

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА СТЕПЕН

Стандартната функция $\text{pow}(a, n) = a^n$ не пресмята точно, защото a и n са от тип `double`.

Наивен метод за повдигане на степен с n на брой умножения :

```
f = a;  
for (i = 2; i <= n; i++)  
f = f * a;
```

Тъй като според условието на задачата n е голямо число, то и самата степен става много голяма.

За по-ефективно и бързо пресмятане степенния показател се разлага в двоична бройна система като двоично число. За да пресметнем a^b , представяме

$$b = b_0 + b_1 2 + b_2 2^2 + b_3 2^3 + \dots + b_n 2^n,$$

където всяко b_i е 0 или 1. Тогава

$$a^b = a^{b_0} \times (a^{b_1})^2 \times (a^{b_2})^{2^2} \times \dots \times (a^{b_n})^{2^n}$$

Двоичните цифри на b се пресмятат с цикъла: `while(b>0){cout << b%2; b = b/2;}` Едновременно се намира степента :

```
int a, b; cin >> a >> b;  
int p=1; int c=a;  
while(b>0)  
{if(b%2) p = p * c;  
  b = b/2; c = c * c;  
}  
cout << p << endl;
```

По време на цикъла променливата c последователно съдържа: $a, a^2, a^4, a^8, a^{16}, \dots$, но само тези от тях се умножават, за които съответната двоична цифра b_i не е нула.

В задачата се иска да се получат степените по модул m :

```
long long stepen(long long a, long long n, long long m)  
{  
  long long f=1, c=a;  
  while(n>0)  
  {if(n%2) f = (f * c)%m;  
    n = n/2; c=(c*c) % m;  
  }  
  return f % m;  
}
```

Автор: Зорница Джанкова