

循环递推传递数据：当一个循环中的数据要跟上一次循环中的一个或多个数据产生联系时，要注意多个数据之间的传递。

图表的打印：采用双层循环，外层用总行数 n 来控制行号 i ，表示一共要打印几行，内层用 n 和 i 来表示 j ，表示每行打印多少个指定字符或打印单元。

由于外层的 i 是具体行号，所以只需要倒序遍历，就可以让图案倒置。打印时可以根据具体情况限制打印内容。

随机数：

头文件 `#include <stdlib.h>`，标准库头文件

设置随机数种子：srand()函数，在小括号中传入一个作为种子的数字，一般传入 `time(NULL)`，并需要额外的 `#include <time.h>` 头文件

取出随机数：一般使用rand()函数，但是这个函数范围较大，通常用 `rand() % n`来取出0~ $n-1$ 之间的随机数。

如果要得到 $m \sim n$ 之间的随机数，可以让 `rand() % (n - m + 1)`，取出0~($n-m$)的随机数后，加上 m ，就可以得到 $m \sim n$ 之间的随机数。

在上一条的基础上除以一个数字，就可以在某个小数范围内取出随机数。

预编译指令：也叫预编译语句，主要包括以下几种——

#include：

用法：后面可以跟`<>`或者`"`，`<>`表示引用系统库提供的头文件，`"`表示优先查找自定义的头文件，如果找不到再去系统库里找。

作用：预处理时，将该头文件的所有内容递归展开在指令调用位置。

#define：

例1： `#define ARR_SIZE 10`

用法：后面跟两个字段，第一个字段是命名字段（ARR_SIZE），第二个字段为内容字段（10）。

作用：查找所有字段1的位置（大小写敏感、全字匹配、跳过字符串），将其替换为字段2。

例2： `#define MAX(a,b) ((a) > (b) ? (a) : (b))`

用法：可以传入若干参数，在内容字段可以使用这些参数。

作用：传入参数后，先替换参数，例如传入 m 和 n ，则将内容中的 a 替换为 m ， b 替换为 n ，然后将式子整体替换回去。但注意这种情况下直接粗暴的查找替换可能会导致优先级混乱，所以要带括号。

注意，如果一个宏占用多行，行末要加反斜杠`\`，表示下一行也属于这个宏。

#if/#elif/#else/#endif：

用法：if或elif后面可以跟一个常数或者defined表达式，用法类似if/else if/else架构。多个defined表达式间可以用逻辑运算符中的与或非表示。

作用：常用于调试，通过判断if、elif后面是不是0，可以选择不同的代码块进行编译。是一种简单的多分支调试或者注释代码的手段。

#ifdef/#ifndef:

用法：后面跟一个字段。

作用：这个字段如果被#define定义过，则ifdef为真，ifndef为假，反之亦然。

一维数组:

一维数组是一种C语言/C++的数据类型，可以定义多个相同类型的变量，他们的空间是连续的。

一维数组初始化时，如果没有将所有空间都初始化，则没有初始化的空间系统会自动初始化成0。

一维数组可以省略中括号中的数字，但是这么做的话一定要初始化，系统会根据初始化元素的个数决定数组的大小。

数组遍历:

循环头: `for(i = 0; i < n; i++)`

通过 `arr[i]` 可以遍历长度为n的数组arr的每一个元素。

通过 `sizeof(a) / sizeof(a[0])` 的方式可以求出数组的元素个数。

中括号也是一个运算符，它是一个单目运算符（右）。

——定义变量时的中括号不是运算符。

——这个单目运算符是一个有辅助信息（下标）的运算符。

——这个单目运算符表达式的值是一个变量。

卡池：用于取出若干个互不相同的随机数。方法是先把待取的所有数字放到一个数组里，然后随机生成下标去取，取走之后对应下标的位置填入-1，去取的时候先判断这个数字是不是已经被取走了（值是否为-1）如果没有则取走，如果已经被取走，则让下标增大1之后再去取，如果取到了数组的末尾，则重新打到开头继续取。