**Create a simple linked list with 3 nodes**

// A simple C program to introduce a linked list

#include<stdio.h>

#include<stdlib.h>

struct node

{

  int data;

  struct node \*next;

};

// Program to create a simple linked list with 3 nodes

int main()

{

  struct node\* head = NULL;

  struct node\* second = NULL;

  struct node\* third = NULL;

  // allocate 3 nodes in the heap

  head = (struct node\*)malloc(sizeof(struct node));

  second = (struct node\*)malloc(sizeof(struct node));

  third = (struct node\*)malloc(sizeof(struct node));

  /\* Three blocks have been allocated  dynamically.

     We have pointers to these three blocks as first, second and third

       head           second           third

        |                |               |

        |                |               |

    +---+-----+     +----+----+     +----+----+

    | #  | #  |     | #  | #  |     |  # |  # |

    +---+-----+     +----+----+     +----+----+

   # represents any random value.

   Data is random because we haven’t assigned anything yet  \*/

  head->data = 1; //assign data in first node

  head->next = second; // Link first node with the second node

  /\* data has been assigned to data part of first block (block

    pointed by head).  And next pointer of first block points to

    second.  So they both are linked.

       head          second         third

        |              |              |

        |              |              |

    +---+---+     +----+----+     +-----+----+

    | 1  | o----->| #  | #  |     |  #  | #  |

    +---+---+     +----+----+     +-----+----+

  \*/

  second->data = 2; //assign data to second node

  second->next = third; // Link second node with the third node

  /\* data has been assigned to data part of second block (block pointed by

     second). And next pointer of the second block points to third block.

    So all three blocks are linked.

       head         second         third

        |             |             |

        |             |             |

    +---+---+     +---+---+     +----+----+

    | 1  | o----->| 2 | o-----> |  # |  # |

    +---+---+     +---+---+     +----+----+      \*/

  third->data = 3; //assign data to third node

  third->next = NULL;

  /\* data has been assigned to data part of third block (block pointed

    by third). And next pointer of the third block is made NULL to indicate

    that the linked list is terminated here.

     We have the linked list ready.

           head

             |

             |

        +---+---+     +---+---+       +----+------+

        | 1  | o----->|  2  | o-----> |  3 | NULL |

        +---+---+     +---+---+       +----+------+

    Note that only head is sufficient to represent the whole list.  We can

    traverse the complete list by following next pointers.    \*/

  return 0;

#Alternative approach of **Create a simple linked list with 3 nodes**

#include<stdio.h>

struct node

{

int value;

struct node\* next;

};

void main()

{

struct node a,b,c;

struct node\* i;

struct node addfirst,addlast;

int item;

clrscr();

a.value=100;

b.value=200;

c.value=300;

a.next=&b;

b.next=&c;

c.next=NULL;

for(i=&a;i!=NULL;i=i->next)

printf("%d ",i->value);

getch();

}

Insertion of a node into the linked list

#include<stdio.h>

struct node

{

int value;

struct node\* next;

};

void main()

{

struct node a,b,c;

struct node\* i;

struct node addfirst,addlast;

int item;

clrscr();

a.value=100;

b.value=200;

c.value=300;

a.next=&b;

b.next=&c;

c.next=NULL;

for(i=&a;i!=NULL;i=i->next)

printf("%d ",i->value);

printf("\nAdd first\n");

scanf("%d",&item);

addfirst.value=item;

addfirst.next=&a;

for(i=&addfirst;i!=NULL;i=i->next)

printf("%d ",i->value);

printf("\nAdd last\n");

scanf("%d",&item);

c.next=&addlast;

addlast.value=item;

addlast.next=NULL;

for(i=&addfirst;i!=NULL;i=i->next)

printf("%d ",i->value);

getch();

}