Basic things

变量交换

三变量法

```
t = a;
a = b;
b = t;
```

加减法

```
a = a + b;
b = a - b;
a = a - b;
```

异或法

```
a ^= b ^= a ^= b;
```

C语言...

在这部分内容的学习中，首先熟悉C语言，具体地说是C99，虽然有出现的C11，但新的特性没有太多影响（C11移除了gets)。在基本部分的内容全部可以用 `gcc -std=c99` 编译通过，在后续部分将会学习掌握C++的一些特性。

多数情况下C语言所指为 `ANSI C`，即C89而并非C99，所以完成的这些C语言程序可以作为C++程序提交。

由于为了兼容C99产生的一些问题，比如C99中double的输出必须用%f，输入用%lf，而对于C89和C++则都可以用%lf。

数据类型的大小

C99中规定了int整数至少是16位，却没有规定具体值，因此同时有一些固定长度的整数，如int32_t，uint32_t。

long long在Linux下的输入输出格式符为%lld，但Windows平台有时为%I64d。为保险起见，可以用C++的流。

time.h

使用计时函数 `clock()` 返回程序目前为止运行的时间，除以常数 `CLOCKS_PER_SEC` 得到以“秒”为单位的时间。

由于键盘输入的时间也会计算在内，在只需要运行时间是可以使用命令行的管道“|”或使用读文件的方式近似忽略读取时间。

输入输出框架

```
while(scanf("%d", &x) == 1);
```

以上为最常见的循环读取单个数据的输入形式。

在输入完成后，因为回车符是不会被读取的，在Windows下首先回车后按下 `Ctrl+Z` 再次回车结束，在Linux下则使用 `Ctrl+D` 即可。

本地进行测试时使用文件输入输出是更好的选择。下面是两种读写文件的方法。

输入输出重定向

```
freopen("input.txt", "r", stdin);
freopen("output.txt", "w",stdout);
```

这是一种较简单的办法，但一些情况下使用输入输出重定向是不被允许的，需要仔细了解具体规则。

下面是一种只在本机测试时使用文件重定向的方法。

```
#define LOCAL
....
#ifdef LOCAL
freopen("input.txt", "r", stdin);
freopen("output.txt", "w",stdout);
#endif
....
```

这样在程序提交时就可以删除 `#define LOCAL` 或更好的方法是在编译选项中定义LOCAL符号。

fopen

```
....
FILE *fin, *fout;
fin = fopen("data.in", "rb");
fout = fopen("data.out", "wb");
....
while(fscanf(fin, "%d", &x) == 1);
....
fprintf(fout, "%d %d %.3f\n", min, max, (double)s/n);
fclose(fin);
fclose(fout);
....
```

重定向和fopen两种方法各有优劣。重定向的方法写起来较简单，但是不能同时读写文件和标准输入输出。fopen写法较繁琐，但是灵活性较好。

也可以把fopen版的程序改成读写标准输入输出只需赋值 `fin = stdin; fout = stdout;`。

数组

C语言中数组不能够进行赋值操作，比如“`int a[maxn],b[maxn]`”不能赋值`b = a`，如果需要从数组a复制k个元素到数组b，可以这样；

```
memcpy(b,a,sizeof(int) * k);
```

如果需要复制全部，可以这样；

```
memcpy(b,a,sizeof(a));
```

使用`memcpy()`函数需要包含头文件`string.h`。

基本问题

开灯问题

用初始化为零的整型数组表示灯的状态，两层循环分别遍历人数和灯泡编号，编号整除对应人数时灯的状态取反。

蛇形填数

使用二维数组存储数据，分两步，首先以右上角为起点按照规律用两层循环向下，向左，向上，向右...分配数字，外层循环条件至分配完所有数字，内层循环两个判断条件，先判断正在移动的角标是否越界，再判断目标位置是否未分配数字；第二步按行打印。

竖式问题

使用 `scanf("%s",s)` 可以读入一个不含空格、TAB和回车符的字符串，存入字符数组s，注意这里并没有&。

输入为一数字集合，输出abc*de的竖式满足在完整竖式中所有数字来自指定的数字集合，两层遍历三位和两位的乘数可以获得积，为判断出现数字是否来自指定集合，使用 `sprintf()` 输出所有数字到一字符串，遍历此字符串，使用 `strchr()` 判别。

常用函数

string.h:

```
memcpy(b,a,sizeof(int) * k);//从数组a复制k个元素到数组b
memset(a,0,sizeof(a));//数组a设置初始值
strchr(s,ch);//在字符串s中查找单字符ch
strlen(s);//获取字符串s的实际长度
strcpy(a,b); //赋值
strcmp(a,b);//比较
strcat(a,b);//连接
```

stdio.h:

```
sprintf();//输出到字符串
fprintf();//输出到文件
fgetc(fin);//从文件流读取一个字符
fgetc(stdin);
getchar();//从标准输入流读取一个字符
fgets(buf,maxn,fin);//读取不超过maxn-1个字符，在末尾添上结束符\0，读取时遇到\n停止，可以用来读取一行字符
gets();//存在缓冲区溢出漏洞，C11已移除;
```

→back

Contact me

Please contact me about any suggestion(s): *aliceb0b@hotmail.com*.