include <iostream>

#include<string>​

using namespace std;​

​

//Class of Persons​

class Person​

{​

    private:​

        string name;​

        float salary;​

    public:​

        void setParson();​

        string getName();​

        float getSalary();​

};​

//Creates person with user input name and salary​

void Person::setParson()​

{​

    cout << "Enter name: ";​

    cin >> name;​

    cout << "Enter Salary: ";​

    cin >> salary;​

}​

//Returns person's name​

string Person::getName()​

{​

    return name;​

}​

//Returns person's name​

float Person::getSalary()​

{​

    return salary;​

}​

//n is an the number of person objects stored in the array​

//if s is true, then sort by name;otherwise, sort by salary.​

void bsort(Person\*\* p,int n,bool s)​

{​

    for(int i = 0;i < n;i++)​

    {​

        for(int j = i+1;j < n;j++)​

        {​

            Person\* temp;​

            if((s==false && p[i] -> getSalary() > p[j] -> getSalary()) || (s==true && p[i] -> getName() > p[j] -> getName()))​

            {​

                temp = p[i];​

                p[i] = p[j];​

                p[j] = temp;​

            }​

        }​

    }​

}​

//This method print the array of name and salary​

void print(Person\*\* p,int n)​

{​

    for(int i = 0;i < n;i++)​

    {​

        cout << (p[i]->getName()) << " " << (p[i] -> getSalary()) << endl;​

    }​

}​

//This method is user inputted and prints out unsorted​

//and sorted list via alphabetical and salary base​

int main()​

{​

    int n;​

    cout << "Enter number of person: ";​

    cin >> n;​

    Person\* p[n];​

    for(int i = 0;i < n;i++)​

    {​

        p[i] = new Person();​

        p[i] -> setParson();​

    }​

    cout << "\nUnsorted order" << endl;​

    print(p,n);​

    cout << endl;​

    cout << "Sorted base of name alphabetically" << endl;​

    bsort(p,n,true);​

    print(p,n);​

    cout << endl;​

    cout << "Sorted based on salary from least to greatest" << endl;​

    bsort(p,n,false);​

    print(p,n);​

    cout << endl;​

    return 0;​

}

Phuong Nguyen