**Bài tập ứng dụng 3:**

#include <iostream>

using namespace std;

//Cấu trúc cho phòng thi

struct PhongThi {

    int SoPhong;

    char Nha;

    int KhaNangChua;

};

//In danh sách

void InDanhSach(PhongThi PhongThis[], int n) {

    for(int i = 0; i < n; i++) {

        cout<<"Phong "<<PhongThis[i].SoPhong<<", Nha "<<PhongThis[i].Nha<<", Kha nang chua: "<<PhongThis[i].KhaNangChua<<endl;

    }

}

//Chia và hợp nhất từng mảng cho Merge Sort

void Merge(PhongThi PhongThis[], int low, int mid, int high) {

    int leftSize = mid - low + 1;

    int rightSize = high - mid;

    PhongThi \*left = new PhongThi[leftSize];

    PhongThi \*right = new PhongThi[rightSize];

    for(int i = 0; i < leftSize; i++) left[i] = PhongThis[low + i];

    for(int j = 0; j < rightSize; j++) right[j] = PhongThis[mid + 1 + j];

    int i = 0, j = 0, k = low;

    while(i < leftSize && j < rightSize) {

        if (left[i].KhaNangChua > right[j].KhaNangChua) {

            PhongThis[k++] = left[i++];

        } else {

            PhongThis[k++] = right[j++];

        }

    }

    while(i < leftSize) PhongThis[k++] = left[i++];

    while(j < rightSize) PhongThis[k++] = right[j++];

}

//Sử dụng thuật toán Merge Sort

void MergeSort(PhongThi PhongThis[], int low, int high) {

    if(low < high) {

        int mid = (low + high) / 2;

        MergeSort(PhongThis, low, mid);

        MergeSort(PhongThis, mid + 1, high);

        Merge(PhongThis, low, mid, high);

    }

}

//Sắp xếp dựa theo Nhà và Số phòng

void MergeNhaVaPhong(PhongThi PhongThis[], int low, int mid, int high) {

    int leftSize = mid - low + 1;

    int rightSize = high - mid;

    PhongThi \*left = new PhongThi[leftSize];

    PhongThi \*right = new PhongThi[rightSize];

    for(int i = 0; i < leftSize; i++) left[i] = PhongThis[low + i];

    for(int j = 0; j < rightSize; j++) right[j] = PhongThis[mid + 1 + j];

    int i = 0, j = 0, k = low;

    while(i < leftSize && j < rightSize) {

        if(left[i].Nha < right[j].Nha || (left[i].Nha == right[j].Nha && left[i].SoPhong < right[j].SoPhong)) {

            PhongThis[k++] = left[i++];

        } else {

            PhongThis[k++] = right[j++];

        }

    }

    while(i < leftSize) PhongThis[k++] = left[i++];

    while(j < rightSize) PhongThis[k++] = right[j++];

}

//Merge Sort Nhà và Phòng

void MergeSortNhaVaPhong(PhongThi PhongThis[], int low, int high) {

    if (low < high) {

        int mid = (low + high) / 2;

        MergeSortNhaVaPhong(PhongThis, low, mid);

        MergeSortNhaVaPhong(PhongThis, mid + 1, high);

        MergeNhaVaPhong(PhongThis, low, mid, high);

    }

}

//Sắp xếp dựa theo Nhà và Khả năng chứa

void MergeNhaVaKhaNangChua(PhongThi PhongThis[], int low, int mid, int high) {

    int leftSize = mid - low + 1;

    int rightSize = high - mid;

    PhongThi \*left = new PhongThi[leftSize];

    PhongThi \*right = new PhongThi[rightSize];

    for (int i = 0; i < leftSize; i++) left[i] = PhongThis[low + i];

    for (int j = 0; j < rightSize; j++) right[j] = PhongThis[mid + 1 + j];

     int i = 0, j = 0, k = low;

    while(i < leftSize && j < rightSize) {

        if (left[i].Nha < right[j].Nha || (left[i].Nha == right[j].Nha && left[i].KhaNangChua < right[j].KhaNangChua))

            PhongThis[k++] = left[i++];

        else

            PhongThis[k++] = right[j++];

    }

    while(i < leftSize) PhongThis[k++] = left[i++];

    while(j < rightSize) PhongThis[k++] = right[j++];

}

//Merge Sort cho Nhà và Khả năng chứa

void MergeSortNhaVaKhaNangChua(PhongThi PhongThis[], int low, int high) {

    if (low < high) {

        int mid = (low + high) / 2;

        MergeSortNhaVaKhaNangChua(PhongThis, low, mid);

        MergeSortNhaVaKhaNangChua(PhongThis, mid + 1, high);

        MergeNhaVaKhaNangChua(PhongThis, low, mid, high);

    }

}

int main() {

    int n;

    cout<<"Moi ban nhap so luong phong thi: ";

    cin >> n;

    if(n < 1 || n > 200) {

        cout<<"Ban da nhap sai, moi nhap lai: ";

        cin>>n;

    }

    PhongThi PhongThis[200];

        for(int i = 0; i < n; i++) {

        cout<<"Nhap thong tin cho phong thi thu " << i + 1 << ":" << endl;

        cout<<"So phong: ";

        cin>>PhongThis[i].SoPhong;

        if(PhongThis[i].SoPhong < 1 || PhongThis[i].SoPhong > 200) {

            cout<<"Ban da nhap sai, moi nhap lai: ";

            cin>>PhongThis[i].SoPhong;

        }

        cout<<"Nha: ";

        cin>>PhongThis[i].Nha;

        if (PhongThis[i].Nha < 'A' || PhongThis[i].Nha > 'Z') {

            cout<<"Ban da nhap sai, moi nhap lai: ";

            cin>>PhongThis[i].Nha;

        }

        cout<<"Kha nang chua: ";

        cin>>PhongThis[i].KhaNangChua;

        if(PhongThis[i].KhaNangChua < 10 || PhongThis[i].KhaNangChua > 250) {

            cout<<"Ban da nhap sai, moi nhap lai: ";

            cin>>PhongThis[i].KhaNangChua;

        }

    }

    MergeSort(PhongThis, 0, n - 1);

    cout<<"\nDanh sach phong thi theo kha nang chua:\n";

    InDanhSach(PhongThis, n);

    MergeSortNhaVaPhong(PhongThis, 0, n - 1);

    cout<<"\nDanh sach phong thi theo nha va so phong:\n";

    InDanhSach(PhongThis, n);

    MergeSortNhaVaKhaNangChua(PhongThis, 0, n - 1);

    cout<<"\nDanh sach phong thi theo nha va kha nang chua:\n";

    InDanhSach(PhongThis, n);

    return 0;

}

Ý niệm: Sử dụng thuật toán Merge Sort để chia số phòng thi ra thành từng mảng và rồi hợp nhất và sắp xếp lại theo yêu cầu.

Vd: Moi ban nhap so luong phong thi: 3

Nhap thong tin cho phong thi thu 1:

So phong: 123

Nha: A

Kha nang chua: 10

Nhap thong tin cho phong thi thu 2:

So phong: 135

Nha: B

Kha nang chua: 15

Nhap thong tin cho phong thi thu 3:

So phong: 159

Nha: C

Kha nang chua: 20

Kết quả cho ra:

Danh sach phong thi theo kha nang chua:

Phong 159, Nha C, Kha nang chua: 20

Phong 135, Nha B, Kha nang chua: 15

Phong 123, Nha A, Kha nang chua: 10

Danh sach phong thi theo nha va so phong:

Phong 123, Nha A, Kha nang chua: 10

Phong 135, Nha B, Kha nang chua: 15

Phong 159, Nha C, Kha nang chua: 20

Danh sach phong thi theo nha va kha nang chua:

Phong 123, Nha A, Kha nang chua: 10

Phong 135, Nha B, Kha nang chua: 15

Phong 159, Nha C, Kha nang chua: 20