

项目

总结

- 运算符 != 为不等于, *为取值符
- #define N 100 为用100来替换n, 末尾不用加分号
- define用法 用宏定义的宏名来表示一个字符串, 在宏展开时又用字符串来取代宏名
- void * 为"无类型指针"
- memset初始化
- switch分支语句用来表示一种函数的所有值,

【注意 常量表达式的值不能改变】

- 把表达式都通过case列举出来,
- 表达式的值与case后的常量表达式进行比较
- 与case后的常量表达式匹配则继续执行下一个case, 直到break
- 若都不匹配则执行default后的语句

```
语法:
switch(表达式)
{
    case 常量表达式1:
        语句1;
        break;
    case 常量表达式2:
        语句2;
        break;
    .....
    case 常量表达式m:
        语句m;
        break;
    default:    //其它
        语句n;
        break;
```

- C 库函数strcpy

```
**char *strcpy(char *dest, const char *src)**
```

把 **src** 所指向的字符串【复制】到 **dest**。

- 制表符 为 '\t'
- head_print(); 提示
- init_sys(); 初始化
- manage_sys(); 进入系统
- |为 位或, &为位与, !为非
- void init 为声明函数

- 一维数组 数据类型 数据名称 [0] 从0开始
- 二维数组 数据类型 数组名[行下标] [列下标]
 - 例子int a[3] [4]表示a是一个具有三行，每行有四个int的数组
 - 其中a[1] [2]表示a这个数组第一行第二个元素
- 逻辑
 - 先头文件
 - 加一个字符串，是数组能使用
 - 给n下定义为了。。。。。。。。。。
 - 分别给学生名字，语数英定义并赋值
 - 定义一个系统数据的条数，并赋值，便于之后系统执行
 - 加一个输出函数的头指针进行输出操作，便于管理
 - 输出各个量，进行解释说明（）
 - 对函数进行初始化，否则输出函数可能太大
 - 需要列举所有情况加一个函数，管理函数用while循环，switch列举数据需要变化的所有情况，这种更方便
 - 对所有需要插入的函数进行一个输入和输出，便于检查？
 - 开始设置执行程序
 - 删除数据，若删除第n个数，则name n=n+1
 - 修改数据，直接打印
 - 查询数据，设置一个表，以供查询
 - 插入一个进入系统的函数

问题

- #define N 100 ，如果只是为了定义是不是可以int n=100
这句话的含义不是用100来替换N，那下面出现的N都代表100？
- 学生名字为什么用一个二维数组

```

9
10 #include<stdio.h>
11 #include<string.h>
12 #define N 100//define为
13 // 学生名字
14 char stu_name[256][N]; //name有256行，每行N个char型的数
15 // 英文成绩
16 int stu_en_score[N];
17 // 语文成绩
18 int stu_ch_score[N];
19 // 数学成绩
20 int stu_ma_score[N];
21
22 // 记录系统的数据条数
23 int num =0;
24
25 void head_print()
26 {
27     printf("*****学生成绩管理系统*****\n");
28     printf("*      请输入以下按键:      *\n");
29     printf("*      1、增加学生成绩      *\n");
30     printf("*      2、删除学生成绩      *\n");

```

- a[0] [2]不应该为3吗

```

~
a[2][3] = 0;
(3):部分赋值:按行部分赋值
int a[3][4] = {
    {1,2},
    {3},
    {4,5}
};
a[0][0] = 1;
a[0][2] = 0;
a[2][1] = 5;

int a[3][4] = {
    {1,2},
    {},
    {4,5}
};
(4). 全行赋值时 可以省略行数 ***但不能省略列数

```

- print函数的返回值为return，因为是void类型？printf的返回值为return 0？