**作 业 报 告**

|  |  |
| --- | --- |
| **课程名称** | C程序设计语言 |
| **作业名称** | 用户自己建立数据类型 |

|  |  |
| --- | --- |
| **学院** | 生命科学学院 |
| **专业** | 生物信息学 |
| **班级** | 生信1801 |
| **学号** | 8305180710 |
| **姓名** | 张彦哲 |
| **实验日期** | 2019年11月14日 |
| **指导教师** | 彭小清 |

1. **编译环境：**

*//!encoding: UTF-8  
//IDE: Clion 2019.2.2*

*//CMake: Version 3.14.5  
//Architecture: Windows 10 SDK version 1803  
//Author: Sean Peldom Zhang  
//Create Time: 2019/11/14*

1. **作业要求**：

任选教材第9章课后1道题，完成并打印或手抄，要求每道题使用指针。

1. **第9章第10题**

题目：有a，b两个非空链表，每个链表包含学号，成绩，将其合并，并按照学号升序排列

思路：我先生成a链表，假设其生成时就是升序排列的，然后假设b是随机的，把b逐一抽取，遍历插入a，输出；实际上有点武断，应该对a先排序的

源代码(171行)：

#include **<stdio.h>**#include **<stdlib.h>**#include **<assert.h>**#include **<time.h>****unsigned int** st\_num\_a,st\_num\_b;  
**void** initiate\_a\_b(**void**){*//随机生成a，b，分别为2个班的人数*  
 srand((**unsigned**)(time(**NULL**)));  
 st\_num\_a=rand()%28+3;  
 st\_num\_b=rand()%28+3;*//each class has 3 to 30 students;*}  
**struct** scores{  
 **int** Physics;  
 **int** Chemistry;  
 **int** Biology;  
};  
**typedef struct** node{  
 **unsigned int** Student\_ID;*//a班全为2n-1，b班全为2n，即使a，b应该不同也不会重复* **char** \*name;*//重名没关系，反正不是排序依据* **struct** scores subject;  
 **struct** node \*next\_node;  
}Student\_node;  
**struct** names{*//生成一个名字库，指针随机指*  
 **char** name[10];  
}rand\_names[30]={**"William"**,**"Daniel"**,**"Olivia"**,**"James"**,**"Noah"**,  
 **"Sophia"**,**"Madison"**,**"Dylan"**,**"Tyler"**,**"Eliza"**,  
 **"Sarah"**,**"Brandon"**,**"Jessica"**,**"Angel"**,**"Lily"**,  
 **"Ira"**,**"Gavin"**,**"Carter"**,**"Alex"**,**"Sofia"**,  
 **"Addison"**,**"Jake"**,**"Aaron"**,**"Whitney"**,**"Joshua"**,  
 **"Joyce"**,**"Park"**,**"Alita"**,**"Hugo"**,**"Victor"**};  
**void** print\_each\_score\_type\_standard(**unsigned int** Student\_ID,**char** \*name,**int** Physics,**int** Chemistry,**int** Biology){  
 printf\_s(**"%u\t%s\t%d\t%d\t%d\n"**,Student\_ID,name,Physics,Chemistry,Biology);  
}  
**void** puts\_line(){  
 puts(**"==========================================================="**);  
}  
Student\_node \*create\_linked\_lists\_a(**unsigned int** a) {  
 **assert**(a >= 3);  
 puts\_line();  
 puts(**"Class A:\nSt\_ID\tSt\_name\tPhysics\tChem\tBiology"**);  
 Student\_node \*head\_a = (Student\_node \*) malloc(**sizeof**(Student\_node));*//a头节点指针* head\_a->Student\_ID = 1;*//初始化a班头节点* head\_a->name=rand\_names[rand()%30].name;*//0~29* head\_a->subject.Physics = rand() % 100 + 1;  
 head\_a->subject.Chemistry = rand() % 100 + 1;  
 head\_a->subject.Biology = rand() % 100 + 1;  
 head\_a->next\_node = **NULL**;  
 print\_each\_score\_type\_standard(head\_a->Student\_ID,head\_a->name,head\_a->subject.Physics,head\_a->subject.Chemistry,head\_a->subject.Biology);  
 Student\_node \*Cycle\_point\_a = head\_a;  
 **for** (**unsigned int** i = 2; i <= a - 1; ++i) {*//初始化a班的中间节点* Student\_node \*body\_a = (Student\_node \*) malloc(**sizeof**(Student\_node));  
 body\_a->Student\_ID = 2\*i-1;  
 body\_a->name=rand\_names[rand()%30].name;  
 body\_a->subject.Physics = rand() % 100 + 1;  
 body\_a->subject.Chemistry = rand() % 100 + 1;  
 body\_a->subject.Biology = rand() % 100 + 1;  
 body\_a->next\_node = **NULL**;  
 print\_each\_score\_type\_standard(body\_a->Student\_ID,body\_a->name,body\_a->subject.Physics,body\_a->subject.Chemistry,body\_a->subject.Biology);  
 Cycle\_point\_a->next\_node = body\_a;  
 Cycle\_point\_a = Cycle\_point\_a->next\_node;  
 }  
 Student\_node \*tail\_a = (Student\_node \*) malloc(**sizeof**(Student\_node));  
 tail\_a->Student\_ID = 2\*a-1;*//初始化a班尾节点* tail\_a->name=rand\_names[rand()%30].name;  
 tail\_a->subject.Physics = rand() % 100 + 1;  
 tail\_a->subject.Chemistry = rand() % 100 + 1;  
 tail\_a->subject.Biology = rand() % 100 + 1;  
 tail\_a->next\_node = **NULL**;  
 print\_each\_score\_type\_standard(tail\_a->Student\_ID,tail\_a->name,tail\_a->subject.Physics,tail\_a->subject.Chemistry,tail\_a->subject.Biology);  
 Cycle\_point\_a->next\_node=tail\_a;  
 puts\_line();  
 **return** head\_a;  
}  
Student\_node \*create\_linked\_lists\_b(**unsigned int** b) {  
 **assert**(b>=3);  
 puts\_line();  
 puts(**"Class B:\nSt\_name\tSt\_ID\tPhysics\tChem\tBiology"**);  
 Student\_node \*head\_b=(Student\_node\*)malloc(**sizeof**(Student\_node));*//b头节点指针* head\_b->Student\_ID=2;*//初始化b班头节点* head\_b->name=rand\_names[rand()%30].name;*//0~29* head\_b->subject.Physics=rand()%100+1;  
 head\_b->subject.Chemistry=rand()%100+1;  
 head\_b->subject.Biology=rand()%100+1;  
 head\_b->next\_node=**NULL**;  
print\_each\_score\_type\_standard(head\_b->Student\_ID,head\_b->name,head\_b->subject.Physics,head\_b->subject.Chemistry,head\_b->subject.Biology);  
 Student\_node \*Cycle\_point\_b=head\_b;  
 **for** (**unsigned int** i = 2; i <= b-1; ++i) {*//初始化b班中间节点* Student\_node \*body\_b=(Student\_node\*)malloc(**sizeof**(Student\_node));  
 body\_b->Student\_ID=i\*2;  
 body\_b->next\_node=**NULL**;  
 body\_b->name=rand\_names[rand()%30].name;  
 body\_b->subject.Physics=rand()%100+1;  
 body\_b->subject.Chemistry=rand()%100+1;  
 body\_b->subject.Biology=rand()%100+1;print\_each\_score\_type\_standard(body\_b->Student\_ID,body\_b->name,body\_b->subject.Physics,body\_b->subject.Chemistry,body\_b->subject.Biology);  
 Cycle\_point\_b->next\_node=body\_b;  
 Cycle\_point\_b=Cycle\_point\_b->next\_node;  
 }  
 Student\_node \*tail\_b = (Student\_node \*) malloc(**sizeof**(Student\_node));  
 tail\_b->Student\_ID = b\*2;*//初始化b班尾节点* tail\_b->name=rand\_names[rand()%30].name;  
 tail\_b->subject.Physics = rand() % 100 + 1;  
 tail\_b->subject.Chemistry = rand() % 100 + 1;  
 tail\_b->subject.Biology = rand() % 100 + 1;  
 tail\_b->next\_node = **NULL**;  
print\_each\_score\_type\_standard(tail\_b->Student\_ID,tail\_b->name,tail\_b->subject.Physics,tail\_b->subject.Chemistry,tail\_b->subject.Biology);  
 Cycle\_point\_b->next\_node=tail\_b;  
 puts\_line();  
 **return** head\_b;  
}  
Student\_node \*integrate(Student\_node \*head\_a,Student\_node \*head\_b){*//假设a是升序排序好的，b不是，把b逐一插入a中* **assert**(head\_a!=NULL && head\_b!=NULL);*//假设ab都不是空表* Student\_node \*cycle\_b=head\_b,\*cycle\_a=head\_a;  
 **for** (**unsigned int** j=1;j<=st\_num\_b;j++){*//循环b次，插入所有节点* head\_b=head\_b->next\_node;*//现在仅有cycle\_b知道原head，等同于剪下了head，定义新的head,head\_b最后会变成NULL* printf\_s(**"cycle\_b%d\n"**,cycle\_b->Student\_ID);  
 **if**(head\_a->Student\_ID>cycle\_b->Student\_ID){*//如果b比a的第一个还小，把b里来的插入在a的头，理论上这一步不会发生* head\_a->next\_node=cycle\_b;  
 cycle\_b->next\_node=cycle\_a->next\_node;  
 printf\_s(**"before\n"**);  
 *// goto insert\_jump;//跳过中间* }**else**{  
 **while**(cycle\_a->next\_node!=**NULL** && cycle\_a->Student\_ID < cycle\_b->Student\_ID){*//cycle\_a不是最后1个节点  
 //插在1~n-1中* **if**(cycle\_a->next\_node->Student\_ID>cycle\_b->Student\_ID){*//在中间插入* cycle\_b->next\_node=cycle\_a->next\_node;*//先插后* cycle\_a->next\_node=cycle\_b;*//再插前*printf\_s(**"a%d,b%d,a%d\n"**,cycle\_a->Student\_ID,cycle\_b->Student\_ID,cycle\_b->next\_node->Student\_ID);  
 }  
 printf\_s(**"%d "**,cycle\_a->Student\_ID);  
 cycle\_a=cycle\_a->next\_node;*//下1个，遍历* **if**(cycle\_a->next\_node->next\_node==**NULL** && cycle\_a->next\_node->Student\_ID < cycle\_b->Student\_ID){*//如果停在n-1处且n小于其，需要插在n后* cycle\_a->next\_node->next\_node=cycle\_b;  
 cycle\_b->next\_node=**NULL**;  
 puts(**"jump\n"**);  
 **goto** insert\_jump;*//跳到最后* }  
 }  
 }  
 insert\_jump:  
 cycle\_b=head\_b;*//初始化cycle\_b* cycle\_a=head\_a;*//初始化cycle\_a* }  
 puts(**""**);  
 **return** head\_a;  
}  
**void** print\_integrated\_linked\_lists(Student\_node \*head\_integrate){  
 Student\_node \*cycle\_itg=head\_integrate;  
 puts\_line();  
 puts(**"Class A+B:\nSt\_ID\tSt\_name\tPhysics\tChem\tBiology"**);  
 **for** (**unsigned int** i = 0; i < st\_num\_a+st\_num\_b; ++i) {  
 print\_each\_score\_type\_standard(cycle\_itg->Student\_ID,cycle\_itg->name,cycle\_itg->subject.Physics,cycle\_itg->subject.Chemistry,cycle\_itg->subject.Biology);  
 **if**(cycle\_itg->next\_node!=**NULL**)cycle\_itg=cycle\_itg->next\_node;  
 }  
 puts\_line();  
}  
**int** main(**void**){  
 puts(**"This is a program that integrate 2 classes final exam scores."**);  
 initiate\_a\_b();  
 printf\_s(**"Student number in class A: %d\nStudent number in class B: %d\n"**,st\_num\_a,st\_num\_b);  
 Student\_node \*head\_a=create\_linked\_lists\_a(st\_num\_a);  
 Student\_node \*head\_b=create\_linked\_lists\_b(st\_num\_b);  
 puts(**"Integrating..."**);  
 Student\_node \*head\_integrate=integrate(head\_a,head\_b);  
 print\_integrated\_linked\_lists(head\_integrate);  
 **return** 0;  
}

程序运行结果：

This is a program that integrate 2 classes final exam scores.

Student number in class A: 9

Student number in class B: 15

===========================================================

Class A:

St\_ID St\_name Physics Chem Biology

1 William 7 59 59

3 Eliza 83 17 55

5 Hugo 30 40 45

7 Hugo 75 71 17

9 Victor 80 14 93

11 Eliza 9 98 27

13 Whitney 59 16 39

15 Brandon 59 1 36

17 Victor 90 24 71

===========================================================

===========================================================

Class B:

St\_name St\_ID Physics Chem Biology

2 Joyce 79 68 51

4 Daniel 31 79 19

6 Jessica 28 71 85

8 Angel 81 61 9

10 Jake 57 100 6

12 Aaron 87 51 2

14 Olivia 29 81 7

16 Ira 27 7 76

18 Aaron 53 11 58

20 William 83 20 93

22 Daniel 38 34 2

24 William 92 43 34

26 Dylan 88 62 87

28 James 62 75 97

30 Alex 7 11 24

===========================================================

Integrating...

cycle\_b2

a1,b2,a3

1 cycle\_b4

1 2 a3,b4,a5

3 cycle\_b6

1 2 3 4 a5,b6,a7

5 cycle\_b8

1 2 3 4 5 6 a7,b8,a9

7 cycle\_b10

1 2 3 4 5 6 7 8 a9,b10,a11

9 cycle\_b12

1 2 3 4 5 6 7 8 9 10 a11,b12,a13

11 cycle\_b14

1 2 3 4 5 6 7 8 9 10 11 12 a13,b14,a15

13 cycle\_b16

1 2 3 4 5 6 7 8 9 10 11 12 13 14 a15,b16,a17

15 cycle\_b18

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 jump

cycle\_b20

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 jump

cycle\_b22

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 jump

cycle\_b24

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 jump

cycle\_b26

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 jump

cycle\_b28

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 jump

cycle\_b30

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 22 24 jump

===========================================================

Class A+B:

St\_ID St\_name Physics Chem Biology

1 William 7 59 59

2 Joyce 79 68 51

3 Eliza 83 17 55

4 Daniel 31 79 19

5 Hugo 30 40 45

6 Jessica 28 71 85

7 Hugo 75 71 17

8 Angel 81 61 9

9 Victor 80 14 93

10 Jake 57 100 6

11 Eliza 9 98 27

12 Aaron 87 51 2

13 Whitney 59 16 39

14 Olivia 29 81 7

15 Brandon 59 1 36

16 Ira 27 7 76

17 Victor 90 24 71

18 Aaron 53 11 58

20 William 83 20 93

22 Daniel 38 34 2

24 William 92 43 34

26 Dylan 88 62 87

28 James 62 75 97

30 Alex 7 11 24

===========================================================

进程已结束，退出代码 0