

1. 对浮点数x四舍五入取整: `floor(x + 0.5)`

- 在进行浮点数比较时要考虑浮点误差: 判断浮点数x是否为整数:

```
floor(x+0.5) == x //防止1被表示成0.999999的情况
```

2. 计算程序耗时

```
clock_t startTime = clock();

// code lines

clock_t endTime = clock();
double seconds = double(endTime - startTime) / CLOCKS_PER_SEC;
```

3. 几种输入输出框架

- 标准输入输出

```
int x = 0;
while(scanf("%d", &x) == 1){
    //...
}
//windows结束输入方式: 输入完毕后Enter, Ctrl+z,Enter
//Linux下结束输入方式: 输入完毕后按Ctrl+d
```

- 重定向方法输入输出文件

```
#define LOCAL
#ifdef LOCAL
    freopen("data.in", "r", stdin);
    freopen("data.out", "w", stdout);
#endif
//之后正常使用scanf,printf, 会重定向到对应文件
//有些比赛禁止使用重定向方式, 需要仔细阅读规则
```

- 使用fopen输入输出文件

```
FILE *fin, *fout;
fin = fopen("data.in", "rb");
fout = fopen("data.out", "wb");
int x = 0;
while(fscanf(fin, "%d", &x) == 1){
    //...
}
fprintf(fout, "%d\n", x);
fclose(fin);
fclose(fout);
```

4. 浮点数整数比较

- 浮点数和整数的比较会把整数先转换成浮点数进行
- 浮点数之间不能直接做==或!=判断（因为精度问题，浮点数只保证在一定精度内正确）

```
for(i = 0; i <= 10; i += 0.1) //循环无法停止
    printf("%.1f\n", i);

for(i = 0; fabs(i-10) > 0.0000001; i += 0.1) //使用fabs(), 限定精度, 循环正常终止
    printf("%.1f\n", i);
//abs()是针对整数的
```