//============================================================================

// Name : 最终.cpp

// Author :

// Version :

// Copyright : Your copyright notice

// Description : Hello World in C++, Ansi-style

//============================================================================

#include <iostream>

#include <cstdio>

#include <cstdlib>

#include <string>

#include <vector>

#include <algorithm>

using namespace std;

//学生结构体

struct stu {

int num; //学号

double c; //绩点

bool chose; //是否被选择

int teacher[6]; //所选择的导师数组

};

//导师所拥有学生的结构体

struct stu2 {

int num;

double c;

};

//导师结构体

struct tea {

vector<stu2> student; //所拥有的学生结构体向量组

int o; //当前所拥有的学生数

int up\_limit; //所能带学生的最大数量

int chosen[9]; //最终所选择的学生数组

};

bool comp2(const stu2 &a, const stu2 &b) {

return a.c > b.c;

} //比较选同一个导师中的学生，按绩点排序

bool comp(const stu &a, const stu &b) {

return a.c > b.c;

} //比较全体学生，选出前20%与后20%，优先选择

int main() {

//文件输入输出

FILE \*fin, \*fout;

fin = fopen("/Users/cafe3165/Desktop/randon\_out.txt", "rb");

fout = fopen("/Users/cafe3165/Desktop/consequense2\_out.txt", "wb");

vector<stu> S; //存放全体学生结构体的向量组

vector<stu2> S2; //存放选择导师的学生组

vector<tea> T; //导师组

int count = 0; //记录有多少学生被选了

//int count2 = 0;

stu s; //实例化

stu2 s2; //实例化

int snum; //学生数

fprintf(fout, "请输入选课学生数：\n");

fscanf(fin, "%d", &snum);

fprintf(fout, "%d\n", snum);

fprintf(fout, "请依次输入学号、绩点、以及5位导师：\n");

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

//由文件输入 学生的学号、绩点

fscanf(fin, "%d %lf", &s.num, &s.c);

//第二个学生数组赋值

s2.num = s.num;

s2.c = s.c;

//由文本输入 学生的五个志愿

for (int j = 0; j < 5; j++) {

fscanf(fin, "%d", &s.teacher[j]);

}

//将学生结构体送入向量组

S.push\_back(s);

S2.push\_back(s2);

}

int tnum; //导师数量

fprintf(fout, "请输入导师数:\n");

fscanf(fin, "%d", &tnum);

fprintf(fout, "%d\n", tnum);

tea t;

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

t.o = 0;

int ul = random() % 9; //随机生成导师所能带领的最大学生数

if (ul > 8 || ul < 0)

ul = 8; //有时候会生成一个很大的数字，将它设置成8

t.up\_limit = ul; //设置导师最大所能带的学生数

T.push\_back(t); //将导师送入向量组

}

//此循环用于将学生的学号送入他所填志愿的导师的学生向量组中

// for (int i = 0; i <= snum; i++) {

//

// for (int j = 0; j < 5; j++) {

// int tt = S[i].teacher[j]; //导师号

// T[tt].student.push\_back(S2[i]); //将自己的学号送进去

// }

// }

//

//

// for (int i = 1; i <= tnum; i++) {

// sort(T[i].student.begin(), T[i].student.end(), comp2);//对每个导师的拥有学生进行绩点排序

// }

//全体学生进行排序

sort(S.begin(), S.end(), comp);

//绩点靠后的20%先排序，照顾他们，不然按照绩点排序很容易选不上

for (int i = snum - 1; i > snum - (snum / 5); i--) {

for (int j = 0; j < 5; j++) {

int tno = S[i].teacher[j]; //tno为导师编号

//判断该导师还有没有剩余名额，有的话将自己注册进去

if (T[tno].o < T[tno].up\_limit) {

int to = T[tno].o;

T[tno].chosen[to] = S[i].num;

T[tno].o++;

count++;

S[i].chose = true;

break;

}

//若果该学生的第一志愿导师满了的话，就跳到第二志愿去选导师，以此类推

else

continue;

}

}

//原理与后20%的学生相似，此为绩点排名前20%的学生选择导师

for (int i = 0; i < snum / 5; i++) {

for (int j = 0; j < 5; j++) {

int tno = S[i].teacher[j];

if (T[tno].o < T[tno].up\_limit) {

int to = T[tno].o;

T[tno].chosen[to] = S[i].num;

T[tno].o++;

count++;

S[i].chose = true;

break;

}

else

continue;

}

}

int p = snum / 5;

int k = (snum \* 4) / 5;

//接下来进行排名居中的60%的同学选导师，分为两个部分，同时进行，选择原理相似

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

//从21%往后选到50%

for (int j = 0; j < 5; j++) {

int tno = S[p].teacher[j];

if (T[tno].o < T[tno].up\_limit) {

int to = T[tno].o;

T[tno].chosen[to] = S[p].num;

T[tno].o++;

count++;

S[p].chose = true;

break;

}

else {

continue;

}

}

p++;

if (p - 1 == k)

break;

//从79%往前选择到50%

for (int j = 0; j < 5; j++) {

int tno = S[k].teacher[j];

if (T[tno].o < T[tno].up\_limit) {

int to = T[tno].o;

T[tno].chosen[to] = S[k].num;

T[tno].o++;

count++;

S[k].chose = true;

break;

} else

continue;

}

k--;

if (p > k)

break;

}

fprintf(fout, "选课结束！\n");

for (int i = 1; i <= tnum; i++) {

fprintf(fout, "第 %d 位导师所有的学生：", i);

switch (T[i].o) {

case 0:

fprintf(fout, "导师%d今年不带学生 .\n", i);

break;

case 1:

fprintf(fout, "导师%d可以带%d个学生，带了1个学生是 %d .\n", i, T[i].up\_limit,T[i].chosen[0]);

break;

case 2:

fprintf(fout, "导师%d可以带%d个学生，带了2个学生是 %d %d .\n", i, T[i].up\_limit,T[i].chosen[0],

T[i].chosen[1]);

break;

case 3:

fprintf(fout, "导师%d可以带%d个学生，带了3个学生是 %d %d %d .\n", i, T[i].up\_limit,T[i].chosen[0],

T[i].chosen[1], T[i].chosen[2]);

break;

case 4:

fprintf(fout, "导师%d可以带%d个学生，带了4个学生是 %d %d %d %d .\n", i, T[i].up\_limit,T[i].chosen[0],

T[i].chosen[1], T[i].chosen[2], T[i].chosen[3]);

break;

case 5:

fprintf(fout, "导师%d可以带%d个学生，带了5个学生是 %d %d %d %d %d .\n", i, T[i].up\_limit,T[i].chosen[0],

T[i].chosen[1], T[i].chosen[2], T[i].chosen[3],

T[i].chosen[4]);

break;

case 6:

fprintf(fout, "导师%d可以带%d个学生，带了6个学生是 %d %d %d %d %d %d .\n", i,

T[i].up\_limit,T[i].chosen[0], T[i].chosen[1], T[i].chosen[2],

T[i].chosen[3], T[i].chosen[4], T[i].chosen[5]);

break;

case 7:

fprintf(fout, "导师%d可以带%d个学生，带了7个学生是 %d %d %d %d %d %d %d .\n", i,

T[i].up\_limit,T[i].chosen[0], T[i].chosen[1], T[i].chosen[2],

T[i].chosen[3], T[i].chosen[4], T[i].chosen[5],

T[i].chosen[6]);

break;

case 8:

fprintf(fout, "导师%d可以带%d个学生，带了8个学生是 %d %d %d %d %d %d %d %d.\n", i,

T[i].up\_limit,

T[i].chosen[0], T[i].chosen[1], T[i].chosen[2],

T[i].chosen[3], T[i].chosen[4], T[i].chosen[5],

T[i].chosen[6], T[i].chosen[7]);

break;

default:

fprintf(fout, "系统崩溃!\n");

}

}

int count\_stu = 0;

fprintf(fout, "未选名单：\n");

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

if (S[i].chose == true)

continue;

else {

fprintf(fout, " %d \n", S[i].num);

count\_stu++;

}

}

if (count\_stu == 0) {

fprintf(fout, "无\n");

} else {

fprintf(fout, " 未中选人数：%d \n", count\_stu);

}

cout << "已选人数：" << count << endl;

fprintf(fout, "已选人数：%d", count);

fclose(fin);

fclose(fout);

return 0;

}