Assignment 7

Q1)

Code:

//Just dont mind, I have combined all the cases asked in the question in one. Sorry.

//Basic Linked List implementation

#include <stdio.h>

#include <stdlib.h>

struct Node

{

int data;

struct Node \*next;//stores the addrerss of the node

};

struct Node \*head=NULL;

struct Node \*tail=NULL;

void takeinput();//inseting the node in the linked list

void insert();//inserting ith node in the linked list

void delete();//deleting ith node in the linked list

void length();//finding the length of the linked list

void print();//printing the linked list

int main()

{

int data,pos,delpos;

int choice;

while(1)

{

printf("\n Main Menu");

printf("\n 1.Create");

printf("\n 2.Insert");

printf("\n 3.Delete");

printf("\n 4.Display");

printf("\n 5.Length");

printf("\n 6.Exit");

printf("\n Enter your choice:");

scanf("%d",&choice);

switch(choice)

{

case 1:printf("enter the data in the linked list before it implementing it");

takeinput();

print();

break;

case 2:insert();

print();

break;

case 3:delete();

print();

break;

case 4:print();

break;

case 5:length();

break;

case 6:exit(0);

break;

default: printf("invalid entry");

}

}

return 0;

}

void takeinput()

{

struct Node \*n=(struct Node\*) malloc(sizeof(struct Node));

int data;

printf("\n enter the data");

scanf("%d",&n->data);

n->next=NULL;

if(head==NULL)

{

head=n;

tail=n;

}

else

{

//tail=head;

tail->next=n;

tail=tail->next;

}

}

void insert()

{

int pos;

struct Node \*n=(struct Node\*) malloc(sizeof(struct Node));

printf("enter the position in which the data is to be inserted");

scanf("%d",&pos);

printf("enter the data");

scanf("%d",&n->data);

n->next=NULL;

struct Node \*temp;

int ctr=0;

if(pos==0)

{

n->next=head;

head=n;

}

temp=head;

while(temp!=NULL && ctr<pos-1)

{

temp=temp->next;

ctr++;

}

if(temp!=NULL)

{

struct Node \*a=temp->next;

temp->next=n;

n->next=a;

}

}

void delete()

{

int delpos;

if(head==NULL)

{

printf("pls enter the element in the linked list");

return;

}

printf("enter the position in which the data is to be deleted");

scanf("%d",&delpos);

int ctr=0;

struct Node \*temp=head;

if(delpos==0)

{

head=temp->next;

temp->next=NULL;

free(temp);

return;

}

while(temp->next!=NULL && ctr<delpos-1)

{

temp=temp->next;

ctr++;

}

if(temp->next!=NULL)

{

struct Node \*a=temp->next;

struct Node \*b=a->next;

temp->next=b;

a->next=NULL;

free(a);

}

return;

}

void print()

{

struct Node \*temp=head;

if(temp==NULL)

{

printf("\n pls enter any enement in the linked list");

return;

}

while(temp!=NULL)

{

printf("%d ",temp->data);

temp=temp->next;

}

printf("\n");

return;

}

void length()

{

struct Node \*temp=head;

int i=0;

while(temp!=NULL)

{

i++;

temp=temp->next;

}

printf("\n the length of the linked list is %d",i);

}

Output:

Main Menu

 1.Create

 2.Insert

 3.Delete

 4.Display

 5.Length

 6.Exit

 Enter your choice:1

enter the data in the linked list before it implementing it

 enter the data1

1

 Main Menu

 1.Create

 2.Insert

 3.Delete

 4.Display

 5.Length

 6.Exit

 Enter your choice:2

enter the position in which the data is to be inserted1

enter the data2

1 2

 Main Menu

 1.Create

 2.Insert

 3.Delete

 4.Display

 5.Length

 6.Exit

 Enter your choice:2

enter the position in which the data is to be inserted0

enter the data3

3 1 2

 Main Menu

 1.Create

 2.Insert

 3.Delete

 4.Display

 5.Length

 6.Exit

 Enter your choice:2

enter the position in which the data is to be inserted2

enter the data4

3 1 4 2

 Main Menu

 1.Create

 2.Insert

 3.Delete

 4.Display

 5.Length

 6.Exit

 Enter your choice:3

enter the position in which the data is to be deleted1

3 4 2

 Main Menu

 1.Create

 2.Insert

 3.Delete

 4.Display

 5.Length

 6.Exit

 Enter your choice:3

enter the position in which the data is to be deleted0

4 2

 Main Menu

 1.Create

 2.Insert

 3.Delete

 4.Display

 5.Length

 6.Exit

 Enter your choice:4

4 2

 Main Menu

 1.Create

 2.Insert

 3.Delete

 4.Display

 5.Length

 6.Exit

 Enter your choice:5

 the length of the linked list is 2

 Main Menu

 1.Create

 2.Insert

 3.Delete

 4.Display

 5.Length

 6.Exit

 Enter your choice:6

**...Program finished with exit code 0**

**Press ENTER to exit console.**

Q2)

Code:

#include <stdio.h>

#include <stdlib.h>

struct Node

{

int data;

struct Node \*next;//stores the addrerss of the node

};

struct Node \*head=NULL;

struct Node \*tail=NULL;

void takeinput();//inseting the node in the linked list

void insert();//inserting ith node in the linked list

void print();//printing the linked list

int search(int ele);//searching the element in the linked list

void delete1();//deleting node data of the linked list

void delete2();//deleting next node of the linked list

int main()

{

int data,pos,delpos;

int choice;

while(1)

{

printf("\n Main Menu");

printf("\n 1.Create");

printf("\n 2.Insert");

printf("\n 3.Display");

printf("\n 4.Delete node data");

printf("\n 5.Delete next node");

printf("\n 6.Exit");

printf("\n Enter your choice:");

scanf("%d",&choice);

switch(choice)

{

case 1:printf("enter the data in the linked list before it implementing it");

takeinput();

print();

break;

case 2:insert();

print();

break;

case 3:print();

break;

case 4:delete1();

print();

break;

case 5:delete2();

print();

break;

case 6:exit(0);

break;

default: printf("invalid entry");

}

}

return 0;

}

void takeinput()

{

struct Node \*n=(struct Node\*) malloc(sizeof(struct Node));

int data;

printf("\n enter the data");

scanf("%d",&n->data);

n->next=NULL;

if(head==NULL)

{

head=n;

tail=n;

}

else

{

//tail=head;

tail->next=n;

tail=tail->next;

}

}

void insert()

{

int pos;

struct Node \*n=(struct Node\*) malloc(sizeof(struct Node));

printf("enter the position in which the data is to be inserted");

scanf("%d",&pos);

printf("enter the data");

scanf("%d",&n->data);

n->next=NULL;

struct Node \*temp;

int ctr=0;

if(pos==0)

{

n->next=head;

head=n;

}

temp=head;

while(temp!=NULL && ctr<pos-1)

{

temp=temp->next;

ctr++;

}

if(temp!=NULL)

{

struct Node \*a=temp->next;

temp->next=n;

n->next=a;

}

}

/\*void delete()

{

int delpos;

if(head==NULL)

{

printf("pls enter the element in the linked list");

return;

}

printf("enter the position in which the data is to be deleted");

scanf("%d",&delpos);

int ctr=0;

struct Node \*temp=head;

if(delpos==0)

{

head=temp->next;

temp->next=NULL;

free(temp);

return;

}

while(temp->next!=NULL && ctr<delpos-1)

{

temp=temp->next;

ctr++;

}

if(temp->next!=NULL)

{

struct Node \*a=temp->next;

struct Node \*b=a->next;

temp->next=b;

a->next=NULL;

free(a);

}

return;

}\*/

void print()

{

struct Node \*temp=head;

if(temp==NULL)

{

printf("\n pls enter any enement in the linked list");

return;

}

while(temp!=NULL)

{

printf("%d ",temp->data);

temp=temp->next;

}

printf("\n");

return;

}

int search(int ele)

{

int ctr=0;

struct Node \*temp=head;

while(temp!=NULL)

{

if(temp->data==ele)

{

return ctr;

}

ctr++;

temp=temp->next;

}

return -1;

}

void delete1()

{

int ele;

if(head==NULL)

{

printf("pls enter the element in the linked list");

return;

}

printf("enter the element to be deleted");

scanf("%d",&ele);

int ctr=0;

struct Node \*temp=head;

int x=search(ele);

if(x==-1)

{

printf("No such element is found");

return;

}

else if(x==0)

{

head=temp->next;

temp->next=NULL;

free(temp);

return;

}

else

{

int ctr=0;

while(temp->next!=NULL && ctr<x-1)

{

temp=temp->next;

ctr++;

}

if(temp->next!=NULL)

{

struct Node \*a=temp->next;

struct Node \*b=a->next;

temp->next=b;

a->next=NULL;

free(a);

}

}

return;

}

void delete2()

{

int ele;

if(head==NULL)

{

printf("pls enter the element in the linked list");

return;

}

printf("enter the element for which next element is to be deleted");

scanf("%d",&ele);

int ctr=0;

struct Node \*temp=head;

int x=search(ele)+1;

if(x==-1)

{

printf("No such element is found");

return;

}

else if(x==0)

{

head=temp->next;

temp->next=NULL;

free(temp);

return;

}

else

{

int ctr=0;

while(temp->next!=NULL && ctr<x-1)

{

temp=temp->next;

ctr++;

}

if(temp->next!=NULL)

{

struct Node \*a=temp->next;

struct Node \*b=a->next;

temp->next=b;

a->next=NULL;

free(a);

}

}

return;

}

Output:

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Delete node data

 5.Delete next node

 6.Exit

 Enter your choice:1

enter the data in the linked list before it implementing it

 enter the data4

4

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Delete node data

 5.Delete next node

 6.Exit

 Enter your choice:2

enter the position in which the data is to be inserted3

enter the data4

4

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Delete node data

 5.Delete next node

 6.Exit

 Enter your choice:2

enter the position in which the data is to be inserted0

enter the data3

3 4

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Delete node data

 5.Delete next node

 6.Exit

 Enter your choice:2

enter the position in which the data is to be inserted1

enter the data9

3 9 4

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Delete node data

 5.Delete next node

 6.Exit

 Enter your choice:2

enter the position in which the data is to be inserted3

enter the data3

3 9 4 3

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Delete node data

 5.Delete next node

 6.Exit

 Enter your choice:4

enter the element to be deleted9

3 4 3

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Delete node data

 5.Delete next node

 6.Exit

 Enter your choice:5

enter the element for which next element is to be deleted4

3 4

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Delete node data

 5.Delete next node

 6.Exit

 Enter your choice:6

**...Program finished with exit code 0**

**Press ENTER to exit console.**

Q3)

Code:

#include <stdio.h>

#include <stdlib.h>

struct Node

{

int data;

struct Node \*next;//stores the addrerss of the node

};

struct Node \*head=NULL;

struct Node \*tail=NULL;

void takeinput();//inseting the node in the linked list

void insert();//inserting ith node in the linked list

void print();//printing the linked list

struct Node \*append();//appending the first node to the last

int length();

int main()

{

int data,pos,delpos;

int choice;

while(1)

{

printf("\n Main Menu");

printf("\n 1.Create");

printf("\n 2.Insert");

printf("\n 3.Display");

printf("\n 4.Append first to last");

//printf("\n 5.Delete next node");

printf("\n 5.Exit");

printf("\n Enter your choice:");

scanf("%d",&choice);

switch(choice)

{

case 1:printf("enter the data in the linked list before it implementing it");

takeinput();

print();

break;

case 2:insert();

print();

break;

case 3:print();

break;

case 4:head=append();

print();

break;

case 5:exit(0);

break;

default: printf("invalid entry");

}

}

return 0;

}

void takeinput()

{

struct Node \*n=(struct Node\*) malloc(sizeof(struct Node));

int data;

printf("\n enter the data");

scanf("%d",&n->data);

n->next=NULL;

if(head==NULL)

{

head=n;

tail=n;

}

else

{

//tail=head;

tail->next=n;

tail=tail->next;

}

}

void insert()

{

int pos;

struct Node \*n=(struct Node\*) malloc(sizeof(struct Node));

printf("enter the position in which the data is to be inserted");

scanf("%d",&pos);

printf("enter the data");

scanf("%d",&n->data);

n->next=NULL;

struct Node \*temp;

int ctr=0;

if(pos==0)

{

n->next=head;

head=n;

}

temp=head;

while(temp!=NULL && ctr<pos-1)

{

temp=temp->next;

ctr++;

}

if(temp!=NULL)

{

struct Node \*a=temp->next;

temp->next=n;

n->next=a;

}

}

int length()

{

//Write your code here

int i=0;

struct Node \*temp=head;

while(temp!= NULL)

{

i++;

temp=temp->next;

}

return i;

}

/\*int length()

{

if(head==NULL)

{

return 0;

}

struct Node \*temp=head;

int a=length(temp->next);

return a+1;

}\*/

/\*void delete()

{

int delpos;

if(head==NULL)

{

printf("pls enter the element in the linked list");

return;

}

printf("enter the position in which the data is to be deleted");

scanf("%d",&delpos);

int ctr=0;

struct Node \*temp=head;

if(delpos==0)

{

head=temp->next;

temp->next=NULL;

free(temp);

return;

}

while(temp->next!=NULL && ctr<delpos-1)

{

temp=temp->next;

ctr++;

}

if(temp->next!=NULL)

{

struct Node \*a=temp->next;

struct Node \*b=a->next;

temp->next=b;

a->next=NULL;

free(a);

}

return;

}\*/

void print()

{

struct Node \*temp=head;

if(temp==NULL)

{

printf("\n pls enter any enement in the linked list");

return;

}

while(temp!=NULL)

{

printf("%d ",temp->data);

temp=temp->next;

}

printf("\n");

return;

}

/\*int search(int ele)

{

int ctr=0;

struct Node \*temp=head;

while(temp!=NULL)

{

if(temp->data==ele)

{

return ctr;

}

ctr++;

temp=temp->next;

}

return -1;

}

void delete1()

{

int ele;

if(head==NULL)

{

printf("pls enter the element in the linked list");

return;

}

printf("enter the element to be deleted");

scanf("%d",&ele);

int ctr=0;

struct Node \*temp=head;

int x=search(ele);

if(x==-1)

{

printf("No such element is found");

return;

}

else if(x==0)

{

head=temp->next;

temp->next=NULL;

free(temp);

return;

}

else

{

int ctr=0;

while(temp->next!=NULL && ctr<x-1)

{

temp=temp->next;

ctr++;

}

if(temp->next!=NULL)

{

struct Node \*a=temp->next;

struct Node \*b=a->next;

temp->next=b;

a->next=NULL;

free(a);

}

}

return;

}

void delete2()

{

int ele;

if(head==NULL)

{

printf("pls enter the element in the linked list");

return;

}

printf("enter the element for which next element is to be deleted");

scanf("%d",&ele);

int ctr=0;

struct Node \*temp=head;

int x=search(ele)+1;

if(x==-1)

{

printf("No such element is found");

return;

}

else if(x==0)

{

head=temp->next;

temp->next=NULL;

free(temp);

return;

}

else

{

int ctr=0;

while(temp->next!=NULL && ctr<x-1)

{

temp=temp->next;

ctr++;

}

if(temp->next!=NULL)

{

struct Node \*a=temp->next;

struct Node \*b=a->next;

temp->next=b;

a->next=NULL;

free(a);

}

}

return;

}\*/

struct Node \*append()

{

//struct Node \*h=head;

int n;

printf("enter the no of nodes to be appended");

scanf("%d",&n);

struct Node \*temp=head;

int len=length();

int ctr=len-n;

if(n!=0 && n<len && len!=0)

{

int i=1;

while(i<ctr)

{

temp=temp->next;

i++;

}

struct Node \*h2=temp->next;

temp->next=NULL;

int j=1;

struct Node \*t2=h2;

while(j<n)

{

t2=t2->next;

j++;

}

t2->next=head;

return h2;

}

else

{

return head;

}

}

Output:

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Append first to last

 5.Exit

 Enter your choice:1

enter the data in the linked list before it implementing it

 enter the data6

6

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Append first to last

 5.Exit

 Enter your choice:1

enter the data in the linked list before it implementing it

 enter the data7

6 7

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Append first to last

 5.Exit

 Enter your choice:1

enter the data in the linked list before it implementing it

 enter the data4

6 7 4

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Append first to last

 5.Exit

 Enter your choice:1

enter the data in the linked list before it implementing it

 enter the data8

6 7 4 8

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Append first to last

 5.Exit

 Enter your choice:4

enter the no of nodes to be appended2

4 8 6 7

 Main Menu

 1.Create

 2.Insert

 3.Display

 4.Append first to last

 5.Exit

 Enter your choice:5

**...Program finished with exit code 0**

**Press ENTER to exit console.**