

姓名	王昱	学号	PB21030814	日期	2022. 5. 22
实验名称	随机分组				
实验环境： CPU: Intel(R) Core(TM) i7-10870H CPU @ 2.20GHz 2.21 GHz 内存: 16.0 GB (15.8 GB 可用) 操作系统: Windows 11 家庭中文版 软件平台: Visual Studio Code 1.65.2					
一、问题分析与求解思路 本次实验是我与杨峻尧同学一起完成的。我们讨论一起提出的需求，实现是两个人分工完成的，我的分工大概是结构体和变量命名初始化、分项目程序、计算权重、生成学生初始信息、读取学生初始信息。 本次实验我们尽可能地靠近实际情况来提出我们的需求。采用了结构体数组，结构体中包含的学生信息有姓名、学号、对每个项目的意向值、一定不组队的人的学号、对每个项目的熟悉程度以及意向值与熟悉程度的加权值，旨在更好地模拟现实当中学生分组的情况。 与此同时，我们还设置了分组人数的上下限，最多 40 人成组最少 20 人成组（每组 5 到 6 人）。在计算加权值的时候将意向值占比设置为 25%，熟悉程度占比设置为 75%。 整个实验共有 6 个项目，学生人数 200 个。 具体实现的思路是：先根据每个学生对项目的意向值进行初筛成组；然后根据加权值，将人数爆出的组的成员分给人数最少的组别。如果人数少于 20 则将从其他组借人（借的成员位置固定）；接着再考虑这样分出的组是否与 banID 冲突；最后成组。					
二、核心代码说明					
<pre>#define NAME 15 #define PROJECT 6 #define STUDENTNUMBER 200 #define COUNT 300 //随机打乱数组的次数，数字越大打得越乱 //分组的下限是20上限是40 #define LOWERLIMIT 20 #define UPPERLIMIT 40 //如果人数少于20移动的人的位置 #define POSITION 10 //意向值占比0.25 //熟悉程度占比0.75 #define I_Percent 0.25 #define P_percent 0.75 struct randomgroup { char studentName[NAME]; int studentID; int intention[PROJECT]; //对每个项目的意向值 int banID; //一定不组队的人的编号 int proficiency[PROJECT]; //对项目的熟练程度 double weightedValue; //意向值和对项目的熟练程度的加权值 } student[STUDENTNUMBER];</pre>					

```
int projectIntention[PROJECT][STUDENTNUMBER];
int projectStudentNumber[PROJECT];
int ProjectGroupInfo[PROJECT][8][6];
```

①一些准备工作：结构体数组和一些宏定义不再赘述。这里定义了三个全局数组，projectIntention 用于存放 6 个项目下的学生的学号(ID)。projectStudentNumber 用于存放每个项目的人数。ProjectGroupInfo 用于存放每个项目的每个组别下的学生的学号（ID）。

```
//意向考量(判断意向值作为初筛)
//将其放在意向高的那一个项目
//如果对多个项目意向相同且最大，任选一个即可
void chooseAGroup(void)
{
    int Project[PROJECT] = {0};
    for (int i = 0; i < STUDENTNUMBER; i++)
    {
        int maxIntention = 0;
        int project = 0;
        int count = 0;
        for (; count < PROJECT; count++)
        {
            if (maxIntention <= student[i].intention[count])
            {
                maxIntention = student[i].intention[count];
                project = count;
            }
        }
        //把学生的编号放在projectIntention数组里面
        //方便之后调用学生信息
        projectIntention[project][Project[project]++] = i;
        projectStudentNumber[project]++;
    }
}
```

②意向考量作为初筛：根据每个人对 6 个项目的意向值不同进行初次分组，采用两层 for 循环，外层用于遍历学生人数，内层用于遍历 6 个项目。并调整 projectIntention 和 projectStudentNumber 数组的信息。

```
//计算人数爆掉的某个项目中成员的加权值
void calculateWeightedValue(int project)
{
    for (int i = 0; i < projectStudentNumber[project]; i++)
    {
        student[projectIntention[project][i]].weightedValue =
            student[projectIntention[project][i]].intention[i] * I_Percent +
            student[projectIntention[project][i]].proficiency[i] * P_percent;
    }
}
```

③计算加权值：该函数用于之后将初筛每组爆掉的人分给其他组，加权值（对每个项目的意向值与熟悉程度的加权）小的人将会被分配。

```
void generateInfo(void)
{
    srand((unsigned)time(NULL));
    char studentName[STUDENTNUMBER][NAME];
    int studentID;
    int intention[PROJECT];
    int banID;
    int proficiency[PROJECT];
    double weightedValue;
    FILE *fp1 = fopen("name.txt", "r");
    FILE *fp2 = fopen("Info.xls", "w+");
    for (int i = 0; i < STUDENTNUMBER; i++)
    {
        fscanf(fp1, "%s", studentName[i]);
    }
    for (int i = 0; i < STUDENTNUMBER; i++)
    {
        studentID = i;
        //学号整除20的人有banID
        if (i % 20 == 0)
        {
            banID = rand() % 200;
        }
        else
        {
            banID = -1;
        }
        weightedValue = 0;
        for (int j = 0; j < PROJECT; j++)
        {
            intention[j] = rand() % 11;
            proficiency[j] = rand() % 11;
        }
        fprintf(fp2, "%s\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%lf\n", studentName[i], studentID,
            intention[0], intention[1], intention[2], intention[3], intention[4], intention[5], banID, proficiency[0], proficiency[1],
            proficiency[2], proficiency[3], proficiency[4], proficiency[5], weightedValue);
    }
    fclose(fp1);
    fclose(fp2);
}
```

④生成学生信息：这里提前准备了有 200 个名字的 txt 文件，学生的姓名通过读取该文件而得到。每个人的学号范围是[0, 199]，指定学号整除 20 的学生有 banID（即一定不想组队的学生学号），其余学生的 banID 设置为-1。学生的 banID、每个项目的意向值、对每个项目的熟悉程度采用随机数生成。最后利用 fprintf 函数将信息输出到 xls 文件中。

```
void ReadInfo(void)
{
    FILE *fp = fopen("Info.xls", "r");
    for (int i = 0; i < STUDENTNUMBER; i++)
    {
        fscanf(fp, "%s %d %d %d %d %d %d %d %d %d %d %d %d %d %d %f", student[i].studentName, &student[i].studentID,
            &student[i].intention[0], &student[i].intention[1], &student[i].intention[2], &student[i].intention[3], &student[i].intention[4],
            &student[i].intention[5], &student[i].bankID, &student[i].proficiency[0], &student[i].proficiency[1],
            &student[i].proficiency[2], &student[i].proficiency[3], &student[i].proficiency[4], &student[i].proficiency[5], &student[i].weightedValue);
    }
}
```

⑤读取学生信息：将信息放在结构体数组中。

三、测试、运行与分析

中国科学技术大学

Project 1														
Group 1														
Ernst	41	10	5	8	6	6	3	-1	4	8	9	3	3	
Niki	130	8	5	4	4	4	6	-1	9	9	8	7	6	
Narmette	135	9	1	4	3	5	4	-1	1	3	7	5	5	
Kittie	103	10	6	9	5	5	9	-1	1	3	0	10	4	
Phillip	152	10	6	3	9	0	7	-1	9	5	9	6	1	
Bartley	11	9	6	5	6	8	8	-1	1	9	6	9	2	
Group 2														
Verity	194	7	7	6	0	7	9	-1	3	1	5	1	6	
Hamlin	79	9	8	8	2	2	4	-1	0	3	3	7	3	
Magical	120	8	3	2	6	7	6	1	4	10	1	7	9	
Rose	162	8	4	5	5	3	6	-1	10	10	4	3	8	
Ahern	7	8	2	5	3	7	4	-1	8	5	0	4	5	
Group 3														
Roxanne	169	0	1	2	10	9	10	-1	10	0	7	8	7	
Kaela	102	9	6	5	3	4	7	-1	6	3	4	2	5	
Veronique	199	0	7	2	6	5	10	-1	4	3	1	3	4	
Irene	86	10	6	9	2	2	3	-1	10	8	2	7	1	
Gore	60	10	4	7	2	0	0	137	6	10	0	7	2	
Group 4														
Chloe	23	10	3	1	4	8	7	-1	4	0	7	4	8	
Tanner	187	9	0	1	4	2	1	-1	9	6	8	5	7	
Sophie	170	9	6	8	8	4	4	-1	1	1	3	0	1	
Phylcia	156	10	2	2	7	4	5	-1	8	4	4	0	3	
Raven	163	8	1	3	4	3	1	-1	9	7	2	10	8	
Group 5														
Kajal	100	7	5	8	4	5	8	45	1	1	0	9	2	
Len	119	10	2	8	3	7	3	-1	2	8	7	4	5	
Tamah	185	9	2	1	0	0	6	-1	7	2	7	9	7	
Augustine	4	10	7	6	0	1	2	-1	6	6	6	7	0	
Johnny	90	6	1	1	5	3	4	-1	3	3	2	4	4	
Group 6														
Garfield	62	9	1	3	6	6	1	-1	9	3	8	1	6	
Stephanie	171	10	2	9	2	0	4	-1	1	3	9	10	6	
Bard	13	10	7	0	2	9	0	-1	5	10	4	9	3	

运行之后会出现这样的生成这样的 xls 文件，并自动打开。显示项目信息以及组别信息，每个组的成员信息也会出现在表格中。每组的人数上下限，每个项目的人数上下限都满足要求。

四、备注

```
//至少有一组能组成队(不能组队的最大人数是19?)
//大于10不能组队的有13、14、19
//200以内不能组队的有
//1、2、3、4、7、8、9、13、14、19
```

在完成实验的过程中我们发现 200 人不能成组的人数情况只有图片中列出的 10 种，由于原来需求的设置过于简单，故我们修改了最开始讨论的需求，将分组的人数设置上下限，根据加权值踢人。这样会使得分组情况更接近实际。

总结

本次实验是双人编程，首先是增强了我与人沟通交流的能力，然后跟队友的沟通交流中也学到了很多知识，比如以 xls 文件的方式输出更加美观，比如一些更为合理的需求的提出等等。并通过此次实验再次回顾文件的读写操作。

另附（源代码文件名）

randomgroup.c
name.txt