DS ASSIGNMENT-1

(**Roll no : 160120733167**)

12)

**AIM:** Consider a singly linked list having n nodes. The data items d1,d2,….dn are stored in these n nodes. Let X be a pointer to the jth node(1<=j<=n) in which dj is stored. A new data item d stored in node with address Y is t be inserted. Write a program to insert d into the list to obtain a list having items d1,d2,… dj-1, d1,d2,……dn in the order without using the header.

**CODE:**

**#include<stdio.h>**

**#include<stdlib.h>**

**struct node**

**{**

**int data;**

**struct node\*next;**

**}\*head,\*first,\*X,\*third,\*fourth;**

**void traverse(struct node\* ptr){**

**while(ptr!=NULL){**

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

**ptr=ptr->next;**

**}**

**}**

**void Insert(struct node\*X){**

**int temp\_data,temp\_address;**

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

**Y->data=25; //d is 25 and Y is the node to be inserted before X.**

**Y->next=X->next;**

**X->next=Y;**

**//code to swap the data**

**temp\_data=X->data;**

**X->data=Y->data;**

**Y->data=temp\_data;**

**//code to swap address**

**temp\_address=X;**

**X=Y;**

**Y=temp\_address;**

**}**

**int main(){**

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

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

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

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

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

**head->data=10;**

**head->next=first;**

**first->data=20;**

**first->next=X;**

**X->data=30; //dj=30**

**X->next=third;**

**third->data=40;**

**third->next=fourth;**

**fourth->data=50;**

**fourth->next=NULL;**

**printf("list when Y is not inserted\n");**

**traverse(head);**

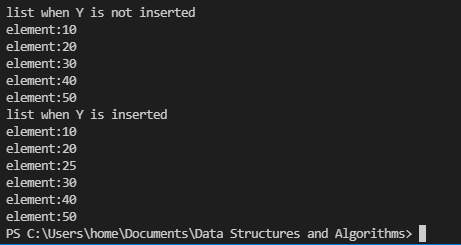
**printf("list when Y is inserted\n");**

**Insert(X);**

**traverse(head);**

**}**

**OUTPUT:**

****