ASSIGNMENT-5

Q1

#include <iostream>

#include<stdlib.h>

using namespace std;

struct node {

int data;

struct node \*next;

};

void traversal(struct node\*ptr){

while(ptr!=NULL){

cout<<" "<<ptr->data;

ptr=ptr->next;

}

}

struct node\* insertbeg(struct node\*head,int val){

struct node\*ptr;

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

ptr->next=head;

ptr->data=val;

return ptr;

}

struct node\* insertend(struct node\*head,int val){

struct node\*ptr;

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

struct node\*p=head;

while(p->next!=NULL){

p=p->next;

}

ptr->data=val;

p->next=ptr;

ptr->next=NULL;

return head;

}

struct node\* insertafterval(struct node\*head,int val1,int val){

struct node\*ptr;

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

struct node\*p=head;

while(p!=NULL&&p->data!=val1){

p=p->next;

}

if(p==NULL){

cout<<"no value found in the nodes";

}

else{

ptr->data=val;

ptr->next=p->next;

p->next=ptr;

}

return head;

}

struct node\* insertafterpos(struct node\*head,int index ,int val){

struct node\*ptr;

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

if(index<0){

cout<<"invalid index";

}

else if(index==0){

ptr->data=val;

ptr->next=head;

return ptr;

}

else{

struct node\*p=head;

int i=0;

while(p!=NULL&& i!=index-1){

p=p->next;

i++;

}

if(p==NULL){

cout<<"position out of range";

}

else{

ptr->data=val;

ptr->next=p->next;

p->next=ptr;

return head;

}

}

}

struct node\* deletebeg(struct node\*head){

struct node\*ptr=head;

head=head->next;

free(ptr);

return head;

}

struct node\* deleteend(struct node\*head){

struct node\*ptr=head;

struct node\*p=head->next;

while(p->next!=NULL){

p=p->next;

ptr=ptr->next;

}

ptr->next=NULL;

free(p);

return head;

}

struct node\* deletenode(struct node\*head, struct node\*prevnode ){

struct node\*ptr=head;

struct node\*p=head->next;

while(p!=prevnode){

p=p->next;

ptr=ptr->next;

}

ptr->next=prevnode->next;

free(p);

return head;

}

void search(struct node\*head, struct node\*prevnode ){

int i=0;

struct node\*ptr=head;

while(ptr!=prevnode){

ptr=ptr->next;

i++;

}

cout<<"position of node from head is :"<<i;

}

int main()

{

struct node\*head;

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

struct node\*second;

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

struct node\*third;

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

head->data=7;

head->next=second;

second->data=11;

second->next=third;

third->data=22;

third->next=NULL;

search(head,second);

return 0;

}

Q2

#include <iostream>

#include<stdlib.h>

using namespace std;

struct node {

int data;

struct node \*next;

};

void traversal(struct node\*ptr){

while(ptr!=NULL){

cout<<" "<<ptr->data;

ptr=ptr->next;

}

}

void count (struct node\*head,int key){

struct node\*ptr=head;

int c=0;

while(ptr!=NULL){

if(ptr->data==key){

c++;

}

ptr=ptr->next;

}

cout<<"element is present:"<<c;

}

struct node\* del(struct node\*head,int key){

while(head!=NULL&& head->data==key){

struct node\*ptr=head;

head=head->next;

free(ptr);

}

struct node\*p=head;

while(p!=NULL&& p->next!=NULL){

if(p->next->data==key){

struct node\*temp=p->next;

p->next=temp->next;

free(temp);

}

else{

p=p->next;

}

}

return head;

}

int main()

{

struct node\*head;

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

struct node\*second;

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

struct node\*third;

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

struct node\*fourth;

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

struct node\*fifth;

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

struct node\*six;

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

struct node\*sev;

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

head->data=1;

head->next=second;

second->data=2;

second->next=third;

third->data=1;

third->next=fourth;

fourth->data=2;

fourth->next=fifth;

fifth->data=1;

fifth->next=six;

six->data=3;

six->next=sev;

sev->data=1;

sev->next=NULL;

count(head,1);

head=del(head,1);

cout<<endl;

traversal(head);

return 0;

}

Q3

#include <iostream>

#include<stdlib.h>

using namespace std;

struct node {

int data;

struct node \*next;

};

void traversal(struct node\*ptr){

while(ptr!=NULL){

cout<<" "<<ptr->data;

ptr=ptr->next;

}

}

void middle(struct node\*head){

struct node\*ptr=head;

struct node\*p=head;

int i=0;

while(ptr->next!=NULL){

ptr=ptr->next;

i++;

}

int b=(i)/2;

cout<<"middle element is on position:"<<(i/2)+1<<endl;

int a=0;

while(a<b){

p=p->next;

a++;

}

cout<<"middle number:"<<p->data<<endl;

}

int main()

{

struct node\*head;

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

struct node\*second;

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

struct node\*third;

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

struct node\*fourth;

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

struct node\*fifth;

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

head->data=1;

head->next=second;

second->data=2;

second->next=third;

third->data=3;

third->next=fourth;

fourth->data=4;

fourth->next=fifth;

fifth->data=5;

fifth->next=NULL;

middle(head);

return 0;

}

OR

#include <iostream>

#include<stdlib.h>

using namespace std;

struct node {

int data;

struct node \*next;

};

void traversal(struct node\*ptr){

while(ptr!=NULL){

cout<<" "<<ptr->data;

ptr=ptr->next;

}

}

void middle(struct node\*head){

struct node\*slow=head;

struct node\*fast=head;

while(fast!=NULL&& fast->next!=NULL){

slow=slow->next;

fast=fast->next->next;

}

cout<<"middle element:"<<slow->data;

}

int main()

{

struct node\*head;

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

struct node\*second;

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

struct node\*third;

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

struct node\*fourth;

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

struct node\*fifth;

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

head->data=1;

head->next=second;

second->data=2;

second->next=third;

third->data=3;

third->next=fourth;

fourth->data=4;

fourth->next=fifth;

fifth->data=5;

fifth->next=NULL;

middle(head);

return 0;

}

Q5

#include <iostream>

#include<stdlib.h>

using namespace std;

struct node {

int data;

struct node \*next;

};

void traversal(struct node\*ptr){

while(ptr!=NULL){

cout<<" "<<ptr->data;

ptr=ptr->next;

}

}

void count (struct node\*head,int key){

struct node\*ptr=head;

int c=0;

while(ptr!=NULL){

if(ptr->data==key){

c++;

}

ptr=ptr->next;

}

cout<<"element is present:"<<c;

}

struct node\* rev(struct node\*head){

struct node\*prev=NULL;

struct node\*curr=head;

struct node\*next=NULL;

while(curr!=NULL){

next=curr->next;

curr->next=prev;

prev=curr;

curr=next;

}

return prev;

}

int main()

{

struct node\*head;

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

struct node\*second;

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

struct node\*third;

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

struct node\*fourth;

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

struct node\*fifth;

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

struct node\*six;

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

struct node\*sev;

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

head->data=1;

head->next=second;

second->data=2;

second->next=third;

third->data=1;

third->next=fourth;

fourth->data=2;

fourth->next=fifth;

fifth->data=1;

fifth->next=six;

six->data=3;

six->next=sev;

sev->data=1;

sev->next=NULL;

head=rev(head);

cout<<endl;

traversal(head);

return 0;

}