Assignment 3-Data Structures and Algorithms

**Question 1:**

**Write a function “insert\_any()” for inserting a node at any given position of the linked list. Assume position starts at 0.**

**Answer:-**

**#include<stdio.h>**

**#include<cstdlib>**

**struct node {**

**int data;**

**struct node \*link;**

**};**

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

**void insert\_any();**

**int main() {**

**int choice;**

**int cont = 1;**

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

**header->data = NULL;**

**header->link = NULL;**

**insert\_any();**

**ptr = header;**

**while(ptr->link != NULL) {**

**ptr = ptr->link;**

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

**}**

**return 0;**

**}**

**//Function to insert a node at any position after a particular node.**

**void insert\_any()**

**{**

**int data\_value, key;**

**printf("\nEnter data of the node: ");**

**scanf("%d", &data\_value);**

**printf("\nEnter data of the node after which new node is to be inserted: ");**

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

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

**//Traverse till key is found or end of the linked list is reached.**

**ptr = header;**

**while(ptr->link != NULL && ptr->data != key) {**

**ptr = ptr->link;**

**}**

**if(ptr->data == key) {**

**temp->data = data\_value;**

**temp->link = ptr->link;**

**ptr->link = temp;**

**}**

**else {**

**printf("\nValue %d not found\n",key);**

**}**

**}**

**Question 2:**

**Write a function “delete\_beg()” for deleting a node from the beginning of the linked list.**

**Answer:**

**#include<stdio.h>**

**#include<stdlib.h>**

**struct node{**

**int data;**

**struct node \*next;**

**}\*start=NULL,\*q,\*t;**

**int main(){**

**int ch;**

**void insert\_beg();**

**void display();**

**void delete\_beg();**

**while(1) {**

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

**printf("\n1.Insert\n2.Display\n3.Delete\n4.Exit\n\n");**

**printf("Enter your choice(1-4):");**

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

**switch(ch) {**

**case 1: insert\_beg();**

**break;**

**case 2: display();**

**break;**

**case 3: delete\_beg();**

**break;**

**case 4: exit(0);**

**default: printf("Wrong Choice!!");**

**}**

**}**

**return 0;**

**}**

**void insert\_beg(){**

**int num;**

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

**printf("Enter data:");**

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

**t->data=num;**

**if(start==NULL) {**

**t->next=NULL;**

**start=t;**

**}**

**else{**

**t->next=start;**

**start=t;**

**}**

**}**

**void display(){**

**if(start==NULL) {**

**printf("List is empty!!");**

**}**

**else {**

**q=start;**

**printf("The linked list is:\n");**

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

**printf("%d->",q->data);**

**q=q->next;**

**}**

**}**

**}**

**void delete\_beg(){**

**if(start==NULL) {**

**printf("The list is empty!!");**

**}**

**else{**

**q=start;**

**start=start->next;**

**printf("Deleted element is %d",q->data);**

**free(q);**

**}**

**}**

**Question 3:**

**Write a function “delete\_end()” for deleting a node from the end of the linked list.**

**Answer:**

**#include<stdio.h>**

**#include<stdlib.h>**

**struct node{**

**int data;**

**struct node \*next;**

**}\*start=NULL,\*q,\*t;**

**int main(){**

**int ch;**

**void insert\_beg();**

**void display();**

**void delete\_beg();**

**void delete\_end();**

**int delete\_pos();**

**while(1) {**

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

**printf("\n1.Insert\n2.Display\n3.Delete\n4.Exit\n\n");**

**printf("Enter your choice(1-4):");**

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

**switch(ch) {**

**case 1: insert\_beg();**

**break;**

**case 2: display();**

**break;**

**case 3: printf("\n---- Delete Menu ----");**

**printf("\n1.Delete from beginning\n2.Delete from end\n3.Delete position\n4.Exit");**

**printf("\n\nEnter your choice(1-4):");**

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

**switch(ch)**

**{**

**case 1: delete\_beg();**

**break;**

**case 2: delete\_end();**

**break;**

**case 3: delete\_pos();**

**break;**

**case 4: exit(0);**

**default: printf("Wrong Choice!!");**

**}**

**break;**

**case 6: exit(0);**

**default: printf("Wrong Choice!!");**

**}**

**}**

**return 0;**

**}**

**void insert\_beg(){**

**int num;**

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

**printf("Enter data:");**

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

**t->data=num;**

**if(start==NULL) {**

**t->next=NULL;**

**start=t;**

**}**

**else{**

**t->next=start;**

**start=t;**

**}**

**}**

**void display(){**

**if(start==NULL) {**

**printf("List is empty!!");**

**}**

**else {**

**q=start;**

**printf("The linked list is:\n");**

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

**printf("%d->",q->data);**

**q=q->next;**

**}**

**}**

**}**

**void delete\_beg()**

**{**

**if(start==NULL)**

**{**

**printf("The list is empty!!");**

**}**

**else**

**{**

**q=start;**

**start=start->next;**

**printf("Deleted element is %d",q->data);**

**free(q);**

**}**

**}**

**void delete\_end()**

**{**

**if(start==NULL)**

**{**

**printf("The list is empty!!");**

**}**

**else**

**{**

**q=start;**

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

**q=q->next;**

**t=q->next;**

**q->next=NULL;**

**printf("Deleted element is %d",t->data);**

**free(t);**

**}**

**}**

**int delete\_pos()**

**{**

**int pos,i;**

**if(start==NULL)**

**{**

**printf("List is empty!!");**

**return 0;**

**}**

**printf("Enter position to delete:");**

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

**q=start;**

**for(i=1;i<pos-1;i++)**

**{**

**if(q->next==NULL)**

**{**

**printf("There are less elements!!");**

**return 0;**

**}**

**q=q->next;**

**}**

**t=q->next;**

**q->next=t->next;**

**printf("Deleted element is %d",t->data);**

**free(t);**

**return 0;**

**}**