

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Kod binarny Fibonacciego – LOGIA 22 (2021/22), etap 3

Treść zadania

W systemie binarnym do zapisu liczb wykorzystywane są tylko dwie cyfry: 0 i 1. Aby obliczyć wartość dziesiętną liczby zapisanej w naturalnym kodzie binarnym wystarczy wymnożyć jej cyfry przez odpowiednie potęgi liczby 2 i dodać wyniki.

Na przykład: $10100_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 16 + 4 = 20$

Tosia postanowiła zamiast potęg liczby 2 wykorzystać liczby Fibonacciego. Kolejna liczba Fibonacciego jest sumą dwóch poprzednich. Przyjęła za dwie pierwsze liczby 1 i 2. Otrzymała w ten sposób liczby Fibonacciego 1, 2, 3, 5, 8, 13, 21, ...

Na przykład: $10100_F = 1 \cdot 8 + 0 \cdot 5 + 1 \cdot 3 + 0 \cdot 2 + 0 \cdot 1 = 8 + 3 = 11$

Tosia zrobiła ponadto założenie, że w kodzie binarnym Fibonacciego dwie jedynki nie mogą wystąpić obok siebie, ponieważ zapis nie byłby jednoznaczny (np. $11_F = 1 \cdot 2 + 1 \cdot 1 = 3$,

$100_F = 1 \cdot 3 + 0 \cdot 2 + 0 \cdot 1 = 3$).

Poniższa tabela przedstawia zapis kolejnych liczb w kodzie binarnym Fibonacciego.

Liczba dziesiętna	Kod binarny Fibonacciego
1	1
2	10
3	100
4	101
5	1000
6	1001
7	1010
8	10000
9	10001
10	10010

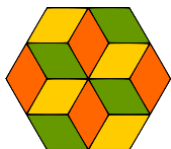
Pomóż Tosi i napisz program, który zamienia liczbę dziesiętną na kod binarny Fibonacciego.

Wejście:

Liczba naturalna składająca się maksymalnie z 1000 cyfr.

Wyjście:

Zapis liczby z wejścia w kodzie binarnym Fibonacciego.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

	Przykład 1	Przykład 2	Przykład 3
Wejście	3	11	16
Wyjście	100	10100	100100

Omówienie rozwiązania

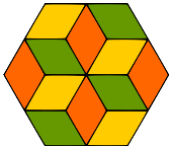
Główny problem w zadaniu to generowanie liczb Fibonacciego. Można zastosować jedno z dwóch podejść: rekurencyjne lub iteracyjne. Rekurencyjne generowanie liczb Fibonacciego jest obciążone ograniczeniami. Jesteśmy w stanie w skończonym czasie uzyskać jedynie niewielkie liczby (do 10 cyfr). W zadaniu daną może stanowić liczba naturalna składająca się z maksymalnie 1000 cyfr. Jeśli nasze zadanie ma uporać się z tak dużymi liczbami musimy postawić na rozwiązanie iteracyjne.

```
lista[0] ← 1
lista[1] ← 2
i ← 1
dopóki lista[i - 1] + lista[i] ≤ liczba wykonuj
    i ← i + 1
    lista[i] ← lista[i - 2] + lista[i - 1]
```

Ostatnia wygenerowana liczba Fibonacciego jest nie większa niż liczba, którą poddajemy przekształcaniu.

Liczbę na kod binarny Fibonacciego będziemy zamieniać metodą zachłanną. Przeglądanie liczb Fibonacciego rozpoczynamy od największej wygenerowanej liczby, czyli ostatniego elementu listy. Odpowiada ona najbardziej znaczącej jedynce w binarnym kodzie Fibonacciego. Do dalszego procedowania bierzemy różnicę przekształcanej liczby i liczby Fibonacciego. W kolejnych krokach szukamy pierwszej liczby Fibonacciego równej lub mniejszej aktualnej wartości przekształcanej liczby. Pominięte liczby Fibonacciego odpowiadają zerom w kodzie. W momencie, gdy różnica osiągnie wartość zero, należy wypisać wszystkie zera, które odpowiadają pozostałym liczbom Fibonacciego.

```
dopóki liczba > 0 wykonuj
    jeśli lista[i] ≤ liczba:
        wypisz 1
        liczba ← liczba - lista[i]
    w przeciwnym przypadku
        wypisz 0
    i ← i - 1
dopóki i >= 0 wykonaj
    wypisz 0
    i ← i - 1
```

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

01001010101010010100000010100000001000000100100000010101010100000001000010010000
000010000010000000010100100010010000001010100100010001010101000101001000100001000
100010000001010100010010101000100001001001000100000100101010010100101000100010100
101000100100010010001000001010010000101010000100000001000000001010101000100
10100010000000101000010100100000001000000000100101000000001000100010010001000100
01000100000001000010000010000100100000000010010100010100000000010001000100000100
0100100101000001000000001010000010100101010000101010001010100100101000000001000
00101000010000100100100000010101010010000010101000010100001000010010100000100010
0010101010100000101000000010001000101000100000001010000000101000010101000010001
001000100010000001001010101000010101000101010000001010100100101000010101000001010
010000100000000100101000101000101001001010001000010001010010100010001001000000100
101001000101010010010010001010000000101010100001000001000001001010100001010001010
0010100010010001010100100001001000001000101000001001000100000100100101010101000
0010000000100010001001000100000010010010010001010010101000000000000000100010100010
100100010000101010000010000010101000000100100001001000100010101000010100001010010
000000010101010000101000010000010010000001001010101010000100001000100100001001001
00100010100100100100100100001001000100001010001001000001001001001001001001001000
10100100001010100000101010100010001000000001001010000000000000001001010100001010
000101