**Practical -: Create Linked List**

**Name -: Aditya Babaso Birangaddi**

**Code -:**

**#include<stdio.h>**

**#include<stdlib.h>**

**struct node**

**{**

**int data;**

**struct node \*next;**

**};**

**struct node \*head;**

**void beginsert ();**

**void lastinsert ();**

**void randominsert();**

**void display();**

**void main ()**

**{**

**int choice =0;**

**while(choice != 9)**

**{**

**printf("\n\n\*\*\*\*\*\*\*\*\*Main Menu\*\*\*\*\*\*\*\*\*\n");**

**printf("\nChoose one option from the following list ...\n");**

**printf("\n===============================================\n");**

**printf("\n 1.Insert in begining\n2.Insert at last\n3.Insert at any random location\n4.Show\n5.Exit\n");**

**printf("\nEnter your choice?\n");**

**scanf("\n%d",&choice);**

**switch(choice)**

**{**

**case 1:**

**beginsert();**

**break;**

**case 2:**

**lastinsert();**

**break;**

**case 3:**

**randominsert();**

**break;**

**case 4:**

**display();**

**break;**

**case 5:**

**exit(0);**

**break;**

**default:**

**printf("Please enter valid choice..");**

**}**

**}**

**}**

**void beginsert()**

**{**

**struct node \*ptr;**

**int item;**

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

**if(ptr == NULL)**

**{**

**printf("\nOVERFLOW");**

**}**

**else**

**{**

**printf("\nEnter value\n");**

**scanf("%d",&item);**

**ptr->data = item;**

**ptr->next = head;**

**head = ptr;**

**printf("\nNode inserted");**

**}**

**}**

**void lastinsert()**

**{**

**struct node \*ptr,\*temp;**

**int item;**

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

**if(ptr == NULL)**

**{**

**printf("\nOVERFLOW");**

**}**

**else**

**{**

**printf("\nEnter value?\n");**

**scanf("%d",&item);**

**ptr->data = item;**

**if(head == NULL)**

**{**

**ptr -> next = NULL;**

**head = ptr;**

**printf("\nNode inserted");**

**}**

**else**

**{**

**temp = head;**

**while (temp -> next != NULL)**

**{**

**temp = temp -> next;**

**}**

**temp->next = ptr;**

**ptr->next = NULL;**

**printf("\nNode inserted");**

**}**

**}**

**}**

**void randominsert()**

**{**

**int i,loc,item;**

**struct node \*ptr, \*temp;**

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

**if(ptr == NULL)**

**{**

**printf("\nOVERFLOW");**

**}**

**else**

**{**

**printf("\nEnter element value");**

**scanf("%d",&item);**

**ptr->data = item;**

**printf("\nEnter the location after which you want to insert ");**

**scanf("\n%d",&loc);**

**temp=head;**

**for(i=0;i<loc;i++)**

**{**

**temp = temp->next;**

**if(temp == NULL)**

**{**

**printf("\ncan't insert\n");**

**return;**

**}**

**}**

**ptr ->next = temp ->next;**

**temp ->next = ptr;**

**printf("\nNode inserted");**

**}**

**}**

**void display()**

**{**

**struct node \*ptr;**

**ptr = head;**

**if(ptr == NULL)**

**{**

**printf("Nothing to print");**

**}**

**else**

**{**

**printf("\n printing values . . . . .\n");**

**while (ptr!=NULL)**

**{**

**printf("\n%d",ptr->data);**

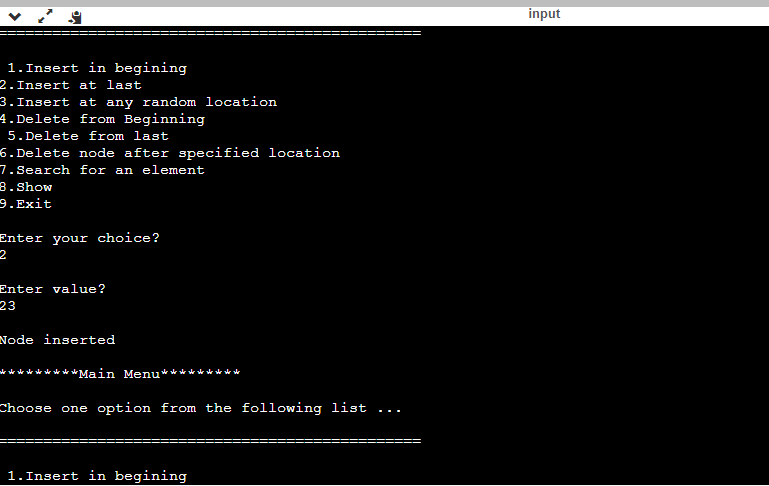
**ptr = ptr -> next;**

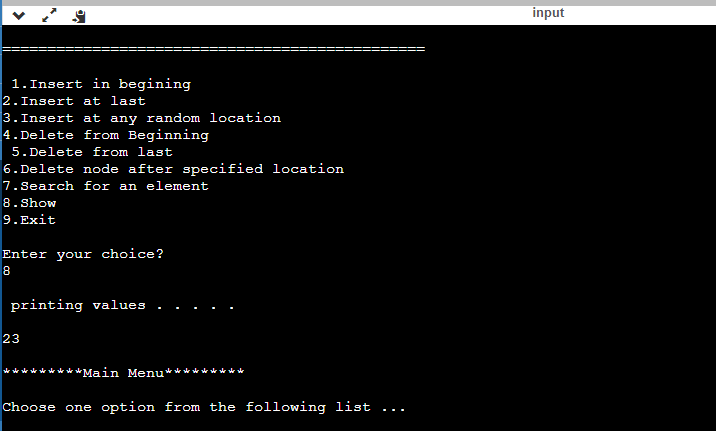
**}**

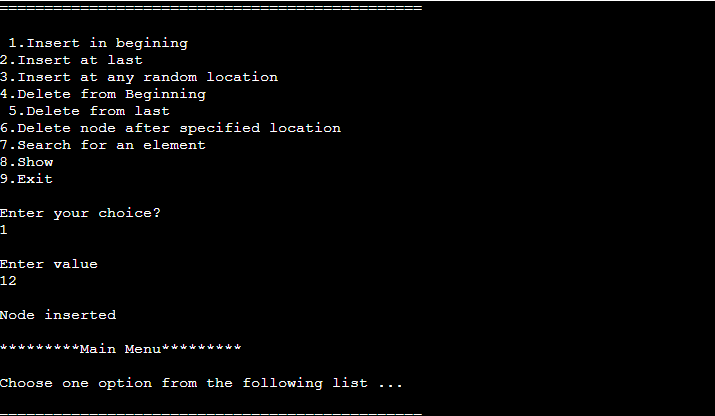
**}**

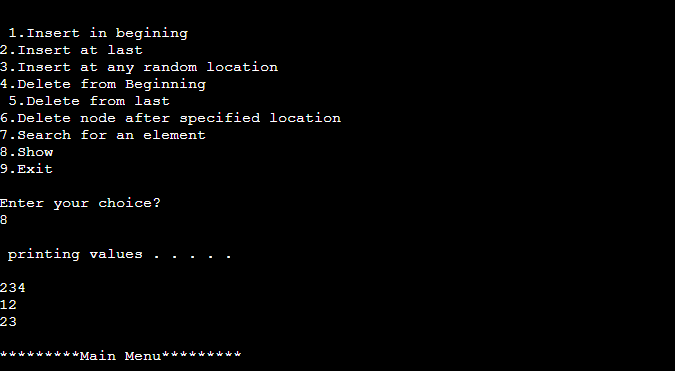
**}**

**Output -:**

****

****

****

****