

A - 国庆快乐！：)

难度	考点
1	<code>printf()</code> 的换行输出

题目分析

仿照 Hello World 程序中的输出样例，输出题目中要求的字符串即可。

(请直接复制题目中的输出数据到你的程序中，避免手动输入错误)

示例程序

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("2020/10/10\n");
6      printf("I love you, China");
7
8      return 0;
9  }
```

B - Hinge Loss

难度	考点
2	if 判断语句

题目分析

本题需要依次判断三种不同情况并分别处理，采用多个 `if` 语句可以实现分类判断，并且注意要覆盖所有情况。另外，判断相等时要用 `a == b` 而不是 `a = b`，输出字符串时建议直接从题目中复制并粘贴。

使用 `scanf()` 输入变量时，需要特别注意：

1. 变量前一定要加取地址符 `&`
2. 如果输入两个或多个变量，则各个 `%d` 之间不要有空格

示例程序

```
1  #include<stdio.h>
2
3  int main()
4  {
5      int a, x;
6      scanf("%d%d", &x, &a);
7      if (x > a)
8          printf("0");
9      if (x <= a)
10         printf("%d", -x + a);
11     if (x == a)
12         printf("\nIndifferentiable Point"); // 注意此处判断相等的表达式。
13
14     return 0;
15 }
16
```

C - 五则运算

难度	考点
2	数学运算, <code>if...else</code> 判断语句

题目分析

本题来源于 PPT 中的例题。

题目保证了结果不会超出 `int` 范围，因此对于加减乘运算只要直接用 `printf()` 输出结果即可，注意每输出一行都要用换行符 `\n` 进行换行。

对于整除和取模操作，需要用 `if` 额外判断除数是否为 `0`，如果为 `0` 则输出 `Divided by zero!`，这里同样注意不要遗漏换行符 `\n`。另外，如果 `if` 或 `else` 中有多条语句，则需要添加一对花括号 `{}` 将多条语句包括起来。

在 C 语言中，两个整数用 `/` 操作默认为整除。提示：如果想获得小数结果，可以用 `(double)a / b` 或者 `1.0 * a / b` 先将 `a` 强制转换为浮点数类型后再相除。

示例程序

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a, b;
6      scanf("%d%d", &a, &b);
7      printf("%d\n", a + b);
8      printf("%d\n", a - b);
9      printf("%d\n", a * b);
10     if (b == 0)
11     {
12         printf("Divided by zero!\n");
13         printf("Divided by zero!\n");
14     }
15     else
16     {
17         printf("%d\n", a / b);
18         printf("%d\n", a % b);
19     }
20
21     return 0;
22 }
```

D - 散落的零钱

难度	考点
3	循环

题目分析

本题需要根据输入的 n ，重复执行 n 次“输出 1”的操作。使用循环语句循环 n 次可以有如下两种方法：

```
1 while (n > 0)
2 {
3     ... // 循环内容
4     n = n - 1;
5 }
```

```
1 int i;
2 for(i = 1; i <= n; i = i + 1) //或 for(i = 0; i < n; i = i + 1)
3 {
4     ... // 循环内容
5 }
```

提示：如果将 `while` 循环中 `(n > 0)` 改为 `(n >= 0)`，实际会执行几次？如果将 `for` 循环中 `(i = 1; i <= n; i = i + 1)` 改为 `(i = 1; i < n; i = i + 1)`，实际会执行几次？你可以自行测试并理解“如何根据循环次数设计循环条件”。

另外需要注意，OJ 会忽略行末空格，因此你可以重复输出 n 次字符串 `"1 "`（不包括引号，注意后面有一个空格），实现每两个 `1` 之间用空格分隔的效果。

示例程序

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int n, i;
6     scanf("%d", &n);
7     for (i = 1; i <= n; i = i + 1)
8         printf("1 ");
9
10    return 0;
11 }
```

E - 简单平方和运算

难度	考点
3	数学运算，循环

题目分析

本题是一道简单的数学问题。注意：如果在 `main()` 函数内部声明一个 `int` 型的变量，则注意是否需要给该变量赋初始值，如题目中的 `ans`。

示例程序

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, ans = 0, temp;
6      scanf("%d", &n); // 读取数据组数n
7      while (n > 0)    // 执行n次循环
8      {
9          scanf("%d", &temp);
10         ans = ans + temp * temp;
11         n = n - 1;
12     }
13     printf("%d", ans); // 输出结果
14
15     return 0;
16 }
```

使用 `for` 循环代替 `while` 循环的一种实现方法如下：

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, ans = 0, temp, i;
6      scanf("%d", &n); // 读取数据组数n
7      for(i = 1; i <= n; i = i + 1) // 执行n次循环
8      {
9          scanf("%d", &temp);
10         ans = ans + temp * temp;
11     }
12     printf("%d", ans); // 输出结果
13
14     return 0;
15 }
```

F - 减肥

难度	考点
4	判断、循环、简单计算

题目分析

本题需要按照题目要求进行判断。统计星号个数可以采用循环累加，也可以根据 a, b 的大小直接利用面积公式计算。

示例程序

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int s, a, b, i, m, cnt = 0;
6      scanf("%d", &s);
7      if (s == 3)          // 蛋糕为三角形
8      {
9          scanf("%d", &a);
10         for (i = 1; i <= a; i = i + 1) // i 表示当前是三角形的第几层
11             cnt += i * 2 - 1;          // 第 i 层的星号个数为 2i - 1, cnt 表示
            从三角形第一行到当前第i行的星号总数
12     }
13     else                  // 蛋糕为长方形
14     {
15         scanf("%d%d", &a, &b);
16         cnt = a * b;
17     }
18     scanf("%d", &m);
19     if (cnt <= m)
20         printf("ENJOY YOUR MEAL!");
21     else
22         printf("FBI WARNING!");
23
24     return 0;
25 }
```

G - 向上取整

难度	考点
3	条件判断，循环，向上取整和向下取整

题目分析

本题有三种解法，具体分析如下：

方法一：利用循环暴力求解

我们只需要找到满足 $a \leq bx$ 的最小整数解 x_{\min} 即可，因此可以先声明并初始化一个变量 `int x = 0`，如果 x 不满足 $a \leq bx$ （即 `!(a <= b * x)`），就令 `x = x + 1`，直到条件满足并输出结果。

方法二：利用判断分类求解

通过观察发现，大部分情况下有 $\left\lceil \frac{a}{b} \right\rceil = \left\lfloor \frac{a}{b} \right\rfloor + 1$ ，并且只有当 $\frac{a}{b}$ 是整数时，有 $\left\lceil \frac{a}{b} \right\rceil = \left\lfloor \frac{a}{b} \right\rfloor$ ，因此只需要特殊判断 `a % b` 是否为 0 即可。

方法三：利用表达式直接求解

向上取整和向下取整可以双向转化： $\left\lceil \frac{a}{b} \right\rceil = \left\lfloor \frac{a + b - 1}{b} \right\rfloor$ ，因此直接利用表达式 `(a + b - 1) / b` 也可实现本题。

示例程序 1

```
1 // 方法一：利用循环暴力求解
2 #include <stdio.h>
3
4 int main()
5 {
6     int a, b;
7     int x = 0; // 根据题意，x 最小为 0，因此初始值设置为 0
8     scanf("%d%d", &a, &b);
9     while (!(a <= b * x)) // 当 x 不满足 a <= bx 时，就令 x = x + 1
10         x = x + 1;
11     printf("%d", x);
12
13     return 0;
14 }
```

示例程序 2

```
1 // 方法二：利用判断分类求解
2 #include <stdio.h>
3
4 int main()
5 {
6     int a, b;
```

```
7     scanf("%d%d", &a, &b);
8     if(a % b == 0)        // 如果 a / b 是整数
9         printf("%d", a / b);
10    else
11        printf("%d", a / b + 1);
12
13    return 0;
14 }
```

示例程序 3

```
1 // 方法三：利用表达式直接求解
2 #include <stdio.h>
3
4 int main()
5 {
6     int a, b;
7     scanf("%d%d", &a, &b);
8     printf("%d", (a + b - 1) / b);
9
10    return 0;
11 }
```


H - Easy GCD

难度	考点
3	最大公因数

题目分析

本题来源于 PPT 中的例题，比较简单，使用 PPT 中的代码可以直接通过本题。

本题的难点是根据最大公约数的求解方法构造循环条件，也就是代码中的 `while (!(a % gcd == 0 && b % gcd == 0))` 这一条语句。这个循环条件的逻辑较为复杂，同学们可以想想为什么这么写。

示例程序

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a, b, gcd;
6      scanf("%d%d", &a, &b);
7      gcd = a;
8      if (b < gcd)
9          gcd = b;
10     while (!(a % gcd == 0 && b % gcd == 0))
11     {
12         gcd = gcd - 1;
13     }
14     printf("%d", gcd);
15
16     return 0;
17 }
```

提示：除了 PPT 中的方法，这道题也可以使用递归的方法解决（我们以后会学习到递归，因此在此不做详细讲解）。这里给出更相减损法和辗转相除法的示例程序。

```
1  // 更相减损法
2  #include <stdio.h>
3
4  int gcd(int x, int y)
5  {
6      if (x == y)
7      {
8          return x;
9      }
10     if (x > y)
11     {
12         return gcd(x - y, y);
13     }
14     else
15     {
16         return gcd(y - x, x);
17     }
```

```
18 }
19 int main()
20 {
21     int a, b;
22     scanf("%d%d", &a, &b);
23     printf("%d", gcd(a, b));
24
25     return 0;
26 }
```

```
1 // 辗转相除法
2 #include<stdio.h>
3
4 int gcd(int x, int y)
5 {
6     if(y == 0)
7     {
8         return x;
9     }
10    else
11    {
12        return gcd(y, x % y);
13    }
14 }
15 int main()
16 {
17     int a, b;
18     scanf("%d%d", &a, &b);
19     printf("%d", gcd(a, b));
20
21     return 0;
22 }
```

I - 水水的分数加法

难度	考点
3	简单四则运算，数学

题目分析

题意是计算 $\frac{a}{b} + \frac{c}{d}$ 。根据分数加法的运算法则可知 $\frac{a}{b} + \frac{c}{d} = \frac{ad}{bd} + \frac{bc}{bd} = \frac{ad + bc}{bd}$ ，整体上可以分为三个步骤：

1. 计算公分母 $p = bd$ 。由题意可知，数据的取值范围都在 $[1, 500]$ 之间，因此直接将 bd 分母相乘即可，不会产生数据溢出等错误。
2. 计算通分后的两个分子和 $q = ad + bc$ 。
3. 计算约分后的结果，即对非最简分数，将分子 p 和分母 q 同时除以它们的最大公因数 $\text{gcd}(p, q)$ 。

最大公因数的计算方法可以参考 H 题 Easy GCD 这道题的题解，代码也可直接复用。

输入两个分数的方法根据题目的提示可以相应写出，`scanf()` 的第一个参数(格式字符串)为 `"%d/%d %d/%d"`。

示例程序

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a, b, c, d; // 输入的两个分数的分子和分母
6      int p, q;       // 计算结果的分子和分母
7      int gcd;        // 计算结果的分子分母之间的最大公因数，用于约分
8
9      // 同时输入两个分数
10     scanf("%d/%d %d/%d", &a, &b, &c, &d);
11
12     q = b * d;        // 计算公分母 q = bd
13     p = a * d + b * c; // 计算通分后的两个分子和 q = ad + bc
14
15     // 求 gcd 的代码直接参考 H 题 Easy GCD
16     gcd = p;
17     if (q < gcd)
18         gcd = q;
19     while (!(p % gcd == 0 && q % gcd == 0))
20         gcd = gcd - 1;
21
22     // 约分
23     p = p / gcd;
24     q = q / gcd;
25
26     // 输出
27     printf("%d/%d + %d/%d = %d/%d", a, b, c, d, p, q);
28
29     return 0;
30 }
```

J - 周游加拿大

难度	考点
4	循环与条件判断

题目分析

根据题干解释，我们可声明一个变量，用来存储当前坐标 x 。在读入每天新的旅行距离 x_i 后，判断表达式 $x + x_i$ 是否超出旅行区间范围 $[-50, 50]$ 。如果超出，则输出一行 “can't move, touch fish.”；如果没超出，则将 $x + x_i$ 赋值给 x ，并进行下一天旅行的判断。

最后输出一行结果 x 。

示例程序

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int i, x_end = 0, x, n;
6      scanf("%d", &n);
7      for (i = 0; i < n; i = i + 1)
8      {
9          scanf("%d", &x);
10         if (x_end + x >= -50 && x_end + x <= 50)
11         {
12             x_end = x_end + x;
13         }
14         else
15         {
16             printf("can't move, touch fish.\n");
17         }
18     }
19     printf("%d\n", x_end);
20
21     return 0;
22 }
```