//a1

#include<iostream>

using namespace std;

class complex{

int real,img;

public:

complex(){

real=0;

img=0;

}

complex operator+(complex c1){

complex tmp;

tmp.real=real+c1.real;

tmp.img=img+c1.img;

return tmp;

}

complex operator\*(complex c2){

complex tmp1;

tmp1.real=(real\*c2.real)-(img\*c2.img);

tmp1.img=(real\*c2.img)+(c2.real\*img);

return tmp1;

}

friend istream& operator>>(istream&, complex&);

friend ostream& operator<<(ostream&, complex&);

};

istream& operator>>(istream& din, complex& c3)

{

din>>c3.real>>c3.img;

}

ostream& operator<<(ostream& dout, complex& c4)

{

dout<<c4.real<<"+"<<c4.img<<"i";

}

int main(){

complex c1,c2,c3,c4;

cout<<"default representation of complex numbers"<<endl;

cout<<" "<<c1<<endl;

cout<<"Enter first complex number"<<endl;

cin>>c2;

cout<<"Enter second complex number"<<endl;

cin>>c3;

c4= c2+c3;

cout<<"Addition is"<<c4<<endl;

c4=c2\*c3;

cout<<"Multiplication is"<<c4<<endl;

return 0;

}

//a2

#include<iostream>

using namespace std;

class stud

{

string name;

int roll;

char div;

string dob,phone;

public:

stud()

{

name="";

roll=0;

div='A';

dob="";

phone="";

}

stud(string n,int r,char d,string db,string ph)

{

name=n;

roll=r;

div=d;

dob=db;

phone=ph;

}

stud(stud& othr){

name=othr.name;

roll=othr.roll;

div=othr.div;

phone=othr.phone;

dob=othr.dob;

}

~stud()

{

cout<<"Destuctor called for student "<<name<<"."<<endl;

}

static void disp(const stud& s1)

{

cout<<"\nName\tRoll\tDiv\tDOB\tPHONE";

cout<<endl<<s1.name<<"\t"<<s1.roll<<"\t"<<s1.div<<"\t"<<s1.dob<<"\t"<<s1.phone<<"\t";

}

};

int main()

{

stud s1("Disha",1,'A',"24/2","157137368");

stud s2=s1;

stud::disp(s1);

stud::disp(s2);

stud\* s3=new stud("xyz",2,'B',"1/2","24244");

stud::disp(\*s3);

delete s3;

return 0;

}#include<iostream>

using namespace std;

class stud

{

string name;

int roll;

char div;

string dob,phone;

public:

stud()

{

name="";

roll=0;

div='A';

dob="";

phone="";

}

stud(string n,int r,char d,string db,string ph)

{

name=n;

roll=r;

div=d;

dob=db;

phone=ph;

}

stud(stud& othr){

name=othr.name;

roll=othr.roll;

div=othr.div;

phone=othr.phone;

dob=othr.dob;

}

~stud()

{

cout<<"Destuctor called for student "<<name<<"."<<endl;

}

static void disp(const stud& s1)

{

cout<<"\nName\tRoll\tDiv\tDOB\tPHONE";

cout<<endl<<s1.name<<"\t"<<s1.roll<<"\t"<<s1.div<<"\t"<<s1.dob<<"\t"<<s1.phone<<"\t";

}

};

int main()

{

stud s1("Disha",1,'A',"24/2","157137368");

stud s2=s1;

stud::disp(s1);

stud::disp(s2);

stud\* s3=new stud("xyz",2,'B',"1/2","24244");

stud::disp(\*s3);

delete s3;

return 0;

}

//a3

#include<iostream>

#include<string.h>

using namespace std;

class publication

{

public:

string name;

float price;

void d()

{

cout<<"Enter book name: "<<endl;

cin>>name;

cout<<"Enter price: "<<endl;

cin>>price;

}

};

class book:public publication

{

public:

float pgcount;

int b,x;

void d1()

{

cout<<"Enter pg-count: "<<endl;

cin>>pgcount;

b=pgcount;

if(b!=pgcount)

{

try{

throw x;

}

catch(int x)

{

cout<<"Name of book: --"<<endl<<"Price: 00"<<endl<<"Pg-count:00 "<<endl;

}

}

else{

cout<<"Name of book: "<<name<<endl<<"Price: "<<price<<endl<<"Pg-count: "<<pgcount<<endl; }

}};

class tape:public publication

{

public:

float pt;

int b,x;

void d2()

{

cout<<"Enter playtime"<<endl;

cin>>pt;

b=pt;

if(b!=pt)

{

try{

throw x;

}

catch(int x)

{

cout<<"Name of book: "<<name<<endl<<"Price: "<<price<<endl<<"Plat time: "<<pt<<endl;

}

}

else{

cout<<"Name of book: --"<<endl<<"Price: 00"<<endl<<"Play time:00.00 "<<endl;

}

}

};

int main()

{

char c;

int ch;

do{

cout<<"1.Book"<<endl<<"2.Tape"<<endl<<"Enter your choice: "<<endl;

cin>>ch;

book b1;

tape t1;

switch(ch)

{

case 1:

b1.d();

b1.d1();

break;

case 2:

t1.d();

t1.d2();

break;

default:

cout<<"Invalid choice";

break;

}

cout<<"Press y to continue: ";

cin>>c;

}while(c=='y');

}

//b4

#include<iostream>

#include<fstream>

using namespace std;

class Student

{

int roll;

char name[20];

char div;

public:

void accept()

{

cin>>name;

cin>>roll;

cin>>div;

}

void display()

{

cout<<"Enter Name: "<<name<<endl;

cout<<"Enter Id: "<<roll<<endl;

cout<<"Enter Div: "<<div<<endl;

}

};

int main()

{

Student s[5];

fstream f;

int i,n;

f.open("h.txt",ios::out||ios::app);

cout<<"How many student's data to store: ";

cin>>n;

cout<<"Enter Information in (Name/Id/Div) format---"<<endl;

for(i=1;i<=n;i++)

{

cout<<"Enter information of "<<i<<" student: "<<endl;

s[i].accept();

f.write((char\*)&s[i],sizeof s[i]);

}

f.close();

f.open("h.txt",ios::in);

cout<<"Information displayed: \n";

for(i=1;i<=n;i++)

{

f.read((char\*)&s[i],sizeof s[i]);

s[i].display();

}

f.close();

return 0;

}

//b5

/\*Write a function template for selection sort that inputs, sorts and outputs an integer array and a float array.\*/

#include <iostream>

#define size 10

using namespace std;

int n;

template <class T>

void sort(T s[size])

{

int min;

for (int i = 0; i < n - 1; i++)

{

min = i;

for (int j = i+1; j < n; j++)

{

if (s[j] < s[min])

{

min = j;

}

}

T temp = s[i];

s[i] = s[min];

s[min] = temp;

}

cout << "the sorted array is : ";

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

{

cout << s[k] << " ";

}

cout << endl;

}

int main()

{

int A[size];

float B[size];

int ch;

char c;

cout << " 1)Integer array\n 2)Float array" << endl;

cout << "Enter your choice : ";

cin >> ch;

switch (ch)

{

case 1:

cout << "No. of integer elements : ";

cin >> n;

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

{

cout << "Enter element ";

cin >> A[i];

}

sort(A);

break;

case 2:

cout << "No. of float elements : ";

cin >> n;

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

{

cout << "Enter element ";

cin >> B[i];

}

sort(B);

break;

default:

cout << "Enter valid choice..." << endl;

break;

}

}//c6A

/\*Write C++ program using STL for sorting and searching user defined records such as personal records (Name, DOB, Telephone number etc) using vector container.\*/

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

class stl

{

public:

string name, dob;

long tel;

void add()

{

cout<<endl;

cout << "Enter your Name : ";

cin >> name;

cout << "Enter your DOB : ";

cin >> dob;

cout << "Enter your TeliphoneNo : ";

cin >> tel;

}

void display()

{

cout << name << "\t\t" << dob << "\t\t" << tel << endl;

}

};

int main()

{

int n, ch;

char ch1;

string key;

cout << "No. of data you want to add ";

cin >> n;

vector<stl> v(n);

vector<string> name(n);

do

{

cout << "\n 1)Addition of data\n 2)Sorting of data\n 3)Searching of data\n"

<< endl;

cout << "Enter your choice ";

cin >> ch;

switch (ch)

{

case 1:

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

{

v[i].add();

}

break;

case 2:

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

{

name[i] = v[i].name;

}

sort(name.begin(), name.end());

cout << "...Sorted data..." << endl;

cout << "Name\t\tDOB\t\tTeliphone-No" << endl;

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

{

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

{

if (name[i] == v[j].name)

{

v[j].display();

break;

}

}

}

break;

case 3:

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

{

name[i] = v[i].name;

}

sort(name.begin(), name.end());

cout << "Enter the name that you want to search ";

cin >> key;

if (binary\_search(name.begin(), name.end(), key))

{

cout << "Record found..." << endl;

}

else

{

cout << "Record not found..."<<endl;

}

break;

default:

break;

}

cout << "Do you want to continue other oprations (y/n) : ";

cin >> ch1;

} while (ch1 == 'y');

}

//c6b

/\*

Write C++ program using STL for Sorting and searching with user-defined records such as

Person Record (Name, birth date, telephone no), item record (item code, item name, quantity

and cost)

\*/

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

class Item

{

public:

char name[10];

int quantity;

int cost;

int code;

bool operator==(const Item& i1) //Boolean operators allow you to create more complex conditional statements

{

if(code==i1.code) //operator will return 1 if the comparison is true, or 0 if the comparison is false

return 1;

return 0;

}

bool operator<(const Item& i1)

{

if(code<i1.code) //operator will return 1 if the comparison is true, or 0 if the comparison is false

return 1;

return 0;

}

};

vector<Item> o1;

void print(Item &i1);

void display();

void insert();

void search();

void dlt();

bool compare(const Item &i1, const Item &i2)

{

//if (i1.name != i2.name) return i1.cost < i2.cost;

return i1.cost < i2.cost;

}

int main()

{

int ch;

do

{

cout<<"\n\*\*\*\*\* Menu \*\*\*\*\*";

cout<<"\n1.Insert";

cout<<"\n2.Display";

cout<<"\n3.Search";

cout<<"\n4.Sort";

cout<<"\n5.Delete";

cout<<"\n6.Exit";

cout<<"\nEnter your choice:";

cin>>ch;

switch(ch)

{

case 1:

insert();

break;

case 2:

display();

break;

case 3:

search();

break;

case 4:

sort(o1.begin(),o1.end(),compare);

cout<<"\n\n Sorted on Cost";

display();

break;

case 5:

dlt();

break;

case 6:

exit(0);

}

}while(ch!=7);

return 0;

}

void insert()

{

Item i1;

cout<<"\nEnter Item Name:";

cin>>i1.name;

cout<<"\nEnter Item Quantity:";

cin>>i1.quantity;

cout<<"\nEnter Item Cost:";

cin>>i1.cost;

cout<<"\nEnter Item Code:";

cin>>i1.code;

o1.push\_back(i1);

}

void display()

{

for\_each(o1.begin(),o1.end(),print);

}

void print(Item &i1)

{

cout<<"\n";

cout<<"\nItem Name:"<<i1.name;

cout<<"\nItem Quantity:"<<i1.quantity;

cout<<"\nItem Cost:"<<i1.cost;

cout<<"\nItem Code:"<<i1.code;

}

void search()

{

vector<Item>::iterator p;

Item i1;

cout<<"\nEnter Item Code to search:";

cin>>i1.code;

p=find(o1.begin(),o1.end(),i1);

if(p==o1.end())

{

cout<<"\nNot found.";

}

else

{

cout<<"\nFound.";

}

}

void dlt()

{

vector<Item>::iterator p;

Item i1;

cout<<"\nEnter Item Code to delete:";

cin>>i1.code;

p=find(o1.begin(),o1.end(),i1);

if(p==o1.end())

{

cout<<"\nNot found.";

}

else

{

o1.erase(p);

cout<<"\nDeleted.";

}

}

//c7

#include<iostream>

#include<map>

using namespace std;

class pMap{

public:

map<string,int> pMap;

string state;

int pop;

void add()

{

cout<<"Enter state: ";

cin>>state;

cout<<"\nEnter population: ";

cin>>pop;

pMap[state]=pop;

}

void d()

{

map<string,int>::iterator it;

for(it=pMap.begin();it!=pMap.end();it++)

{

cout<<it->first<<" : "<<it->second<<endl;

}

}

void find()

{

string s;

cout<<"Enter state whose population u wanna find: ";

cin>>s;

map<string,int>::iterator it;

it=pMap.find(s);

if(it!=pMap.end())

{

cout<<"FOUND!!";

cout<<s<<" population is: "<<it->second<<" million";

}

else{

cout<<"\nNOT FOUND!!";

}

}

};

int main()

{

pMap p;

int n;

cout<<"Enter no of records: ";

cin>>n;

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

{

p.add();

}

p.d();

p.find();

return 0;

}