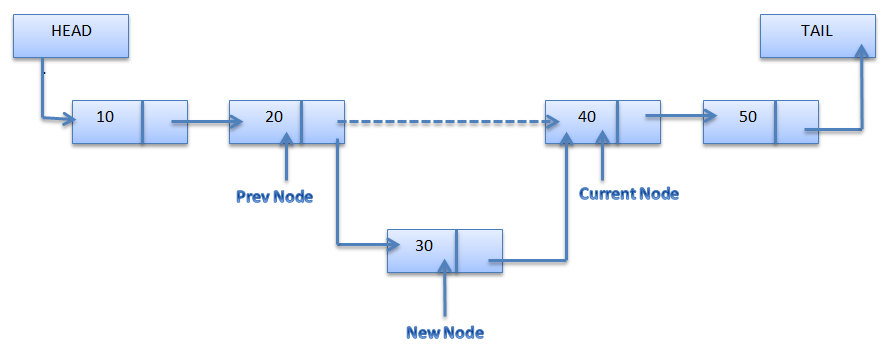
Insertion At Location in linear linked list

Algorithm

1. InsertAtlocDll(info,next,start,end,loc,size)
2. 1.set nloc = loc-1 , n=1
3. 2.create a new node and address in assigned to ptr.
4. 3.check[overflow] if(ptr=NULL)
5. write:overflow and exit
6. 4.set Info[ptr]=item;
7. 5.if(start=NULL)
8. set next[ptr] = NULL
9. set start = ptr
10. else if(nloc<=size)
11. repeat steps a and b while(n != nloc)
13. a. loc = next[loc]
14. b. n = n+1
15. [end while]
16. next[ptr] = next[loc]
17. next[loc] = ptr
18. else
19. set last = start;
20. repeat step (a) while(next[last]!= NULL)
21. a. last=next[last]
22. [end while]
23. last->next = ptr ;
24. [end if]
25. 6.Exit.



Fuction For Insert at Location

1. void insertAtloc(node \*\*start,int item , int i,int k )
2. {
3. node \*ptr,\*loc,\*last;
4. int n=1 ;
5. i=i-1;
6. ptr=(node\*)malloc(sizeof(node));
7. ptr->info=item;
8. loc = \*start ;
9. if(\*start==NULL)
10. {
11. ptr->next = NULL ;
12. \*start = ptr ;
13. }
14. else if(i<=k)
15. { while(n != i)
16. {
17. loc=loc->next;
18. n++;
19. }
20. ptr->next = loc->next ;
21. loc->next = ptr ;


25. }
26. else
27. {
28. last = \*start;
29. while(last->next != NULL)
30. {last=last->next;
31. }
32. last->next = ptr ;
33. }
34. }

C programe for insertion at location in linear linked list

1. #include<stdio.h>
2. #include<malloc.h>
3. #include<conio.h>
4. typedef struct Node
5. {
6. int info ;
7. struct Node \*next;
8. }node;
9. void createsig(node\*\*,int);
10. void insertAtloc(node \*\*,int,int,int);
11. void display(node \*);
13. void main()
14. {
15. int ch, item, pos,loc,i;
16. node \*start ;
17. start = NULL;
18. clrscr();
19. printf("Enter number of node: ");
20. scanf("%d",&i);
21. createsig(&start,i);
22. printf("\nThe list is : ");
23. display(start);
24. printf("\nEnter the loc : ");
25. scanf("%d",&loc);
26. printf("\n\nEnter the item to be inserted at loc : ");
27. scanf("%d",&item);
28. insertAtloc(&start,item,loc,i);
29. printf("\nNow the list is : ");
30. display(start);
31. getch();
32. }
33. void createsig(node \*\*start,int i)
34. { int item ,k=1;
35. while(i)
36. { node \*ptr,\*last;
37. printf("\nEnter the info for node %d : ",k);
38. scanf("%d",&item);
39. ptr=(node\*)malloc(sizeof(node));
40. ptr->info=item;
41. ptr->next=NULL;
42. if(\*start==NULL)
43. {
44. \*start = ptr ;
45. }
46. else
47. { last = \*start;
48. while(last->next != NULL)
49. {last=last->next;
50. }
51. last->next = ptr ;
52. }
53. i--;
54. k++;
55. }
56. }
57. void insertAtloc(node \*\*start,int item , int i,int k )
58. {
59. node \*ptr,\*loc,\*last;
60. int n=1 ;
61. i=i-1;
62. ptr=(node\*)malloc(sizeof(node));
63. ptr->info=item;
64. loc = \*start ;
65. if(\*start==NULL)
66. {
67. ptr->next = NULL ;
68. \*start = ptr ;
69. }
70. else if(i<=k)
71. { while(n != i)
72. {
73. loc=loc->next;
74. n++;
75. }
76. ptr->next = loc->next ;
77. loc->next = ptr ;


81. }
82. else
83. {
84. last = \*start;
85. while(last->next != NULL)
86. {last=last->next;
87. }
88. last->next = ptr ;
89. }
90. }
92. void display(node \*start)
93. {
94. while(start !=NULL)
95. { printf("\t %d",start->info);
96. start = start->next;
97. }
98. }

Output

