测试作业

1 有一个学生结构体，其数据成员有：学号，姓名，3 门课程。从键盘上输入5 个学生的信息。要求输出：

1002 lilei 98 98 100

1004 hanmeimei 87 88 87

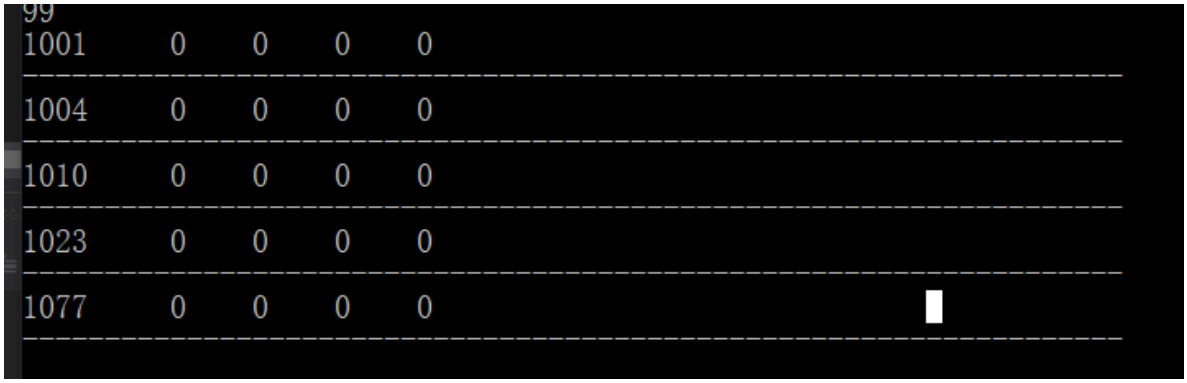
1008 lihua 85 85 89

1016 yiyi 97 97 82

1032 chem 100 98 71

(1) 按照学号递增输出全部学生信息，每个学生的信息一行。（格式：学号姓名分数1 分数2 分数3 总分）

学号按照顺序排了但是出来之后成绩全是0000，一定是我打印函数出问题了。



(2) 输出每门课程最高分的学生的信息

(3) 输出每门课程的平均分

(4) 按照总分输出学生排名

- 主函数
  - ```
    #include"myLibrary.h" //主函数
    int main()
    {
        Student_t sArr[5];
        int a[5] = { 0 };
        pStudent_t pHead = NULL;
        pStudent_t pTail = NULL;

        for (int i = 0; i < 5; i++)
        {
            cin >> sArr[i].num >> sArr[i].name >> sArr[i].cham >> sArr[i].biology
                >> sArr[i].phy;
    ```
    ```

```
        /*scanf_s("%d %s %d %d %d", &sArr[i].num, sArr[i].name, &sArr[i].cham, &sArr[i].biology,
            &sArr[i].phy);*/
        sArr[i].score = sArr[i].cham + sArr[i].biology, +sArr[i].phy;
        numSert(&pHead, &pTail, sArr[i]);
    }
    ListPrint(pHead);
}
```

- 头文件

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
using namespace std;
typedef struct student {
    int num; //学号
    char name[20];
    float cham;
    float biology;
    float phy;
    float score;//总成绩
    struct student* pNext; //链表指针
}Student_t, * pStudent_t;
void numSert(pStudent_t* ppHead, Student_t** ppTail, Student_t val);//
void ListPrint(pStudent_t pHead);
```

- myLibrary.h

```
void numSert(pStudent_t* ppHead, Student_t** ppTail, Student_t val) //头插法实现按照学号建立链表
{
    pStudent_t pCur,pPre;
    pStudent_t pNew = (pStudent_t)calloc(1, sizeof(Student_t));
    pCur = *ppHead;
    pPre = *ppHead;
    pNew->num = val.num;
    if (*ppHead == NULL) //判空
    {
        *ppHead = pNew;
        *ppTail = pNew;
    }
    else if((val.num)<(pCur->num)) //头插法
    {
            pNew->pNext = pCur;
            *ppHead = pNew;
    }
    else
    {
        while (pCur)
        {
            if (pCur->num > val.num)
            {
                pPre->pNext = pNew;
```

```
                pNew->pNext = pCur;
                break;
            }
            pPre = pCur;
            pCur = pCur->pNext;
        }
        if(NULL==pCur) //尾插法
        {
            pPre->pNext = pNew;
            *ppTail = pNew;
        }
    }
}
void ListPrint(pStudent_t pHead) //依次打印
{
    pStudent_t p = pHead;
    while (p != NULL)
    {

        printf("%4d %s %4f %4f %4f %4f\n", p->num, p->name, p->cham,
            p->biology, p->phy, p->score);
        cout << "-------------------------------------------------------------" << endl;
        p = p->pNext;
    }
}
```

2 用尾插法建立链表

- ```
  void ListTailInSert(pStudent_t* ppHead, Student_t** ppTail, int val) //尾插法
  {
      pStudent_t pNew = (pStudent_t)calloc(1, sizeof(Student_t));
      pNew->num = val;
      if (NULL == *ppTail) //判空 如果空则直接将尾指针和头指针指向new；
      {
          *ppHead = pNew;
          *ppTail = pNew;
      }
      else
      {
          (*ppTail)->pNext = pNew; //原尾指针的Next指向pNew
          *ppTail = pNew;          //修改尾指针为pNew
      }
  }
  ```

3 用有序插入建立链表

- ```
  void numSert(pStudent_t* ppHead, Student_t** ppTail, Student_t val)
  {
      pStudent_t pCur,pPre;
      pStudent_t pNew = (pStudent_t)calloc(1, sizeof(Student_t));
      pCur = *ppHead;
      pPre = *ppHead;
      pNew->num = val.num;
  ```

```
    if (*ppHead == NULL) //判空
    {
        *ppHead = pNew;
        *ppTail = pNew;
    }
    else if((val.num)<(pCur->num)) //头插法
    {
            pNew->pNext = pCur;
            *ppHead = pNew;
    }
    else //非第一个则
    {
        while (pCur) //中间段
        {
            if (pCur->num > val.num)
            {
                pPre->pNext = pNew;
                pNew->pNext = pCur;
                break;
            }
            pPre = pCur;
            pCur = pCur->pNext;
        }
        if(NULL==pCur) //末尾则用尾插法
        {
            pPre->pNext = pNew;
            *ppTail = pNew;
        }
    }
}
```

-