Recurrences

จากฟังก์ชันตามนิยามที่กำหนดให้ต่อไปนี้ จงเขียนฟังก์ชันในโครงของโปรแกรมข้างล่างนี้

Greater Common Divisor	`int G(int x, int y)`	$G(x,y) = G(y, x \bmod y) \ \ if\ y > 0, \quad G(x,0) = x$
Josephus Problem	`int J(int n, int k)`	$J(n,k) = (J(n-1,k)+k) \bmod n \ \ if\ n > 1, \ \ J(1,k) = 0$
Ackermann Number	`int A(int m, int n)`	$A(m,n) = \begin{cases} A(m-1,1) & if\ m > 0\ and\ n = 0 \\ A(m-1, A(m,n-1)) & if\ m > 0\ and\ n > 0 \\ n+1 & if\ m = 0 \end{cases}$
Delannoy number	`int D(int m, int n)`	$D(m,n) = D(m-1,n) + D(m-1,n-1) + D(m,n-1) \ \ if \ \ m,n > 0,$ $D(m,0) = D(0,n) = 1$

เขียนฟังก์ชัน ในโครงของโปรแกรมข้างล่างนี้

```
#include <iostream>
#include <map>

using namespace std;

int G(int x, int y) {  // Greater Common Divisor

}
int J(int n, int k) {  // Josephus Problem

}
int A(int m, int n) {  // Ackermann Number

}
int D(int m, int n) {  // Delannoy number

}

int main() {
    map<string, int(*)(int,int)> func = {{"G",G}, {"J",J}, {"A",A}, {"D",D}};
    string fn;
    int p1, p2;
    while (cin >> fn >> p1 >> p2) {
        if (func.find(fn) != func.end())
            cout << fn << '(' << p1 << ',' << p2 << ") = " << func[fn](p1,p2) << endl;
    }
    return 0;
}
```

ข้อมูลนำเข้า

หลายบรรทัด แต่ละบรรทัดมีชื่อฟังก์ชัน ตามด้วยค่าของพารามิเตอร์ (int มีหนึ่งหรือสองตัวขึ้นกับชื่อฟังก์ชัน) คั่นด้วยช่องว่าง

ข้อมูลส่งออก

ผลของการเรียกฟังก์ชันที่มีชื่อและค่าพารามิเตอร์ที่รับมาในแต่ละบรรทัด ในรูปแบบที่แสดงในตัวอย่าง

ตัวอย่าง

input	output (int ทางจอภาพ)
`G 10013 19057` `J 30 5`	`G(10013,19057) = 323` `J(30,5) = 2`
`A 3 2` `D 5 5`	`A(3,2) = 29` `D(5,5) = 1683`