









CODE

//Sassy\_Don30

#include <bits/stdc++.h>

using namespace std;

class Person{

private:

int rollNo; // 3 data members liye class me

string name;

string dateOfBirth;

public:

bool operator ==(const Person &student1){ // iterator me use hota ye

return (this->rollNo==student1.rollNo);

}

bool operator <(const Person &student1){ // comparison wale ko overload kiya hai

return (this->rollNo < student1.rollNo);

}

//output operator ko overload << kiya hai

friend ostream& operator <<(ostream &out, const Person &stu){

out<<"\n\t\t"<<stu.rollNo<<"\t\t"<<stu.name<<"\t\t"<<stu.dateOfBirth;

return out;

}

// input >> ko overload kiya hai

friend istream& operator >>(istream &in, Person &stu){

cout<<"\nEnter roll number: ";

cin>>stu.rollNo;

cout<<"\nEnter name: ";

cin>>stu.name;

cout<<"\nEnter date of birth: (ddmmyy)";

cin>>stu.dateOfBirth;

return in;

}

// getter aur setter ki jarorat pad rahi warna variables public karne padenge

void SetRoll(){

cin>>this->rollNo;

}

int GetRoll(){

return this->rollNo;

}

};

// jo upar comparison wala overload kiya voh yaha use kiya roll number compare karne ke liye

bool Compare(Person& stu1, Person& stu2){

return (stu1<stu2); // true false return karega

}

// end me vector return karne objects ka after reading data

vector<Person> ReadData(){

int total;

Person stu;

vector<Person> stuData;

cout<<"\nEnter total students: "; // total count input karne ke baad

cin>>total;

for(int i=0;i<total;i++){

cin>>stu; // push back karte hue vector fill kiya

stuData.push\_back(stu);

}

return stuData;

}

// ye nahi kiya toh for each loop use nahi ho payga, aur mujhe karne ki iccha hai

void PrintFunction(const Person& stu){

cout<<stu; // overload wala use kiya yaha pe

}

void PrintData(const vector<Person> &stu){

cout<<"\n\t\tROLL NO\t\tNAME\t\tDATE OF BIRTH"; // for each loop matlab us data ke haar item me jayga

for\_each(stu.begin(),stu.end(),PrintFunction); // aur print function call hoga jo upar likha hai

}

//create karne ke baad agar push back karna hai uske liye hai ye

void InsertData(vector<Person> &stu){

Person temp;

cin>>temp;

stu.push\_back(temp); // insert karke randomly bhi kar sakte but koi na

}

void SearchData(vector<Person> &stuData){

Person temp;

cout<<"\nEnter roll number to be searched: ";

temp.SetRoll();

vector<Person>::iterator looper; // ek iterator banaya hai looper name ka

// find function start se end tak dhoondhega temp wale ko

looper = find(stuData.begin(),stuData.end(), temp);// yaha pe voh overload wala use ho raha jo sabse pehle likha hai

if(looper != stuData.end()){ // agar end se pehle mil gaya, toh dereference karke print kiya use overload use karke

cout<<"\n\n\t\tROLL NO\t\tNAME\t\tDATE OF BIRTH";

cout<<\*looper; // Student stu = \*looper aise karke bhi kar sakte, fir uske baad print, lekin this is better

}

else

cout<<"\nNot found!"<<endl; // nahi mila toh aage chalo

}

void DeleteData(vector<Person> &stuData){

Person temp;

cout<<"\nEnter roll number to delete: ";

temp.SetRoll();

vector<Person>::iterator looper; // ek iterator banaya vector object ka with name looper jaise upar kiya

looper = find(stuData.begin(), stuData.end(), temp);

if(looper != stuData.end()) // agar beech me mila toh delete

stuData.erase(looper); // in built delete function

else

cout<<"\nNot found"<<endl; // warna not found

}

// Ek line ka function tha but ok

inline void SortData(vector<Person> &stuData){ // inline kar diya ab

sort(stuData.begin(), stuData.end(), Compare);

//PrintData(stuData);

}

int main(){

vector<Person> stuData;

int ch;

do{ // normal do while loop lagaya

cout<<"\nMENU:"; // menu me sab banaya

cout<<"\n1.Create \n2.Display \n3.Insert \n4.Delete \n5.Search \n6.Sort \n0.Quit";

cout<<"\nEnter choice: ";

cin>>ch;

switch(ch){

case 1:

stuData = ReadData(); // data read karega aur kyuki vector return kar rahe toh yaha store hoga

break;

case 2:

PrintData(stuData); // function run kiya print ka

break;

case 3:

InsertData(stuData); // insert delete search sort sabka same hai

break;

case 4:

DeleteData(stuData);

break;

case 5:

SearchData(stuData);

break;

case 6:

SortData(stuData); // pehle sort fir print, warna sort me hi dal sakte print

PrintData(stuData);

break;

default:

cout<<"Enter appropriate value"<<endl;

break;

}

} while(ch != 0); // 0 pe exit karega

return 0;

}

OUTPUT

MENU:

1.Create

2.Display

3.Insert

4.Delete

5.Search

6.Sort

0.Quit

Enter choice: 1

Enter total students: 3

Enter roll number: 4

Enter name: Sassy

Enter date of birth: (ddmmyy)300901

Enter roll number: 2

Enter name: Kirti

Enter date of birth: (ddmmyy)200401

Enter roll number: 9

Enter name: Arihant

Enter date of birth: (ddmmyy)230600

MENU:

1.Create

2.Display

3.Insert

4.Delete

5.Search

6.Sort

0.Quit

Enter choice: 2

ROLL NO NAME DATE OF BIRTH

4 Sassy 300901

2 Kirti 200401

9 Arihant 230600

MENU:

1.Create

2.Display

3.Insert

4.Delete

5.Search

6.Sort

0.Quit

Enter choice: 3

Enter roll number: 1

Enter name: Shreya

Enter date of birth: (ddmmyy)101201

MENU:

1.Create

2.Display

3.Insert

4.Delete

5.Search

6.Sort

0.Quit

Enter choice: 5

Enter roll number to be searched: 3

Not found!

MENU:

1.Create

2.Display

3.Insert

4.Delete

5.Search

6.Sort

0.Quit

Enter choice: 2

ROLL NO NAME DATE OF BIRTH

4 Sassy 300901

2 Kirti 200401

9 Arihant 230600

1 Shreya 101201

MENU:

1.Create

2.Display

3.Insert

4.Delete

5.Search

6.Sort

0.Quit

Enter choice: 6

ROLL NO NAME DATE OF BIRTH

1 Shreya 101201

2 Kirti 200401

4 Sassy 300901

9 Arihant 230600

MENU:

1.Create

2.Display

3.Insert

4.Delete

5.Search

6.Sort

0.Quit

Enter choice: 4

Enter roll number to delete: 9

MENU:

1.Create

2.Display

3.Insert

4.Delete

5.Search

6.Sort

0.Quit

Enter choice: 2

ROLL NO NAME DATE OF BIRTH

1 Shreya 101201

2 Kirti 200401

4 Sassy 300901

MENU:

1.Create

2.Display

3.Insert

4.Delete

5.Search

6.Sort

0.Quit

Enter choice: 0

Enter appropriate value