实验一 结构体与动态内存

题目：使用结构保存学生的信息：学号、姓名、专业、平均分。然后编写菜单，实现学生信息的录入功能和显示功能。在录入学生信息时，学生数量未知，因此要实现任意多学生的录入。

（1）使用 new 和 delete 运算符申请和回收堆内存；

（2）使用 c++的输入输出流。

编程思路：

仍然使用面向过程的编程思路。

（1） 根据题目，明显需要设计一个结构体保存学生信息；为了保存学生信息，需要设计一个变量。由于是一组学生，所以需要使用数组或链表。由于学生数量未知，所以如果使用数组的话，需要动态改变数组的大小；如果频繁改变数组的大小，程序的效率会比较低；如果数组过大，则会浪费很多空间。所以需要选择合适的大小。

（2） 我们使用动态数组来保存数据。结构体定义如下：

```
struct STUDENT
{
    long int number;
    char name[20];
    char major[20];
    double score;
};
```

数据通过指针变量 gStu 存储到数组中，gStu 的定义如下：

```
STUDENT * gStu;
```

还需要一个变量记录 gStu 指向的数组中保存了多少个学生的信息，该变量定义为：

```
int count;
```

还需要有一个变量指示 gStu 指向的数组中还有多少个空间可用，该变量定义为：

```
int available;
```

在 gStu 指向的空间用尽后，需要增加更多的空间。此时，每次增加的空间大小可以为常数，因此可用一个宏定义来指明这个大小（这里取值为 10）：

```
#define  BLOCK 10
```

还需要设计一个增加学生信息的函数 add_student 和显示学生信息的函数 display，原型如下（注意：在 add_student 函数的参数中使用了引用类型）：

```
void add_student(STUDENT & stu);
void display();
```

（3） 上述定义应在头文件中，其实现应在实现文件中。假设实现该程序的学生的学号为 123456，则根据我们的程序规范，头文件 123456_2.h 定义如下：

```
#ifndef _123456_2_H_
#define _123456_2_H_

#include <iostream>
using namespace std;

namespace N123456
{
  #define BLOCK 10

  struct STUDENT
  {
      long int number;
      char name[20];
      char major[20];
      double score;
  };

  extern STUDENT * gStu;
  extern int count;//已经加入的学生个数
  extern int available;//可用的空间
  void add_student(STUDENT & stu);
  void display();
}

#endif
```

上面的程序中，gStu、count 和 available 仅使用 extern 做了声明，而没有定义。它们的定义应该在实现文件中。这样做的目的是为了防止该头文件被包含到多个实现文件中时出现变量重定义的错误。

（4） 在实现文件 123456_2.cpp 中先定义变量，内容如下：

```
#include "123456_2.h"

namespace N123456
{
    STUDENT * gStu = NULL;
    int count = 0;
    int available = 0;
}
```

在程序开始运行时，由于没有学生信息，所以 gStu 初始化为空指针；count 初始化

为 0，表示 gStu 所指向的空间中没有学生信息；available 初始化为 0，表示 gStu 所指向的空间中没有多余的可用空间。

（5） 实现添加学生信息的函数 add_student。在这个函数中，首先检查是否有可用空间，如果没有，则扩充空间；如果有则将学生信息添加到第一个可用的空间中。在执行上述任务之后要调整 count 和（或）available 的值。参考实现如下（要将其实现在文件 123456_2.cpp 中的名字空间 N123456 中）：

```
void add_student(STUDENT &stu)
{
    if (0 == available)
    {//如果没有空间，需先扩展空间
        STUDENT * tmp = new STUDENT[count + BLOCK];
        for (int i = 0; i < count; i++)
        {
            tmp[i].number = gStu[i].number;
            strcpy_s(tmp[i].name, 20, gStu[i].name);
            strcpy_s(tmp[i].major, 20, gStu[i].major);
            tmp[i].score = gStu[i].score;
        }
        delete[] gStu;
        gStu = tmp;
        available = BLOCK;
    }
    //增加学生信息
    gStu[count].number = stu.number;
    strcpy_s(gStu[count].name, stu.name);
    strcpy_s(gStu[count].major, stu.major);
    gStu[count].score = stu.score;

    count++;
    available--;
}
```

（6） 实现显示学生的信息函数 display。这个函数比较简单。同样，要将其实现在 123456_2.cpp 文件中的名字空间 N123456 中。参考实现如下：

```
void display()
{
    cout << "学号\t姓名\t专业\t成绩" << endl;
    for (int i = 0; i<count; i++)
    {
        cout << gStu[i].number << "\t" << gStu[i].name << "\t"
            << gStu[i].major << "\t" << gStu[i].score << "\n";
```

```
        }
    }
```

（7） 最后实现 main 函数，完成一个菜单，供用户选择使用哪个功能。该函数实现在文件 123456_1.cpp 中，参考实现如下：

```
#include "123456_2.h"
using namespace N123456;
int main()
{
    char choice = '\0';
    STUDENT s;
    while (choice != '0')
    {
        cout << "学生信息管理系统\n";
        cout << "1 录入学生信息 \n";
        cout << "2 显示学生信息 \n";
        cout << "0 退出系统 \n";
        cout << "请选择所需要的操作：";
        cin >> choice;

        switch (choice)
        {
        case '1':
            cout << "请依次输入学号、姓名、专业和成绩\n";
            cin >> s.number >> s.name >> s.major >> s.score;
            add_student(s);
            cout << endl;
            break;
        case '2':
            display();
            cout << endl;
            break;
        case '0':
            break;
        default:
            cout << "你的选择有误，请重新选择！" << endl;
        }
    }
    return 0;
}
```

（8） 这是一个简单的程序，练习动态内存的使用和 C++标准输入输出流的简单使用。对于程序功能来说，可自行扩充一些，比如删除学生的功能（此时涉及到数组空间的减少）。这里不再多述。