Question 1 (A)

#include<iostream>

using namespace std;

template<typename t>

class node{

public:

t data;

node \* next;

node(t d)

{

this->data=d;

this->next=NULL;

}

void create(node <t>\* &head)

{

t v;

cout<<"enter the value : ";

cin>>v;

node<t> \* n=new node<t>(v);

if(head==NULL)

{

head=n;

return;

}

node<t> \* temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=n;

}

int search(node<t> \* &head)

{

t v;

cout<<"enter the key : ";

cin>>v;

int count =0;

node<t> \* temp=head;

while(temp!=NULL)

{

if(temp->data==v)return count;

count++;

temp=temp->next;

}

return -1;

}

void print(node <t>\* &head)

{

node <t>\* temp=head;

while(temp!=NULL)

{

cout<<temp->data<<" -> ";

temp=temp->next;

}

cout<<"NULL\n";

}

};

int main()

{

node <int> \* head=NULL;

int n;

cout<<"enter the length of link list : ";

cin>>n;

for(int i=0;i<n;i++)

{

head->create(head);

}

// head->print(head);

cout<<"\n";

int t=1;

while(t)

{

cout<<"enter 1 to continue or 0 to exit search";

cin>>t;

if(t)

{

int pos=head->search(head);

if(pos==-1) cout<<"element is not found \n";

else cout<<"the position is : "<<pos+1<<" \n";

}

}

}

Question 1(b)

#include <iostream>

#include <bits/stdc++.h>

using namespace std;

template <typename t>

class node

{

public:

t data;

node \*next;

node(t d)

{

this->data = d;

this->next = NULL;

}

void create(node<t> \*&head)

{

t v;

cout << "enter the value : ";

cin >> v;

node<t> \*n = new node<t>(v);

if (head == NULL)

{

head = n;

return;

}

node<t> \*temp = head;

while (temp->next != NULL)

{

temp = temp->next;

}

temp->next = n;

}

void print(node<t> \*&head)

{

node<t> \*temp = head;

while (temp != NULL)

{

cout << temp->data << " -> ";

temp = temp->next;

}

cout << "NULL\n";

}

void bubble\_Sort(node<t> \*&head)

{

// bubble sort

node<int> \*i = head;

while (i != NULL)

{

node<int> \*j = head;

while (j->next != NULL)

{

if (j->data > j->next->data)

{

node<int> \*temp = new node<int>(j->data);

j->data = j->next->data;

j->next->data = temp->data;

}

j = j->next;

}

i = i->next;

// head->print(head);

}

head->print(head);

}

void selection\_sort(node<t> \*&head)

{

node<int> \*i = head;

while (i != NULL)

{

node<int> \*j = i;

node<int> \*y = new node<int>(INT\_MAX);

while (j != NULL)

{

if (j->data < y->data)

{

y = j;

}

j = j->next;

}

node<int> \*temp = new node<int>(y->data);

y->data = i->data;

i->data = temp->data;

delete temp;

i = i->next;

}

head->print(head);

}

void isort(node<t> \* & i,node<t> \* & j, t & temp,node <t> \* & pos)

{

if(j==i)

{

return;

}

isort(i,j->next,temp,pos);

if(j->data>temp )

{

j->next->data=j->data;

}

else if(pos==NULL)

{

pos=j->next;

}

}

void insertion(node<t> \* & head)

{

node<t> \* i=head->next;

while(i!=NULL)

{

t temp=i->data;

node<t> \* j=head;

node<t> \* pos=NULL;

isort(i,j,temp,pos);

if(pos!=NULL) pos->data=temp;

else j->data=temp;

i=i->next;

}

}

};

int main()

{

node<int> \*head = NULL;

int n;

cout << "enter the length of link list : ";

cin >> n;

for (int i = 0; i < n; i++)

{

head->create(head);

}

//head->bubble\_Sort(head);

// head->selection\_sort(head);

head->insertion(head);

head->print(head);

}

Question 1(c)

#include<iostream>

using namespace std;

template<typename t>

class node{

public:

t data;

node \* next;

node(t d)

{

this->data=d;

this->next=NULL;

}

void create(node <t>\* &head)

{

t v;

cout<<"enter the value : ";

cin>>v;

node<t> \* n=new node<t>(v);

if(head==NULL)

{

head=n;

return;

}

node<t> \* temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=n;

}

void delete\_node(node<t> \* &head)

{

t v;

cout<<"enter the value to be deleted : ";

cin>>v;

node<t> \* temp=head;

if(temp->data==v)

{

if(temp->next!=NULL){

head=temp->next;}

else head=NULL;

temp->next=NULL;

delete temp;

return;

}

while( temp->next!=NULL && temp->next->data!=v)

{

temp=temp->next;

}

if(temp->next==NULL || temp==NULL) {cout<<"not found \n"; return;}

node<t> \* to\_del=temp->next;

temp->next=to\_del->next;

to\_del->next=NULL;

delete to\_del;

}

void print(node <t>\* &head)

{

node <t>\* temp=head;

while(temp!=NULL)

{

cout<<temp->data<<" -> ";

temp=temp->next;

}

cout<<"NULL\n";

}

};

int main()

{

node <int> \* head=NULL;

int n;

cout<<"enter the length of link list : ";

cin>>n;

for(int i=0;i<n;i++)

{

head->create(head);

}

// head->print(head);

cout<<"\n";

int t=1;

while(t)

{

cout<<"enter 1 to continue or 0 to exit delete ";

cin>>t;

if(t)

{

head->delete\_node(head);

cout<<"NEW LINK LIST : ";

head->print(head);

}

}

}

Question 1(d,e)

#include<iostream>

using namespace std;

template<typename t>

class node{

public:

t data;

node \* next;

node(t d)

{

this->data=d;

this->next=NULL;

}

void create(node <t>\* &head)

{

t v;

cout<<"enter the value : ";

cin>>v;

node<t> \* n=new node<t>(v);

if(head==NULL)

{

head=n;

return;

}

node<t> \* temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=n;

}

void insert\_at\_head(node<t> \* &head)

{

t v;

cout<<"enter the value : ";

cin>>v;

node<t> \* n=new node<t>(v);

if(head==NULL)

{

head=n;

return;

}

n->next=head;

head=n;

}

void insert\_at\_any\_pos(node<t>\* &head)

{

int pos;

cout<<"enter the position : ";

cin>>pos;

if(pos==1)

{

insert\_at\_head(head);

return;

}

else{

t v;

cout<<"enter the value : ";

cin>>v;

node<t> \* n=new node<t>(v);

int count=1;

node<t> \* temp=head;

while(count<pos-1)

{

temp=temp->next;

count++;

}

n->next=temp->next;

temp->next=n;

}

}

void print(node <t>\* &head)

{

node <t>\* temp=head;

while(temp!=NULL)

{

cout<<temp->data<<" -> ";

temp=temp->next;

}

cout<<"NULL\n";

}

};

int main()

{

node <int> \* head=NULL;

int n;

cout<<"enter the length of link list : ";

cin>>n;

for(int i=0;i<n;i++)

{

head->create(head);

}

// head->print(head);

cout<<"\n";

int t=1;

while(t)

{

cout<<"enter 1 to continue or 0 to exit delete ";

cin>>t;

if(t)

{

head->insert\_at\_any\_pos(head);

cout<<"NEW LINK LIST : ";

head->print(head);

}

}

}

Question 1(f)

#include<iostream>

using namespace std;

template<typename t>

class node{

public:

t data;

node \* next;

node(t d)

{

this->data=d;

this->next=NULL;

}

void create(node <t>\* &head)

{

t v;

cout<<"enter the value : ";

cin>>v;

node<t> \* n=new node<t>(v);

if(head==NULL)

{

head=n;

return;

}

node<t> \* temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=n;

}

void update\_node(node<t> \* &head)

{

t v;

cout<<"enter the value to be updated : ";

cin>>v;

int new\_data;

cout<<"enter the update value : ";

cin>>new\_data;

node<t> \* temp=head;

if(temp->data==v)

{

temp->data=new\_data;

return;

}

while( temp->next!=NULL && temp->next->data!=v)

{

temp=temp->next;

}

if(temp->next==NULL || temp==NULL) {cout<<"not found \n"; return;}

temp->next->data=new\_data;

}

void print(node <t>\* &head)

{

node <t>\* temp=head;

while(temp!=NULL)

{

cout<<temp->data<<" -> ";

temp=temp->next;

}

cout<<"NULL\n";

}

};

int main()

{

node <int> \* head=NULL;

int n;

cout<<"enter the length of link list : ";

cin>>n;

for(int i=0;i<n;i++)

{

head->create(head);

}

// head->print(head);

cout<<"\n";

int t=1;

while(t)

{

cout<<"enter 1 to continue or 0 to exit delete ";

cin>>t;

if(t)

{

head->update\_node(head);

cout<<"NEW LINK LIST : ";

head->print(head);

}

}

}

Question 1(g,h)

#include<iostream>

#include<map>

#include<vector>

using namespace std;

template<typename t>

class node

{

public:

t data;

node \* next;

node(t d)

{

this -> data=d;

this -> next=NULL;

}

void create(node <t> \* &head)

{

int v;

cout<<"enter the value : ";

cin>>v;

node<t> \* n=new node(v);

if(head==NULL)

{

head=n;

return;

}

node<t> \* temp=head;

while(temp->next!=NULL){

temp=temp->next;

}

temp->next=n;

}

void delete\_node (node <t>\* & head,int pos)

{

if(pos==1)

{

node <t>\* temp=head;

head=head->next;

temp->next=NULL;

delete temp;

}

else

{

node <t>\* temp=head;

int curr=1;

while(curr<pos-1)

{

curr++;

temp=temp->next;

}

node<t>\* to\_del=temp->next;

temp->next=to\_del->next;

to\_del=NULL;

delete to\_del;

}

}

vector<int> sel (node<t>\*&head,t d)

{

node<t>\* temp=head;

vector<int> ans;

int count=1;

while(temp!=NULL)

{

if(temp->data==d) ans.push\_back( count);

count++;

temp=temp->next;

}

return ans;

}

void print(node<t> \* & head)

{

node<t>\* temp=head;

while(temp!=NULL)

{

cout<<temp->data<<" -> ";

temp=temp->next;

}

cout<<"NULL \n";

}

};

int main()

{

int t;

cout<<"enter the length of link list : ";

cin>>t;

node<int> \* head=NULL;

for(int i=0;i<t;i++)

{

head->create(head);

}

head->print(head);

map<int ,vector<int> >m;

node <int> \* temp=head;

while(temp)

{

if(m[temp->data].size()==0)

{

m[temp->data]=head->sel(head,temp->data);

}

}

for(auto i:m)

{

if(i.second.size()>2)

{

for(int j=1;j<i.second.size()-1;j++)

{

cout<<i.second[j]<<"\*";

head->delete\_node(head,i.second[j]);

}

}

}

head->print(head);

}

Question 1(i)

#include<iostream>

#include<stack>

using namespace std;

template<typename t>

class node{

public:

t data;

node \* next;

node(t d)

{

this->data=d;

this->next=NULL;

}

void create(node <t>\* &head)

{

t v;

cout<<"enter the value : ";

cin>>v;

node<t> \* n=new node<t>(v);

if(head==NULL)

{

head=n;

return;

}

node<t> \* temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=n;

}

void print(node <t>\* &head)

{

node <t>\* temp=head;

while(temp!=NULL)

{

cout<<temp->data<<" -> ";

temp=temp->next;

}

cout<<"NULL\n";

}

void reverse(node <t>\* &head)

{

stack<t> s;

node <t>\* temp=head;

while(temp!=NULL)

{

s.push(temp->data);

temp=temp->next;

}

while(!s.empty())

{

cout<<s.top()<<" -> ";

s.pop();

}

cout<<"NULL\n";

}

};

int main()

{

node <int> \* head=NULL;

int n;

cout<<"enter the length of link list : ";

cin>>n;

for(int i=0;i<n;i++)

{

head->create(head);

}

cout<<"original link list : ";

head->print(head);

cout<<"\n";

cout<<"reverse link list : ";

head->reverse(head);

}

Question 1(j)

#include<iostream>

#include<set>

using namespace std;

template<typename t>

class node{

public:

t data;

node \* next;

node(t d)

{

this->data=d;

this->next=NULL;

}

void create(node <t>\* &head)

{

t v;

cout<<"enter the value : ";

cin>>v;

node<t> \* n=new node<t>(v);

if(head==NULL)

{

head=n;

return;

}

node<t> \* temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=n;

}

void print(node <t>\* &head)

{

node <t>\* temp=head;

while(temp!=NULL)

{

cout<<temp->data<<" -> ";

temp=temp->next;

}

cout<<"NULL\n";

}

void second\_largest(node <t>\* &head)

{

set<t> s;

node <t>\* temp=head;

while(temp!=NULL)

{

s.insert(temp->data);

temp=temp->next;

}

auto it = s.end();

it--;

it--;

cout<<"the second largest element is : "<<\*it;

cout<<"\n";

}

};

int main()

{

node <int> \* head=NULL;

int n;

cout<<"enter the length of link list : ";

cin>>n;

for(int i=0;i<n;i++)

{

head->create(head);

}

cout<<"original link list : ";

head->print(head);

cout<<"\n";

head->second\_largest(head);

}

Question 2

#include <iostream>

using namespace std;

class book

{

public:

string name;

string a\_name;

int pb\_year;

int isbn;

int copy;

book \*next;

book(string n, string a, int py, int is, int co)

{

this->name = n;

this->a\_name = a;

this->pb\_year = py;

this->isbn = is;

this->copy = co;

this->next = NULL;

}

void add(book \*&head)

{

string n, a;

cout << "enter the name of book : ";

cin >> n;

cout << "enter the author name : ";

cin >> a;

int p, i, c;

cout << "enter the publication year : ";

cin >> p;

cout << "enter the ISBN number : ";

cin >> i;

cout << "enter the number of copies : ";

cin >> c;

book \*t = new book(n, a, p, i, c);

if (head == NULL)

{

head = t;

return;

}

book \*temp = head;

while (temp->next != NULL)

{

temp = temp->next;

}

temp->next = t;

}

void delete\_node(book \*&head, int is)

{

book \*temp = head;

if (temp->isbn == is)

{

head = temp->next;

delete (temp);

return;

}

while (temp->next != NULL && temp->next->isbn != is)

{

temp = temp->next;

}

if (temp->next == NULL || temp == NULL)

{

cout << "the isbn number is wronge.\n";

return;

}

book \*to\_del = temp->next;

temp->next = to\_del->next;

to\_del->next = NULL;

delete (to\_del);

}

void update\_node(book \*&head, int is)

{

book \*temp = head;

string n, a;

cout << "enter the updated name of book : ";

cin >> n;

cout << "enter the updated author name : ";

cin >> a;

int p, i, c;

cout << "enter the updated publication year : ";

cin >> p;

cout << "enter the updated ISBN number : ";

cin >> i;

cout << "enter the updated number of copies : ";

cin >> c;

if (temp->isbn == is)

{

temp->name = n;

temp->a\_name = a;

temp->copy = c;

temp->isbn = i;

temp->pb\_year = p;

return;

}

while (temp != NULL && temp->isbn != is)

{

temp = temp->next;

}

if (temp == NULL)

{

cout << "the isbn number is wronge. or not found in record : \n";

return;

}

temp->name = n;

temp->a\_name = a;

temp->copy = c;

temp->isbn = i;

temp->pb\_year = p;

}

void print(book \*&head)

{

book \*temp = head;

cout << "\nALL BOOK RECORDS \n";

while (temp != NULL)

{

cout << " the name of book : " << temp->name << "\n";

cout << " the author name : " << temp->a\_name << "\n";

cout << " the publication year : " << temp->pb\_year << "\n";

cout << " the ISBN number : " << temp->isbn << "\n";

cout << " the number of copies : " << temp->copy << "\n\n";

temp = temp->next;

}

cout << "\nend\n";

}

};

class issue

{

public:

string name;

string a\_name;

issue \*next;

issue(string n, string a)

{

this->name = n;

this->a\_name = a;

this->next = NULL;

}

void add(issue \*&head)

{

string n, a;

cout << "enter the name of book : ";

cin >> n;

cout << "enter the name of the student who had issued the book : ";

cin >> a;

issue \*t = new issue(n, a);

if (head == NULL)

{

head = t;

return;

}

issue \*temp = head;

while (temp->next != NULL)

{

temp = temp->next;

}

temp->next = t;

}

void delete\_node(issue \*&head, string is)

{

issue \*temp = head;

if (temp->a\_name == is)

{

head = temp->next;

delete (temp);

return;

}

while (temp->next != NULL && temp->next->a\_name != is)

{

temp = temp->next;

}

if (temp->next == NULL || temp == NULL)

{

cout << "the borrower name is not found in our record.\n";

return;

}

issue \*to\_del = temp->next;

temp->next = to\_del->next;

to\_del->next = NULL;

delete (to\_del);

}

void update\_node(issue \*&head, string is)

{

issue \*temp = head;

string n, a;

cout << "enter the updated name of book : ";

cin >> n;

cout << "enter the updated name of the borrower : ";

cin >> a;

if (temp->a\_name == is)

{

temp->name = n;

temp->a\_name = a;

return;

}

while (temp != NULL && temp->a\_name != is)

{

temp = temp->next;

}

if (temp == NULL)

{

cout << "the borrower name is not found in our record.\n";

return;

}

temp->name = n;

temp->a\_name = a;

}

void print(issue \*&head)

{

issue \*temp = head;

cout << "\nALL BOOK RECORDS \n";

while (temp != NULL)

{

cout << "the name of book : " << temp->name << "\n";

cout << " borrower name : " << temp->a\_name << "\n";

temp = temp->next;

}

cout << "\nend\n";

}

};

int main()

{

int t = 1;

book \*head = NULL;

issue \*h = NULL;

while (t)

{

cout << "enter 1 to add a record of book.\nenter 2 to delete a record of book\nenter 3 to update a record of books\nenter 4 to print records of books\nenter 0 to exit";

cout << "\nenter 5 to add a record of issue .\nenter 6 to delete a record of issue\nenter 7 to update a record of issue\nenter 8 to print records of issue\n";

cin >> t;

switch (t)

{

case 1:

head->add(head);

break;

case 2:

int is;

cout << "enter isbn number of the book to be deleted : ";

cin >> is;

head->delete\_node(head, is);

break;

case 3:

// int is;

cout << "enter isbn number of the book to be deleted : ";

cin >> is;

head->update\_node(head, is);

break;

case 4:

head->print(head);

break;

case 0:

break;

}

if (t == 5)

{

h->add(h);

}

else if (t == 6)

{

string s;

cout << "enter name of borrower : ";

cin >> s;

h->delete\_node(h, s);

}

else if (t == 7)

{

// int is;

string s;

cout << "enter name of borrower : ";

cin >> s;

h->update\_node(h, s);

}

else if (t == 8)

{

h->print(h);

}

}

}

Question 3

#include<iostream>

using namespace std;

template<typename t>

class node{

public:

t data;

node \* next;

node(t d)

{

this->data=d;

this->next=NULL;

}

void create(node <t>\* &head,int v)

{

node<t> \* n=new node<t>(v);

if(head==NULL)

{

head=n;

return;

}

node<t> \* temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=n;

}

void delete\_node(node<t> \* &head)

{

t v;

cout<<"enter the value to be deleted : ";

cin>>v;

node<t> \* temp=head;

if(temp->data==v)

{

if(temp->next!=NULL){

head=temp->next;}

else head=NULL;

temp->next=NULL;

delete temp;

return;

}

while( temp->next!=NULL && temp->next->data!=v)

{

temp=temp->next;

}

if(temp->next==NULL || temp==NULL) {cout<<"not found \n"; return;}

node<t> \* to\_del=temp->next;

temp->next=to\_del->next;

to\_del->next=NULL;

delete to\_del;

}

void print(node <t>\* &head)

{

node <t>\* temp=head;

while(temp!=NULL)

{

cout<<temp->data<<" -> ";

temp=temp->next;

}

cout<<"NULL\n";

}

void bubble\_Sort(node<t> \*&head)

{

// bubble sort

node<int> \*i = head;

while (i != NULL)

{

node<int> \*j = head;

while (j->next != NULL)

{

if (j->data > j->next->data)

{

node<int> \*temp = new node<int>(j->data);

j->data = j->next->data;

j->next->data = temp->data;

}

j = j->next;

}

i = i->next;

// head->print(head);

}

}

};

int main()

{

node <int> \* ll1=NULL;

int n;

cout<<"enter the length of link list : ";

cin>>n;

for(int i=0;i<n;i++)

{

int v;

cout<<"enter the value : ";

cin>>v;

ll1->create(ll1,v);

}

node<int> \* ll2=NULL;

node<int> \* temp=ll1;

while(temp!=NULL)

{

ll2->create(ll2,temp->data);

temp=temp->next;

}

ll2->bubble\_Sort(ll2);

temp=ll1;

node<int> \* temp1=ll2;

node<int> \* ll3=NULL;

ll1->bubble\_Sort(ll1);

for(int i=0;i<2\*n;i++)

{

if(temp==NULL)

{

while(temp1!=NULL)

{

ll3->create(ll3,temp1->data);

temp1=temp1->next;

i++;

}

}

else if(temp1==NULL)

{

while(temp!=NULL)

{

ll3->create(ll3,temp->data);

temp=temp->next;

i++;

}

}

else if(temp==NULL && temp1==NULL) break;

else if(temp->data<temp1->data)

{

ll3->create(ll3,temp->data);

temp=temp->next;

}

else

{

ll3->create(ll3,temp1->data);

temp1=temp1->next;

}

}

ll3->print(ll3);

}

Question 4

#include<iostream>

using namespace std;

template<typename t>

class node{

public:

t data;

node \* next;

node(t d)

{

this->data=d;

this->next=NULL;

}

void create(node <t>\* &head,int v)

{

node<t> \* n=new node<t>(v);

if(head==NULL)

{

head=n;

return;

}

node<t> \* temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=n;

}

void delete\_first\_node(node<t> \* &head)

{

if(head==NULL || head->next == NULL)

{

head=NULL;

return;

}

else

{

node<t> \* temp=head;

head=head->next;

temp->next=NULL;

delete(temp);

}

}

void delete\_last\_node(node<t> \* &head)

{

node<t> \* temp=head;

if(temp->next==NULL)

{

head=NULL;

return;

}

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

{

temp=temp->next;

}

node<t> \* to\_del=temp->next;

temp->next=NULL;

delete(to\_del);

}

void print(node <t>\* &head)

{

node <t>\* temp=head;

while(temp!=NULL)

{

cout<<temp->data<<" -> ";

temp=temp->next;

}

cout<<"NULL\n";

}

int sum(node<t> \*&head)

{

node<int> \*i = head;

int sum=0;

while (i != NULL)

{

sum+=i->data;

i=i->next;

}

return sum;

}

int last\_node(node<t> \*&head)

{

node<int> \*i = head;

while (i->next != NULL)

{

i=i->next;

}

return i->data;

}

};

int main()

{

node <int> \* ll1=NULL;

int n;

cout<<"enter the length of link list : ";

cin>>n;

for(int i=0;i<n;i++)

{

int v;

cout<<"enter the value : ";

cin>>v;

ll1->create(ll1,v);

}

node<int> \* ll2=NULL;

int m;

cout<<"enter the length of link list : ";

cin>>m;

for(int i=0;i<m;i++)

{

int v;

cout<<"enter the value : ";

cin>>v;

ll2->create(ll2,v);

}

if(( ll1->sum(ll1) - ll2->sum(ll2) )> ll1->last\_node(ll1) )

{

while(ll1!=NULL && ( (ll1->sum(ll1) - ll2->sum(ll2)) >0 ))

{

ll1->delete\_last\_node(ll1);

}

}

else

{

while(ll1!=NULL && ((ll1->sum(ll1)-ll2->sum(ll2))>0))

{

ll1->delete\_first\_node(ll1);

}

}

ll1->print(ll1);

}