

# 第一部分 语言篇

## 第一章 程序设计入门

- printf的用法
  - %d
  - %.nf 小数点后几位
- 在算法竞赛中，每行输出均应以回车符结束，包括最后一行。
- **注意考虑错误输入的情况**
- 短路(short-circuit)

C语言中的逻辑运算符都是短路运算符，一旦能确定整个表达式的值，就不再继续计算。
- sqrt(-10) 结果是 -nan(ind)

Nan是“Not a number”的缩写

ind是“indeterminate”，即无法确定的数
- 1.0/0.0 0.0/0.0 1/0 都会报错：被零除或对零取模

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int a,b;
6     scanf("%d%d",&a,&b);
7     printf("%d %d\n",a,b);
8     return 0;
9 }
```

输入：12 s a会正常读取 b无法读取 scanf返回值为0

利用下面的代码来验证上述的猜想

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int a=100,b=200;
6     int flag=100;
7     flag=scanf("%d%d",&a,&b);
8     printf("%d %d\n",a,b);
9     printf("%d\n",flag);
10 }
```

输入 12 s

输出结果为

```
1 12 200
2 1
```

证明在输入字符时，b并没有进行输入，保持原有的初始值200，

flag 为scanf()正确读取的字符数量 1。

- [如何用printf输出%d](#)

```
1 #include <stdio.h>
2 int main()
3 {
4     printf("%d\n");
5     getchar();
6     return 0;
7 }
```

```
1 #include <stdio.h>
2 int main()
3 {
4     printf("%c", '%');
5     printf("d\n");
6     getchar();
7     return 0;
8 }
```

```
1 #include <stdio.h>
2 int main()
3 {
4     printf("%sd\n", "");
5     getchar();
6     return 0;
7 }
```

思考问题：

1. int型整数的最小值和最大值是多少？

int占8字节共32位 其中有一位是符号位 最大值 $2^{31}-1=2147483647$  最小值为 $-2^{31}=-2147483648$

2. double型浮点数能精确到多少位小数？或者，这个问题本身值得商榷？

笔者认为这个问题值得商榷，因为笔者认为能精确到小数的位数，与有效数字的大小有关系

```
1 #include<stdio.h>
2 int main()
3 {
4     double a;
5     scanf("%lf",&a);
6     printf("%.20lf",a/3.0);
7 }
```

a输入为 1.0 时，输出结果为 0.33333333333333331000 精确到小数点后第16位。

a输入为 5.0 时，输出结果为1.66666666666666670000精确到小数点后第17位。

所以double的精确度应该在15和16位之间浮动。

3. double型浮点数最大正数值和最小正数值是多少？（不必特别精确）

最小正数值：化成二进制只有最后一位为0，所以最小值为 $2^{-1075}=2.22507e-308$

最大正数值：

IEEE 754规定, 超过 $2^{1024}$ 的数就会发生**正向溢出**, 记为 `Infinity(inf)`

4. 逻辑运算符“&&”、“||”和“!”(表示逻辑非)的相对优先级是怎样的? 也就是说, `a&&b||c`应理解为`(a&&b)||c`还是`a&&(b||c)`, 或者是随便怎么理解都可以?

我们令`a=0`, `b=0`, `c=1`, 如果按第一种结合方式, 结果应该为1, 而如果按照第二种结合方式的话, 结果应该为0, 如果是随机的话, 答案也会随机(肯定不可能)。

```
1 #include<stdio.h>
2 int main()
3 {
4     int a=0,b=0,c=1;
5     printf("%d",a&&b||c);
6 }
```

按照上述程序进行实验, 输出结果为 1, 证明相对优先级是`(a&&b)||c`。

解释: 优先级: `! > && > ||`

结合方向: `&&`、`||`左结合

! 右结合

5. `if(a) if(b)x++; else y++;`的确切含义是什么? 这个`else`应和哪个`if`配套? 有没有办法明确表达出配套方法?

`else y++;`应与 `if(b)x++;`配套

验证:

```
1 #include<stdio.h>
2 int main()
3 {
4     int x=1,y=2;
5     int a=1,b=0;
6     if(a)
7         if(b)x++;
8         else
9             y++;
10    printf("x=%d,y=%d",x,y);
11 }
```

若我们的猜想正确, 输出结果应为 `x=1,y=3`

否则输出结果为 `x=1,y=2`

输出结果验证了我们的猜想

解释: `else`应该与其上最近且**未进行匹配**的`if`配套