

实验报告 学习输出格式和<math.h>

--- c 语言程序设计

姓名：蔡云童

学号：15331007

1. 对 c 语言输出格式的学习和理解

一般格式：printf ("%d",n);

Printf()在<stdio.h>头文件内定义，用于向终端（显示器、控制台等）输出字符。格式控制由要输出的文字和数据格式说明组成。要输出的文字除了可以使用字母、数字、空格和一些数字符号以外，还可以使用一些转义字符表示特殊的含义。

范例 1:

```
#include <stdio.h>

int main(){
    printf("hello world!\n");
    return 0;
}
```

现在打出的是最简单的得 hello world，仅体现出 printf()的输出功能。

范例 2:

```
#include <stdio.h>

int main(){
    int n;
    n=1;
    printf("%d\n",n);
    return 0;
}
```

当然也可以同时输出多个变量的值，但其实本质是一样的。重要的事情是""内%d%d之间不可以加“,”，我这样错过很多次。

另外，还有比较重要的地方是**输出数据的类型**，这个对我来说很重要，但是有点难。目前比较常用的数值型分为整型和浮点型，整型包括：整型 int ,长整型 long ,短整型 short。而浮点型中包括：单精度浮点型 float 和双精度浮点型 double。

2. 对<math.h>的学习和理解

#include<math.h> 是包含 math 头文件的意思，math.h 头文件中声明了常用的一些数学运算，比如乘方，开方运算等等。

1、 三角函数

double sin(double);正弦

double cos(double);余弦

double tan(double);正切

2 、反三角函数

double asin (double); 结果介于[-PI/2,PI/2]

`double acos (double);` 结果介于 $[0, \pi]$

`double atan (double);` 反正切（主值），结果介于 $[-\pi/2, \pi/2]$

`double atan2 (double, double);` 反正切（整圆值），结果介于 $[-\pi, \pi]$

3 、双曲三角函数

`double sinh (double);`

`double cosh (double);`

`double tanh (double);`

4 、指数与对数

`double frexp(double value, int *exp);` 这是一个将 `value` 值拆分成小数部分 `f` 和（以 2 为底的）指数部分 `exp`，并返回小数部分 `f`，即 $f \cdot 2^{\text{exp}}$ 。其中 `f` 取值在 0.5~1.0 范围或者 0。

`double ldexp(double x, int exp);` 这个函数刚好跟上面那个 `frexp` 函数功能相反，它的返回值是 $x \cdot 2^{\text{exp}}$

`double modf(double value, double *iptr);` 拆分 `value` 值，返回它的小数部分，`iptr` 指向整数部分。

`double log (double);` 以 e 为底的对数

`double log10 (double);` 以 10 为底的对数

`double pow(double x, double y);` 计算 x 的 y 次幂

`float powf(float x, float y);` 功能与 `pow` 一致，只是输入与输出皆为浮点数

`double exp (double);` 求取自然数 e 的幂

`double sqrt (double);` 开平方

5 、取整

`double ceil (double);` 取上整，返回不比 x 小的最小整数

`double floor (double);` 取下整，返回不比 x 大的最大整数，即 [高斯函数](#) $[x]$

6 、绝对值

`int abs(int i);` 求整型的绝对值

`double fabs (double);` 求实型的绝对值

`double cabs(struct complex znum);` 求复数的绝对值

7 、标准化浮点数

`double frexp (double f, int *p);` 标准化浮点数， $f = x \cdot 2^p$ ，已知 `f` 求 `x, p` (x 介于 $[0.5, 1]$)

`double ldexp (double x, int p);` 与 `frexp` 相反，已知 `x, p` 求 `f`

8 、取整与取余

`double modf (double, double*);` 将参数的整数部分通过指针回传，返回小数部分

`double fmod (double, double);` 返回两参数相除的余数

9 、其他

`double hypot(double x, double y);` 已知直角三角形两个直角边长度，求斜边长度

`double ldexp(double x, int exponent);` 计算 $x \cdot (2^{\text{exponent}})$ 次幂

`double poly(double x, int degree, double coeffs []);` 计算多项式

大概就是这么多的数学函数，我也查到了一些注意事项，在这里就不赘述了。就我个人而言，`math.h` 的意义在于使用，还记得第一次做指数对数的实验题时百爪挠心百思不得其解的痛苦经历，那之后就把 `math.h` 放在心上了，起码现在如果遇到数学运算的编程问题，会直接跑去搜索 `math.h` 里面的函数，看有没有可以用的。

3. **心得体会**（一个月以来的学习感想，遇到的问题，学习疑问，相对 TA 说的话，等等，畅所欲言...）
- 心得体会：一个月来，我从对 c 语言完全不懂，畏惧恐慌，变得略知皮毛，但仍然有些畏惧恐慌……上第一堂程序设计课的时候，我撑着头听了三节课，感觉自己是不是选错了专业。不谦虚的说，整整三个 45 分钟，我听懂老师的话不超过 5 句。第一堂实验课，我对着 VS 打出“`hellow world`”的代码，心里又激动又惶恐，我明确的知道在我面前是另一个截然不同的世界，更精确，更冷静，更陌生，更充满机遇，也更充满挑战。每堂实验课我的时间都像是不够用，反复思索最简洁的程序，理清思路的过程痛苦极了，却也痛快极了。为什么一模一样的程序我的始终无法运行，“`;`”常常忘记加，中文字符与西文字符又弄混淆了……我不断地犯错，调试 `bug`，写程序的过程有多抓耳挠腮，程序最终运行的那一刻激动就有多真切。现在，虽然我不敢说我有擅长编程，但我能够感觉到自己对编程痛并快乐着的追逐。时刻准备着。时刻努力着。这就是我这一个月的学习感想。