分,这说明跨类型的自动转换可能会造成一些精度损失。

计算答案 ans 的时候,distance 是整数类型,delta 也是浮点数类型,所以它们相除会变为浮点数类型,刚好可以存在 ans 变量里。这种现象被称为**自动类型转换**。

还有另外一种写法可以不使用 delta 变量,就像注释行那样。因为右边的变量全部都是整数,所以需要乘上 1.0 将 distance 变成一个浮点数;否则,计算机就将其当作整除来处理造成答案错误。

常见的数据类型[①]见表 2-1。

表 2-1 常见数据类型

数据类型	占用空间	取值范围
char	1 字节,8 位	-128~127
int	4 字节,32 位	-2^{31}~$2^{31}-1$,大约能够表示绝对值不超过 2.1×10^9 的整数
unsigned int	4 字节,32 位	0~$2^{32}-1$,大约能够表示不超过 4.2×10^9 的非负整数
long long	8 字节,64 位	-2^{63}~$2^{63}-1$,大约能够表示绝对值不超过 9.2×10^{18} 的整数
unsigned long long	8 字节,64 位	0~$2^{64}-1$,大约能够表示不超过 1.8×10^{19} 的非负整数
float	4 字节,32 位	大约指数绝对值不超过 37,6 位有效数字
double	8 字节,64 位	大约指数绝对值不超过 307,15 位有效数字

取值范围中的“大约”是保守的估计,有时候即使稍微超过也能表示(也有可能不能),但是不超过这里的范围是可以确保准确的。另外可以看出,这些数据类型默认是带符号的(也就是 signed),可以表示正数、负数或 0。如果在数据类型前面加上 unsigned,就会变成无符号数:只能表示 0 或者正数,不过能够表示的正数范围相对于带符号数扩大了一倍。

从 2-1 表可以看出,long long 可以表示更大范围的数字。那么,能不能只使用这些大范围的数据类型而不用其他的类型呢?答案是否定的。一方面,计算机内存大小是有限的,如果选用这些数据类型,则能够容纳的数字位数就变少了,毕竟这些数据类型占用空间比较大;另一方面,对于一般的运算来说,大范围的数据类型运算速度比较慢(做一个粗浅的类比:做随机 5 位数乘 5 位数的运算速度肯定会比随机 10 位数乘 10 位数要快)。

因此,权衡精度和空间与速度,一般情况下使用 int 数据类型来存放整数。只有当 int 无法存下的一些特别大的数字时,才会去使用 long long;而浮点数如果因为空间有限制时选择 float,否则选择 double。

例 2-2 英文字母。英文有 26 个字母,其中 A 是第一个字母。现在请编程求出:

1) M 是字母表中的第几个字母?

2) 第 18 个字母是什么?

输出两个数字,使用换行隔开。

分析:首先给出如下代码。

① 不同编译器下的取值范围可能与表 2-1 中不同,本书统一按照 GCC8.2 32 位处理器的标准,多数算法竞赛的编译器和这个标准一致。