ЕСЕНЕН ТУРНИР ПО ИНФОРМАТИКА
Шумен, 10–12 ноември 2006 г.
Тема за група В (10-11 клас)

Задача В1. ФИБОНАЧИ

Разглеждаме редица от цели числа, в която първият и вторият член са равни на 1, а всеки следващ е равен на сумата на двата предишни за него члена в редицата. Напишете програма **`fib`**, която въвежда целите числа n и m, и извежда остатъка при делението с m на n-тия член на редицата ($0 < n < 10^{15}$, $0 < m < 10^7$).

Пример. Вход

```
1 1
```

Изход

```
0
```

Решение:

Наивният начин за пресмятане на n-тото число от редицата на Фибоначи ($f_1 = 1, f_2 = 2, f_n = f_{n-1} + f_{n-2}$) няма да може да реши задачата, поради твърде големия брой итерации за големи стойности на n:

```
long long int f1=1, f2=1, f;
for(long long int i=3;i<=n;i++)
 {
  f=f1%m+f2%m;
  f1=f2;
  f2=c%m;
 }
 cout << c;
```

Един възможен подход за решаване на задачата се основава на използване на формули, чрез които можем бързо да пресмятаме, например като скачаме от n-тия член f_n направо на $2n$-тия f_{2n}. Това може да се осъщести чрез двойката равенства, чието доказателство се извършва с математематическа индукция:

$$f_{2n-1} = f_n^2 + f_{n-1}^2; \qquad f_{2n} = f_n^2 + 2 f_n f_{n-1}$$

или да използваме по-общото равенство:

$$f_{n+m} = f_{m+1} f_n + f_m f_{n-1}$$

(виж: Clifford A. Reiter. *Fast Fibonacci Numbers*, The Mathematica journal 2(3) 1992. Clifford A. Reiter, Fibonacci Numbers: Reduction Formulas and Short Periods, *The Fibonacci Quarterly*, 31(4) 1993, pp. 315-324. Воробиев, Н. Н., Числа Фибоначи, изд. Наука, Москва, 1978 г.)

Друг подход се основава на следната трансформацията, при която от дадена тройка последователни числа на Фибоначи, се получава следващата тройка: $(f_{i-2}, f_{i-1}, f_i) \to (f_{i-1}, f_i, f_{i+1})$ чрез матрично умножение:

$$\begin{pmatrix} f_{i+1} & f_i \\ f_i & f_{i-1} \end{pmatrix} = \begin{pmatrix} f_i & f_{i-1} \\ f_{i-1} & f_{i-2} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

Ако означим

$$A_i = \begin{pmatrix} f_i & f_{i-1} \\ f_{i-1} & f_{i-2} \end{pmatrix}, \qquad f_0 = 0, \quad i = 2, 3, \ldots, n,$$

пресмятането по модул m на степента $A_n = (A_2)^{n-1}$ може да се извърши с алгоритъм за бързо повдигане на степен чрез функцията `power`:

```
int a[][]={{1,0},{0,1}}, p[][]={{1,1},{1,0}};
void power(long long int n)
{
 if (n > 1) {power(n/2); a = mul(a,a);}
 if (n%2==1) a=mul(a,p);
}
```

където първоначално `a` е единичната матрица, `p` е заредена с елементите на A_2, а `mul` е функция за умножение на две матрици, като елементите на произведението се пресмятат по модул m. Това осигурява логаритмична изчислителна сложност вместо линейната сложност при наивния подход.

Емил Келеведжиев

Задача B2. ABC

Разглеждаме равностранен триъгълник с върхове точките A, B, C и с лице, равно на единица. Разделяме го на 4 по-малки еднакви равностранни триъгълника, като ползваме средите на страните му за върхове на новите триъгълници. Тези триъгълници означаваме съответно с буквите O, A, B и C, в зависимост от това, дали имаме предвид централния от тях, или тези, които са разположени откъм върховете A, B или C на първоначалния триъгълник. Всеки един от последните три по-малки триъгълници (триъгълниците A, B и C) го разделяме отново по същия начин на 4 още по-малки триъгълника. За да ги означим, ползваме също буквите O, A, B или C, но добавени към буквата, означаваща триъгълника, от който всеки нов триъгълник съответно е получен. Така имаме триъгълници AO, AA, AB, AC, BO, BA, BB, BC, CO, CA, CB и CC. Разглеждаме триъгълниците от това второ поколение, и за тези от тях, чието означение не завършва с O, повтаряме същата операция и получаваме следващото трето поколение триъгълници: AAO, AAA, AAB, ..., и т.н. Така продължаме построяването и на още следващи поколения триъгълници.

Напишете програма **`abc`**, която въвежда последователност от низове от описания вид (всеки низ задава триъгълник от някое поколение), и пресмята лицето на тази част от равнината, която е покрита със съответните на тези низове триъгълници. Лицето трябва да бъде изведено като обикновена несъкратима дроб, представена от числителя и знаменателя си, записани на един ред в стандартния изход и разделени с един интервал.

Вашата програмата трябва да прочете входните данни от стандартния вход. На първия ред във входа е зададен броят на следващите низове (цяло положително число, по-малко от 999). Следват толкова на брой редове, като всеки съдържа по един низ, състоящ се от някои от главни латински букви `A`, `B`, `C` или O. Дължината на всеки от тези низове не надминава 20 букви.

Пример. Вход

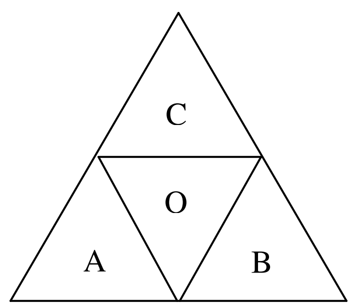
```
O
A
AB
```

Изход

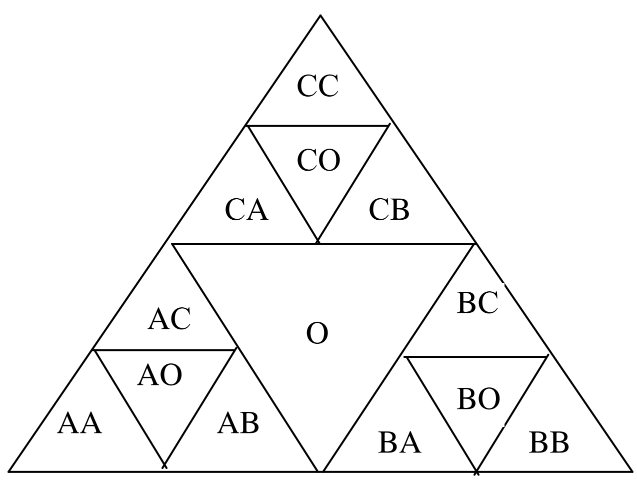
```
1 2
```

Решение:

При първото разделяне на триъгълници се получават четири триъгълника с лица равни на 1/4 от лицето на дадения триъгълник. Това са триъгълниците с имена A, B, C и O:



Второто поколение триъгълници имат лица, равни на 1/16 от лицето на първоначалния триъгълник:



За следващите поколения триъгълници, лицата са равни на 1/64, 1/256, и т.н., т.е. изразяват се като четна степен на 2.

За да решим задачата, трябва да премахнем тези от зададените във входните данни триъгълници, които са части от други триъгълници. За целта дадените низове се сортират лексикографски и след това преминавайки през така получената подредба, ако намерим, че за текущия низ, предходният му се съдържа от първа позиция в него, изключваме текущия. След това, в един числов масив запомняме дължините на непремахнатите низове. Всяка от тези дължини е равна на номера на поколението $k(t)$ на съответния триъгълник t. Понеже нашата цел е да съберем дроби от вида $1/(2^{k(t)})$, за намирането на общия знаменател търсим най-голямата от дължините $k(t)$. При пресмятането на степените на двойката може да използваме побитови измествания.

Накрая, ако е възможно съкращаваме дробта (като пробваме деления с 2 на числителя и знаменателя) и извеждаме резултата.

Зорница Дженкова

Задача B3. ПРАВОЪГЪЛНИ ТРИЪГЪЛНИЦИ

Правоъгълник с дължини на страните m и n (m и n са цели числа, за които $0 < m < 30$ и $0 < n < 30$) е разделен на mn квадрата със страна 1. Всяка точка, която е връх на поне един от тези квадрати, ще наричаме възел.

Да се напише програма **`rtri`**, която въвежда от един ред на стандартния вход стойности за m и n, и извежда на стандартния изход броя на правоъгълните триъгълници, върховете на които са възли.

ПРИМЕР

Вход
```
1 2
```

Изход
```
14
```

Решение:

Нека спрямо правоъгълна координатна система Oxy точките A, B и C имат координати $A(x_A, y_A)$, $B(x_B, y_B)$ и $C(x_C, y_C)$. Правите AB и AC са перпендикулярни тогава и само тогава, когато $(x_B - x_A)(x_C - x_A) + (y_B - y_A)(y_C - y_A) = 0$, а триъгълникът ABC е правоъгълен тогава и само тогава, когато $AB \perp AC$ или $AB \perp BC$ или $AC \perp BC$.

Да означим правоъгълника с $MNPQ$ и нека $MN = m$ и $MQ = n$. Нека спрямо правоъгълна координатна система координатите на върховете на правоъгълника са $M(0,0)$, $N(m,0)$, $P(m,n)$ и $Q(0,n)$. Абсцисата и ординатата на всеки възел могат да бъдат представени като елементи на два масива от цели числа $x[\ ]$ и $y[\ ]$, като на възел с координати (a,b) съпоставяме елементите $x[am+b]=a$ и $y[am+b]=b$.

Броят на всички правоъгълни триъгълници, върховете на които са възли, може да се намери като се изберат по всички възможни начини три възела и се провери дали избраните точки са върхове на правоъгълен триъгълник.

Младен Манев