linked list : |110|->|230|->|320|->|240|

Output : 320

25. Write a program which return smallest element from singly linear linked list.

Function Prototype :int Minimum( PNODE Head);

Input linked list : |110|->|230|->|20|->|240|->|640|

Output : 2