

作业3

1.求自然数之和

由键盘输入正整数 n ，请你用循环计算 $1 + 2 + 3 + 4 + 5 + \dots + n$ 的值，并输出。

输入

输入一个正整数 n ，且 $0 < n \leq 100$ 。

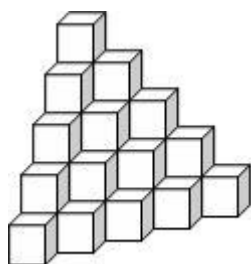
输出

输出 $1 + 2 + 3 + 4 + 5 + \dots + n$ 的值，值在int范围内。

```
#include <stdio.h>
int main(){
    int n, sum;
    scanf("%d", &n);
    while(n>0){
        sum += n--;
    }
    printf("%d", sum);
}
```

2.简单数数小木块

在墙角堆放着一堆完全相同的正方体小木块，如下图所示：



因为木块堆得实在是太有规律了，你只要知道某一层的层号（从上向下编号，层号从1开始），就可以计算该层木块的数量了。由键盘输入正整数 n 表示层号，请你用循环计算该层木块的数量。

输入

输入一个正整数 n ，表示这堆小木块中某一层的层号，已知 $1 \leq n \leq 100$ 。

输出

输出一个整数，表示这层小木块的数量，值在int范围内。

```
#include <stdio.h>
int main(){
    int n,num;
    scanf("%d",&n);
    while(n>0){
        num+=n--;
    }
    printf("%d",num);
}
```

3.再求和

用循环计算 $1*3 + 5*7 + 9*11 + \dots + 597*599$ 的和。

输出

输出 $1*3 + 5*7 + 9*11 + \dots + 597*599$ 的和，值在int范围内。

```
#include <stdio.h>
int main() {
    int a=597,sum;
    while(a>0){
        sum += a*a+a*2;
        a = a-4;
    }
    printf("%d",sum);
}
```

4.找2

小明和小红两位同学互相进行数字游戏，他们的游戏规则是：任意给出一个自然数 n ，查找在自然数 $1\sim n$ 范围内有多少个数字2？且 $1 \leq n \leq 2000$ 。请使用循环来实现。

输入

输入一个正整数 n ，且 $1 \leq n \leq 2000$ 。

输出

输出在自然数 $1\sim n$ 范围内数字2的个数。

```
#include <stdio.h>
int main() {
    int n,i=1, count = 0;
    scanf("%d", &n);
    while (i <= n) {
        int num = i;
        while (num > 0) {
            if (num % 10 == 2) {
                count++;
            }
            num /= 10;
        }
        i++;
    }
    printf("%d", count);
}
```

5.回文数判断

“回文数”是一种数字。如：98789，这个数字正读是98789，倒读也是98789，正读倒读一样，所以这个数字就是回文数。

由键盘输入正整数 n ，请你用循环判断该数是否为回文数，是输出“TRUE”，否则输出“FALSE”。

输入

输入一个正整数 n ，且 $10 \leq n \leq 2000000$ 。

输出

根据题意输出“TRUE”或“FALSE”(不输出引号)。

```
#include <stdio.h>

int main() {
    int n, temp, reversed = 0;
    scanf("%d", &n);

    temp = n; // 保存原始值
    while(temp > 0) {
        reversed = reversed * 10 + temp % 10; // 反转数字
        temp /= 10; // 移除最后一位
    }

    printf(n == reversed ? "TRUE" : "FALSE");
}
```

6.蟠桃记

喜欢西游记的同学肯定都知道悟空偷吃蟠桃的故事，你们一定都觉得这猴子太闹腾了，其实你们是有所不知：悟空是在研究一个数学问题！

当时的情况是这样的：

第一天悟空吃掉桃子总数一半多一个，第二天又将剩下的桃子吃掉一半多一个，以后每天吃掉前一天剩下的一半多一个，到第n天准备吃的时候只剩下一个桃子。

聪明的你，请用循环帮悟空算一下，他第一天开始吃的时候桃子一共有多少个呢？

输入

输入一个正整数n ($1 < n < 30$)，表示只剩下一个桃子的时候是在第n天发生的。

输出

输出第一天开始吃的时候桃子的总数。

```
#include <stdio.h>

int main() {
```

```

int n, total = 1, days = 1; // 第n天初始1个
scanf("%d", &n);

// 从第1天开始逆推
while (days < n) { // 循环n-1次
    total = (total + 1) * 2;
    days++;
}

printf("%d", total);
}

```

7.数列生成

有一列数是：1，5，11，19，29，……。请你找出数列的规律，然后输出这个数列的前N项，并计算前N项的和。请用循环实现。

输入

输入一个正整数N，且 $1 \leq N \leq 20$ 。

输出

输出包含N+1行，前N行为数列。最后一行为数列之和。

```

#include <stdio.h>

int main() {
    int N, current = 1, sum = current, delta = 4, count = 1;
    scanf("%d", &N);
    printf("%d\n", current); // 输出第1项
    while (count < N) { // 生成剩余N-1项
        current += delta; // 计算新项
        sum += current; // 累加总和
        printf("%d\n", current);
        delta += 2; // 差值递增
        count++; // 已生成项数+1
    }

    printf("%d\n", sum); // 输出总和
}

```

8.猴子选桃

猴子面前的桌子上摆了一排大小不一的桃子，它只能拿一个。猴子向你求援，希望你能够帮助它选择最重的一个，将选出的最大值输出。请用循环实现。

输入

输入包含两行：

第一行是一个整数 n ，表示 n 个桃子，且 $0 < n < 2000$ 。

第二行是 n 个整数，表示桃子的重量，邻近两数之间用一个空格隔开。

输出

输出 n 个桃子中最重的一个。

```
#include <stdio.h>

int main() {
    int n, max;
    scanf("%d", &n);    // 读取桃子数量
    scanf("%d", &max);  // 读取第一个重量作为初始最大值

    while (--n) {        // 循环n-1次（通过递减n控制）
        int current;     // 每次循环只用这一个新int
        scanf("%d", &current);
        max = current > max ? current : max;
    }

    printf("%d", max);
}
```

9.逢7过

相信大家都玩过这个游戏，一群人围坐一圈，开始喊数，是7的倍数或者数中含有7的均要说“过”，其余的数就直接说出数的大小。

为了简化问题，我们规定，对于下面的情况我们要求喊“过”：

- 1) 数是7的倍数，
- 2) 数的个位含有7。

也就是说177需要喊过，而176不需要喊过。

我们规定所有需要喊“过”的数为数列{An}，数列{An}从小到大排列，无相同的两项。给出一个正整数n ($1 \leq n \leq 100$)，输出数列{An}的前n项，每项占单独一行。请用循环实现。

输入

输入一个正整数n，且 $1 \leq n \leq 100$ 。

输出

输出n行，每行一个数，为数列{An}的前n项。

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    {
        int cnt = 0;
        {
            int i = 1;
            while (cnt < n) {
                {
                    int mod7 = i % 7;
                    if (mod7 == 0) {
                        printf("%d\n", i);
                        ++cnt;
                        ++i;
                        continue;
                    }
                }
                {
                    int digit = i % 10;
                    if (digit == 7) {
                        printf("%d\n", i);
                        ++cnt;
                    }
                }
                ++i;
            }
        }
    }
}
```

10.输出完全平方数

一个数如果是另一个整数的完全平方，那么我们就称这个数为完全平方数，也叫做平方数。例如：0, 1, 4, 9, 16, 25,.....

由键盘输入正整数n，请你用循环统计并输出小于等于n的所有完全平方数。

输入

输入一个正整数n，且 $1 \leq n \leq 200$ 。

输出

输出若干行，每个完全平方数占一行。请按照从小到大的顺序输出。

```
#include <stdio.h>
int main(){
    int n,i = 0;
    scanf("%d",&n);
    while(i*i<=n){
        printf("%d\n",i*i);
        i++;
    }
}
```