

Recurrences 4

จากฟังก์ชันตามนิยามที่กำหนดให้ต่อไปนี้ จงเขียนฟังก์ชันในโครงของโปรแกรมข้างล่างนี้

Schröder Triangle	$T(\text{int } n, \text{int } k)$	$T(n, k) = T(n, k - 1) + T(n - 1, k - 1) + T(n - 1, k) \text{ if } 1 < k \leq n$ $T(n, k) = 0 \text{ if } k > n$ $T(n, 0) = 1$ <div>$n \geq 0$ และ $k \geq 0$ แน่ ๆ</div>
Schröder	$S(\text{int } n)$	$S(n) = 3S(n - 1) + \sum_{k=1}^{n-2} (S(k)S(n - k - 1)) \text{ if } n \geq 2, \quad S(0) = 1, S(1) = 2$ <div>$n \geq 0$ แน่ ๆ</div>
Wedderburn-Etherington	$W(\text{int } n)$	$W(n) = \frac{W\left(\frac{n}{2}\right)\left(W\left(\frac{n}{2}\right) + 1\right)}{2} + \sum_{i=1}^{\frac{n}{2}-1} (W(i)W(n - i)) \text{ for } n = 2, 4, 6, 8, \dots$ $W(n) = \sum_{i=1}^{\frac{n+1}{2}-1} (W(i)W(n - i)) \text{ for } n = 3, 5, 7, 9, \dots$ <div>$n \geq 0$ แน่ ๆ</div> $W(0) = 0, W(1) = 1$

```
#include <iostream>

using namespace std;

long long T(int n, int k) { // Schroder Triangle
}
long long S(int n) { // Schroder Number
}
long long W(int n) { // Wedderburn-Etherington Number
}

int main() {
    string fn;
    int p1, p2;
    while (cin >> fn >> p1) {
        if (fn == "T") {
            cin >> p2;
            cout << fn << '(' << p1 << ', ' << p2 << ") = " << T(p1, p2) << endl;
        } else if (fn == "S") {
            cout << fn << '(' << p1 << ") = " << S(p1) << endl;
        } else if (fn == "W") {
            cout << fn << '(' << p1 << ") = " << W(p1) << endl;
        }
    }
    return 0;
}
```

ห้ามเปลี่ยนแปลง code ใน
main โดยเด็ดขาด

ข้อมูลนำเข้า

หลายบรรทัด แต่ละบรรทัดมีชื่อฟังก์ชัน ตามด้วยค่าของพารามิเตอร์ (เป็นจำนวนเต็ม มีหนึ่งหรือสองตัวขึ้นกับชื่อฟังก์ชัน) คั่นด้วยช่องว่าง

ข้อมูลส่งออก

ผลของการเรียกฟังก์ชันที่มีชื่อและค่าพารามิเตอร์ที่รับมาในแต่ละบรรทัด ในรูปแบบที่แสดงในตัวอย่าง

ตัวอย่าง

input	output (int ทางจอภาพ)
T 10 11	T(10,11) = 0
T 100 0	T(100,0) = 1
T 12 10	T(12,10) = 12455118
S 0	S(0) = 1
S 1	S(1) = 2
S 20	S(20) = 17518619320890
W 0	W(0) = 0
W 1	W(1) = 1
W 20	W(20) = 293547

ชุดข้อมูลทดสอบ

- 30% ค่า n หรือ k ที่ต้องการหาเป็น **initial conditions** ของ **recurrences**
- 70% ค่า n หรือ k เป็นค่าใด ๆ
- กรณีที่ทดสอบเรียงตามลำดับ **TTTSSSWWWX** โดยที่
 - **T** คือทดสอบ $T(n, k)$
 - **S** คือทดสอบ $S(n)$
 - **W** คือทดสอบ $W(n)$
 - **X** คือทดสอบทั้ง T , S , และ W