

第四章 函数

第4.2课 与实型相关的数学库函数

《信息学奥赛一本通·编程启蒙 C++版》

函数名	格式	功能说明	例子
向下取整	$\text{floor}(x)$	求不大于实数 x 的最大整数	$\text{floor}(3.14)=3$
向上取整	$\text{ceil}(x)$	求不小于实数 x 的最小整数	$\text{ceil}(3.14)=4$
四舍五入	$\text{round}(x)$	实数 x 进行四舍五入	$\text{round}(5.5)=6$
取整	$\text{trunc}(x)$	去除数字的小数部分, 功能和函数 INT 类似	$\text{trunc}(5.5)=5$
自然对数函数	$\log(x)$	求实数 x 的自然数对数	$\log(1)=0$
平方根值函数	$\text{sqrt}(x)$	求实数 x 的平方根	$\text{sqrt}(25)=5$
取绝对值	$\text{fabs}(x)$	求实数 x 的绝对值	$\text{fabs}(-1.2)=1.2$

【例 42.1】 雇佣兵

【题目描述】

雇佣兵的体力最大值为 M ，初始体力值为 0 、战斗力为 N 、拥有 X 个能量元素。

当雇佣兵的体力值恰好为 M 时，才可以参加一个为期 M 天的战斗期，战斗期结束体力值将为 0 。在同一个战斗期内，雇佣兵每连续战斗 n 天，战斗力就会上升 1 点， n 为当前战斗期开始时的战斗力。

一个战斗期结束后，雇佣兵需要用若干个能量元素使其体力恢复到最大值 M 从而参加下一个战斗期。每个能量元素可以恢复若干体力，但是每个能量元素可以恢复的体力值不超过当前的战斗力。每个能量元素只能使用一次。

请问：雇佣兵的战斗力最大可以到达多少。

说明：只在战斗期结束后可以使用能量元素，可以使用多个能量元素。

【输入格式】

一行包括三个整数 M 、 N 、 X ，相邻两个整数之间用单个空格隔开。 M 、 N 、 X 均为不超过 10000 的正整数。

【输出格式】

输出一个整数，为雇佣兵的最大战斗力。

【样例输入】

5 2 10

【样例输出】

6

【代码实现】

```
1. #include<bits/stdc++.h>
2. int n,m,x,cf;
3. int main(){
4.     scanf("%d %d %d",&m,&n,&x);
5.     cf=n;//cf 表示当前战力 currentfight
6.     while(cf<=m&&x)
7.     {
8.         x-=ceil(double (m)/cf);
9.         if(x<0)
10.             break;
11.         cf+=floor(double (m)/cf);
12.     }
13.     printf("%d",cf);
14. }
```

【例 42.2】 数学课上

【题目描述】

在一节生动有趣的数学课上，老师问同学们：给出一个浮点数，怎么判断这个数离前后相邻两个整数哪个更近，则输出距离更近的整数。请你按照四舍五入原则，编程输出这个数。

【输入格式】

输入一行，包含 1 个数： n ($0.0 \leq n \leq 100000.0$)，表示题目要求输入的浮点数。题目保证输入浮点数小数点后保留最多 8 位。

【输出格式】

输出共计 1 行，包含 1 个数，表示题目所求的距离更近的整数。

【样例输入】

4.4999

【样例输出】

4

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. double d;
4. int main(){
5.     cin>>d;
6.     cout<<round(d);
7.     return 0;
8. }
```

【例 42.3】 判断浮点数是否相等

【题目描述】

给出两个浮点数，请你判断它们是否相等。这里两个浮点数 a, b 相等定义为： $|a-b| < 10^{-8}$ 。

【输入格式】

两个浮点数 a, b ($0 \leq a, b < 9$)

【输出格式】

如果相等，输出"yes"；否则，输出"no"

【样例输入】

6.7574929582 6.7574929509

【样例输出】

yes

【代码实现】

```
1. #include<iostream>
2. #include<cmath>
3. using namespace std;
4. double a,b;
5. int main(){
6.     cin>>a>>b;
7.     if(fabs(a-b)<1e-8) cout<<"yes";
8.     else cout<<"no";
9.     return 0;
10. }
```

练 42.1 计算线段长度

【题目描述】

已知线段的两个端点的坐标 $A(X_a, Y_a)$, $B(X_b, Y_b)$, 求线段 AB 的长度。

老师告诉了你计算公式如下: $\sqrt{(X_a - X_b)^2 + (Y_a - Y_b)^2}$ 。

【输入格式】

第一行是两个实数, 即 A 的坐标。

第二行是两个实数, 即 B 的坐标。

输入中所有实数的绝对值均不超过 **10000**。

【输出格式】

一个实数, 即线段 AB 的长度, 保留到小数点后 **3** 位。

【样例输入】

1 1

2 2

【样例输出】

1.414

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int main(){
4.     double a,b,c,d;
5.     cin>>a>>b>>c>>d;
6.     printf("%.3lf\n",sqrt((a-c)*(a-c)+(b-d)*(b-d)));
7.     return 0;
8. }
```

练 42.2 求绝对值

【题目描述】

求实数的绝对值。

【输入格式】

输入一个实数 x ， $0 \leq |x| \leq 10^6$ 。

【输出格式】

输出它的绝对值，结果保留两位小数。

【样例输入】

-234.00

【样例输出】

234.00

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. double a;
4. int main(){
5.     cin>>a;
6.     printf("%.2lf", fabs(a));
7.     return 0;
8. }
```

练 42.3 浮点数向零舍入

【题目描述】

输入一个单精度浮点数，将其向零舍入到整数。说明：向零舍入的含义是，正数向下舍入，负数向上舍入。

【输入格式】

一个单精度浮点数 n 。 ($-100000 < n < 100000$)。

【输出格式】

一个整数，即向零舍入到整数的结果。

【样例输入】

2.3

【样例输出】

2

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. float n;
4. int main(){
5.     scanf("%f",&n);
6.     printf("%d",(int)n);
7. }
```

练 42.4 优秀数字

【题目描述】

小 H 是一个喜欢数数的人。我们称一个数是优秀的数字，当且仅当其约数个数为偶数，现在小 H 想知道， $1 \sim n$ 中有多少个优秀的数字。

【输入格式】

一行，一个数 n 。 $1 \leq n \leq 2000$ 。

【输出格式】

一行，一个数，表示答案。

【样例输入】

3

【样例输出】

2

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n;
4. int main(){
5.     cin>>n;
6.     cout<<n-floor(sqrt(n));
7.     return 0;
8. }
```

练 42.5 大象喝水

【题目描述】

一只大象口渴了，要喝 20 升水才能解渴，但现在只有一个深 h 厘米，底面半径为 r 厘米的小圆桶 (h 和 r 都是整数)。问大象至少要喝多少桶水才会解渴。

【输入格式】

输入有一行：包行两个整数，以一个空格分开，分别表示小圆桶的深 h ($1 \leq h \leq 100$) 和底面半径 r ($1 \leq r \leq 100$)，单位都是厘米。

【输出格式】

输出一行，包含一个整数，表示大象至少要喝水的桶数。

【提示】

如果一个圆桶的深为 h 厘米，底面半径为 r 厘米，那么它最多能装 $\pi \times r \times r \times h$ 立方厘米的水。(设 $\pi = 3.14159$)

1 升=1000 毫升; 1 毫升=1 立方厘米

【样例输入】

23 11

【样例输出】

3

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int h,r;
4. float v;
5. int main(){
6.     cin>>h>>r;
7.     v=3.14159*h*r*r;
8.     cout<<ceil(20000/v);
9.     return 0;
10.}
```

谢谢!

—