

# 作业一讲解

本次作业主要涵盖 **基础输入输出、简单运算、分支判断、循环结构** 以及 **基本图形打印** 等内容

涵盖的基础语法：

- 输入 / 输出 (`scanf`、`printf`)
- 基本算术运算
- 判断结构 (`if/else`)
- 循环结构 (`for`、`while`)
- 简单逻辑运算
- 字符操作
- 图形打印技巧 (空格控制、星号数量控制等)

主要知识点：

**输入与输出**

熟悉 `%d`, `%lf`, `%c` 等格式化字符

了解换行 `\n` 的作用

**数学运算**

题目涉及加减乘除、取余 `%`、整除、平方根。

数字反转、三角形面积等经典入门算法的初步实践。

**分支结构 (`if/else`)**

判断闰年、判断数的性质等题涉及条件判断。

## 循环结构

图形打印：控制行数、空格数、星号数。

累加计算，如总用时、总数量等。

## 字符与字符串基础

字符加减（如字母转下一个字母）

注意读取字符前可能需要吸收空白字符（如 "%c"）。

## 手工模拟思维

几乎每题都要求根据数据“推一步”，培养逻辑能力与细心程度。



# 前置知识总结

## ① 输入与输出基础

### printf 输出

```
printf("Hello, World!");
```

输出数字： printf("%d", x);

输出小数： printf("%.2f", y);

### scanf 输入

输入整数： scanf("%d", &x);

输入浮点数： scanf("%lf", &x); (**double** 用 %lf)

输入字符： scanf(" %c", &ch); (前面空格用于吸收换行)

## ② 基本框架

```
1 #include <stdio.h>
2 int main() {
3
4     return 0;
5 }
```

## ③基本数学公式与计算

取余运算 %：用于判断奇偶，或拆分数字。

```
int x = 123;  
int g = x % 10; // 个位
```

8 |  $i \text{ } \cancel{\%} 2 == 0 \rightarrow$  偶数  
9 |  $i \text{ } \% 2 == 1 \rightarrow$  奇数

整除运算 /：a / b 得到整数结果

海伦公式（计算三角形面积）

若三边为 a, b, c，则

1. 计算半周长

```
ini  
  
p = (a + b + c) / 2
```

2. 面积

```
ini  
  
s = sqrt(p * (p - a) * (p - b) * (p - c));
```

## ④字符知识

字符本质是数字（ASCII码）

'A' 的 ASCII = 65

'B' = 66

```
char c = 'A';  
char next = c + 1; // 变成 'B'
```

## ⑤分支判断

### 闰年判断（重点）

闰年的判断规则：

- ✓ 能被 400 整除 → 是闰年
- ✓ 否则能被 4 整除但不能被 100 整除 → 是闰年

```
if (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0))
```

## 数的性质

判断能否被 2 整除：`x % 2 == 0`

是否是三位数：`x >= 100 && x <= 999`

是否同时满足：使用 `&&`

满足其一：使用 `||`

## ⑥循环控制与图形打印

## ⑦时间计算 --小时和分钟进位规则

## ⑧数组排序知识

### 两数交换

```
temp = a;  
a = b;  
b = temp;
```



# 题目展示（比赛从20道中选8道）

## Q1:

编写一个能够输出 Hello,World! 的程序。

提示：

- 使用英文标点符号；
- Hello,World! 逗号后面没有空格。
- H 和 W 为大写字母。

输入格式

无

输出格式

无

输入输出样例

输入 #1

复制

输出 #1

复制

无

Hello,World!

```
1 #include <stdio.h>
2 int main() {
3     printf("Hello,World!");
4     return 0;
5 }
6
7 
```

C:\WINDOWS\system32\cmd. x  
Hello,World!  
请按任意键继续... |

## Q2:

输入格式

两个以空格分开的整数。

输出格式

一个整数。

输入输出样例

输入 #1

复制

输出 #1

20 30

50

```
1 #include <stdio.h>
2 int main() {
3     int a,b,c;
4     scanf("%d %d",&a,&b);
5     c = a+b;
6     printf("%d",c);
7     return 0;
8 }
```

注意格式化输入

C:\WINDOWS\system32\cmd. x  
20 30  
50  
请按任意键继续... |

## Q3:

### 题目描述

用 \* 构造一个对角线长 5 个字符，倾斜放置的菱形。

### 输入格式

没有输入要求。

### 输出格式

如样例所示。用 \* 构成的菱形。

### 输入输出样例

输入 #1      复制      输出 #1

```
1 #include <stdio.h>
2 int main() {
3     int n;
4     scanf("%d", &n);
5     int mid = n / 2;
6     for (int i = 0; i <= mid; i++) {
7         for (int j = 0; j < mid - i; ++j) printf(" ");
8         for (int j = 0; j < 2*i + 1; ++j) printf("*");
9         printf("\n");
10    }
11    for (int i = mid - 1; i >= 0; i--) {
12        for (int j = 0; j < mid - i; ++j) printf(" ");
13        for (int j = 0; j < 2*i + 1; ++j) printf("*");
14        printf("\n");
15    }
16    return 0;
17 }
```

输出 #1

```
*  
***  
*****  
***  
*
```

```
1 #include <stdio.h>
2 int main() {
3     printf(" *\\n");
4     printf(" ***\\n");
5     printf("*****\\n");
6     printf(" ***\\n");
7     printf(" *\\n");
8 }
```

C:\WINDOWS\system32\cmd.exe + v

```
*  
***  
*****  
***  
*
```

## Q4:

### 输入输出样例

输入 #1      复制      输出 #1

```
*
```

输出 #1

```
*
```

\*\*\*\*\*

```
1 #include <stdio.h>
2 int main() {
3     char c;
4     scanf("%c", &c);
5     printf(" %c\\n", c);           // 第一行: 2 空格 + 1 个字符
6     printf(" %c%c%c\\n", c, c, c); // 第二行: 1 空格 + 3 个字符
7     printf(" %c%c%c%c%c\\n", c, c, c, c, c); // 第三行: 5 个字符
8     return 0;
9 }
```

# Q5:

## 题目描述

牛牛最近学习了 C++ 入门课程，这门课程的总成绩计算方法是：

$$\text{总成绩} = \text{作业成绩} \times 20\% + \text{小测成绩} \times 30\% + \text{期末考试成绩} \times 50\%$$

牛牛想知道，这门课程自己最终能得到多少分。

## 输入格式

三个非负整数  $A, B, C$ ，分别表示牛牛的作业成绩、小测成绩和期末考试成绩。相邻两个数之间用一个空格隔开，三项成绩满分都是 100 分。

## 输出格式

一个整数，即牛牛这门课程的总成绩，满分也是 100 分。

## 输入输出样例

输入 #1	输出 #1
100 100 80	90
输入 #2	输出 #2
60 90 80	79

```

1 #include <stdio.h>
2 int main()
3 {
4     int a,b,c;
5     scanf("%d %d %d",&a,&b,&c);
6     double sum=a*0.2+b*0.3+c*0.5;
7     printf("%.0lf",sum);
8 }
9 }
```

# Q6:

一些整数可能拥有以下的性质：

- 性质 1：是偶数；
- 性质 2：大于 4 且不大于 12。

给定整数  $x$ ：

- 性质 1：**是偶数  $\rightarrow x \% 2 == 0$
- 性质 2：** $4 < x \leq 12$

小 A 喜欢这两个性质同时成立的整数；Uim 喜欢这至少符合其中一种性质的整数；小 B 喜欢刚好有符合其中一个性质的整数；正妹喜欢不符合这两个性质的整数。现在给出一个整数  $x$ ，请问他们是否喜欢这个整数？

人物	喜欢条件
小 A	两种性质 同时 成立
Uim	至少一种性质成立
小 B	刚好一种 成立
正妹	两种性质都不成立

## 输入格式

输入一个整数  $x (0 \leq x \leq 1000)$

## 输出格式

输出这 4 个人是否喜欢这个数字，如果喜欢则输出 1，否则输出 0，用空格分隔。输出顺序为：小 A、Uim、小 B、正妹。

## 输入输出样例

输入 #1	输出 #1
12	1 1 0 0

```

1 #include <stdio.h>
2 int main() {
3     int x;
4     scanf("%d", &x);
5
6     int p1 = (x % 2 == 0);           // 性质1: 偶数
7     int p2 = (x > 4 && x <= 12);    // 性质2: 4 < x <= 12
8
9     int A    = p1 && p2;          // 小A: 两个性质都成立
10    int Uim = p1 || p2;          // Uim: 至少一个成立
11    int B    = p1 ^ p2;          // 小B: 刚好一个成立 (异或)
12    int girl = !(p1 || p2);      // 正妹: 两个都不成立
13
14    printf("%d %d %d %d", A, Uim, B, girl);
15
16 }
17 
```

C:\WINDOWS\system32\cmd.exe +

12  
1 1 0 0

## Q7:

输入一个年份，判断这一年是否是闰年，如果是输出 1，否则输出 0。

1582 年以来，闰年的定义：

普通闰年：公历年份是 4 的倍数，且不是 100 的倍数的，为闰年（如 2004 年、2020 年等就是闰年）。

世纪闰年：公历年份是整百数的，必须是 400 的倍数才是闰年（如 1900 年不是闰年，2000 年是闰年）。

### 输入格式

输入一个正整数  $n$ ，表示年份。

### 输出格式

输出一行。如果输入的年份是闰年则输出 1，否则输出 0。

### 输入输出样例

输入 #1

复制 输出 #1

1926

```
1 #include <stdio.h>
2 int main() {
3     int n;
4     scanf("%d", &n);
5
6     if ((n % 4 == 0 && n % 100 != 0) || (n % 400 == 0))
7         printf("1");
8     else
9         printf("0");
10
11    return 0;
12 }
```

输入 #2

复制 输出 #2

1900

0

12 }

## Q8:

小 B 喜欢吃苹果。她今天吃掉了  $x$  个苹果。英语课上学到了 apple 这个词语，想用它来造句。如果她吃了 1 个苹果，就输出 Today, I ate 1 apple.；如果她没有吃，那么就把 1 换成 0；如果她吃了不止一个苹果，别忘了 apple 这个单词后面要加上代表复数的 s。你能帮她完成这个句子吗？

### 输入格式

输入一行一个自然数  $x$ ，表示吃掉的苹果数。

### 输出格式

根据题目要求输出。

### 输入输出样例

输入 #1

复制 输出 #1

1

Today, I ate 1 apple.

复制

输入 #2

复制 输出 #2

3

Today, I ate 3 apples.

复制

### 说明/提示

对于所有数据， $0 \leq x \leq 100$ 。

```
1 #include <stdio.h>
2 int main() {
3     int x;
4     scanf("%d", &x);
5
6     if (x == 1)
7         printf("Today, I ate 1 apple.");
8     else
9         printf("Today, I ate %d apples.", x);
10
11    return 0;
12 }
```

# Q9:

## 题目描述

[复制 Markdown](#) 展开 进入 IDE 模式

给出三个整数  $a, b, c (0 \leq a, b, c \leq 100)$ , 要求把这三位整数从小到大排序。

## 输入格式

输入三个整数  $a, b, c$ , 以空格隔开。

## 输出格式

输出一行, 三个整数, 表示从小到大排序后的结果。

## 输入输出样例

输入 #1

1 14 5

[复制](#) 输出 #1

1 5 14

输入 #2

2 2 2

[复制](#) 输出 #2

2 2 2

```

1 #include <stdio.h>
2 int main() {
3     int a, b, c, temp;
4     scanf("%d %d %d", &a, &b, &c);
5     // 比较 a 和 b
6     if (a > b) {
7         temp = a; a = b; b = temp;
8     }
9     // 比较 a 和 c
10    if (a > c) {
11        temp = a; a = c; c = temp;
12    }
13    // 比较 b 和 c
14    if (b > c) {
15        temp = b; b = c; c = temp;
16    }
17    printf("%d %d %d\n", a, b, c);
18    return 0;
19 }
```

# Q10:

## 题目描述

[复制 Markdown](#) 展开 进入 IDE 模式

输入年份和月份, 输出这一年的这一月有多少天。需要考虑闰年。

## 输入格式

输入两个正整数, 分别表示年份  $y$  和月数  $m$ , 以空格隔开。

## 输出格式

输出一行一个正整数, 表示这个月有多少天。

## 输入输出样例

输入 #1

1926 8

[复制](#) 输出 #1

31

[复制](#)

输入 #2

2000 2

[复制](#) 输出 #2

29

```

1 #include <stdio.h>
2 int main() {
3     int y, m;
4     scanf("%d %d", &y, &m);
5     int days;
6     // 2月需要判断闰年
7     if (m == 2) {
8         if ((y % 4 == 0 && y % 100 != 0) || (y % 400 == 0))
9             days = 29; // 闰年2月
10        else
11            days = 28; // 平年2月
12    }
13    // 4,6,9,11月为30天
14    else if (m == 4 || m == 6 || m == 9 || m == 11)
15        days = 30;
16    else
17        days = 31; // 其他月份为31天
18    printf("%d\n", days);
19    return 0;
20 }
```

# Q11:

## 题目描述

复制 展开 进入 IDE 模式

输入一个不小于 100 且小于 1000，同时包括小数点后一位的一个浮点数，例如 123.4，要求把这个数字翻转过来，变成 4.321 并输出。

## 输入格式

一行一个浮点数

## 输出格式

一行一个浮点数

## 输入输出样例

输入 #1	输出 #1
123.4	4.321

输入示例: 123.4

- 整数部分: 123 → 翻转为 321
- 小数部分: .4 → 移到整数位前, 变成 4
- 输出: 4.321

可以直接把浮点数转换为整数处理，或者用简单的数学公式计算：

$$\text{翻转后} = \text{小数位} + \frac{\text{整数部分翻转}}{1000}$$

```

1 #include <stdio.h>
2 int main() {
3     double x;
4     scanf("%lf", &x);
5     // 1. 将浮点数乘以10，得到整数（整数部分+一位小数）
6     int p = (int)(x * 10 + 0.5); // +0.5 防止浮点数误差
7     int sum = 0;
8     // 2. 翻转整数
9     while (p != 0) {
10         int digit = p % 10; // 取最后一位数字
11         sum = sum * 10 + digit; // 翻转数字
12         p /= 10; // 去掉最后一位
13     }
14     // 3. 转换为浮点数，除以1000得到最终结果
15     double m = (double)sum / 1000.0;
16     // 4. 输出结果
17     printf("%.3f\n", m);
18     return 0;
19 }
```

# Q12:

## 题目描述

复制 展开 进入 IDE 模式

一个三角形的三边长分别是  $a, b, c$ ，那么它的面积为  $\sqrt{p(p - a)(p - b)(p - c)}$ ，其中  $p = \frac{1}{2}(a + b + c)$ 。输入这三个数字，计算三角形的面积，四舍五入精确到 1 位小数。

## 输入格式

第一行输入三个实数  $a, b, c$ ，以空格隔开。

## 输出格式

输出一个实数，表示三角形面积。精确到小数点后 1 位。

## 输入输出样例

输入 #1	输出 #1
3 4 5	6.0

## 说明/提示

数据保证能构成三角形， $0 \leq a, b, c \leq 1000$ ，每个边长输入时不超过 2 位小数。

```

1 #include <stdio.h>
2 #include <math.h> // sqrt函数需要这个头文件
3 int main() {
4     double a, b, c;
5     scanf("%lf %lf %lf", &a, &b, &c);
6     double p = (a + b + c) / 2.0; // 半周长
7     double area = sqrt(p * (p - a) * (p - b) * (p - c)); // 海伦公式
8     // 输出保留一位小数
9     printf("%.1f\n", area);
10    return 0;
11 }
```