Постановка задачі

Легендарні Щек, Хорив та Кий вирішили влаштувати змагання своїх лічилок.

Лічилка Щека була створена на основі історії Йосифа Флавія, коли по колу знаходяться **N** людей і кожного другого виводять з кола. Номер людини, що залишиться, і буде результатом лічилки. Наприклад, коли в колі знаходяться 5 людей, то з кола будуть виводитись люди в такому порядку їх номерів – 2, 4, 1, 5 і результатом стане номер 3.

Лічилка Хорива базувалась зовсім на іншому принципі. Він брав число **N** і виписував підряд всі числа від **1** до **N**. Потім викреслював спочатку ті, які знаходяться на непарних позиціях. Далі шикував їх заново, але уже викреслював ті, що знаходяться на парних позиціях. Ці дії повторював до тих пір, доки не залишиться одне число, що і буде результатом. Наприклад, для N = 5 спочатку викреслюються числа з непарними номерами – 1, 3, 5, далі із чисел, що залишилися - 2, 4 викреслюється число, яке стоїть на парній позиції, тобто 4. Тому результатом буде 2.

Лічилка Кий базувалась такому принципі. Він брав число **N** і виписував підряд всі числа від **1** до **N**. Потім викреслював спочатку ті, які знаходяться на парних позиціях. Далі шикував їх заново, але уже викреслював ті, що знаходяться на непарних позиціях. Ці дії повторював до тих пір, доки не залишиться одне число, що і буде результатом. Наприклад, для N = 5 спочатку викреслюються числа з парними номерами – 2 та 4, далі із чисел, що залишилися – 1, 3, 5 викреслюються ті, які стоять на непарних позиціях, тобто 1 та 5. Тому результатом буде 3.

Для повної об’єктивності визначення переможця було вирішено провести змагання лічилок для кожного натурального значення від **1** до **N**. Якщо в результаті для якогось значення результат лічилки першого гравця виявиться більшим, ніж результат лічилки другого, то перший отримає одне очко, якщо меншим, то одне очко отримає другий, у випадку нічиї – поточний рахунок не зміниться.

Завдання. Потрібно для заданого числа **N** $\mathbf{(1<N<10^{18})}$ визначити рахунок гри Щека та Хорива. Приклад:

N	**Результат**
10	3 6
100	48 51

Примітка. В першому прикладі лічилка Щека переможе лише при значеннях 3, 5 та 7, при значенні 1 буде нічия а в решті випадків перемогу здобуде лічилка Хорива.

Розв’язання задачі

Спочатку дослідимо закономірність в першій лічилці. Для цього скористаємось схемою, яка запропонована в [9], та удосконалимо її. Спочатку розглянемо, як можна зменшити розмірність задачі у два рази при парному значенні **N**.

1	~~2~~	3	~~4~~	5	~~6~~	7	~~8~~	9	~~10~~
1		2		3		4		5	

Рис. 2. Моделювання задачі Йосифа Флавія при парному значенні **N.**

З рисунку 2 видно, що розмірність задачі зменшилась у два рази і формула переходу від старих до нових значень буде мати вигляд

$$T(N) = 2 * T\left(\frac{N}{2}\right) - 1.$$

Для непарного значення **N** використаємо цю саму схему та відмітимо, що значення 1 в розглянутому випадку ніколи не буде розв’язком (рис. 3).

~~1~~	~~2~~	3	~~4~~	5	~~6~~	7	~~8~~	9	~~10~~	11
		1		2		3		4		5

Рис. 3. Моделювання задачі Йосифа Флавія при непарному значенні **N.**

Знову бачимо, що розмірність задачі зменшилась у 2 рази і легко вивести формулу переходу від старих до нових значень , що матиме вигляд

$$T(N) = 2 * T\left(\frac{N}{2}\right) + 1.$$

Підсумовуючи, наводимо повну схему перерахунку

$$T(N) = \begin{cases} 1, \text{якщо } N = 1; \\ 2 * T\left(\dfrac{N}{2}\right) - 1, \quad \text{якщо } N - \text{парне}; \\ 2 * T\left(\dfrac{N}{2}\right) + 1, \quad \text{якщо } N - \text{непарне}. \end{cases}$$

Але, навіть, і цієї формули недостатньо, щоб справитись із задачею в цілому. Тому випишемо розв’язки для обох лічилок для значень **N** від 1 до 30 (Рис. 4).

1	2	3	4	5	6	7	8	9	10
1	1	3	1	3	5	7	1	3	5
1	2	2	2	2	6	6	6	6	6
Нічия	Хорив	Щек	Хорив	Щек	Хорив	Щек	Хорив	Хорив	Хорив

11	12	13	14	15	16	17	18	19	20
7	9	11	13	15	1	3	5	7	9
6	6	6	6	6	6	6	6	6	6
Щек	Щек	Щек	Щек	Щек	Хорив	Хорив	Хорив	Щек	Щек

21	22	23	24	25	26	27	28	29	30
11	13	15	17	19	21	23	25	27	29
6	22	22	22	22	22	22	22	22	22
Щек	Хорив	Хорив	Хорив	Хорив	Хорив	Щек	Щек	Щек	Щек

Рис. 4. Таблиці для попереднього дослідження.

Із рисунку 4 можна зробити таке спостереження: в першій лічилці при **N**, яке є степенем двійки, завжди відповідь 1, а при наступних **N** відповідь кожного разу збільшується на 2. Тобто, якщо найближча до **N** степінь двійки буде рівна 2^k, то відповідь легко визначається за формулою

$$T(N) = 1 + 2(N - 2^k). \qquad (1)$$

Також відмітимо, що значень степенів двійки для вхідних значень від 1 до 10^{18} буде всього 60.

Перейдемо до аналізу другої лічилки. Із рисунку 4 бачимо, що кількість відповідей є незначною. Причому видно, що для 1 відповіддю буде 1, далі від 2 до 5 відповіддю буде 2, а від 22 і до наступного значення, яке потрібно дослідити – відповіддю буде 22.

Промоделюємо дану задачу і випишемо значення відповідей, які в ній зустрічаються. Це будуть наступні значення

```
1, 2, 6, 22, 86, 342,1366,    5462,    21846,
87382, 349526, 1398102,   5592406,22369622,…
```

Маючи ці значення перед собою, легко визначити схему їх обчислення

$$P(k) = 4*P(k\text{-}1) - 2, \text{ де } P(1) = 1. \qquad (2)$$

Також зрозумілим є те, що на проміжку $[P(k), P(k+1) - 1]$ відповіддю буде значення $P(k)$. Більше того, таких значень буде дуже мало. Так, на проміжку від 1 до 10^{18} їх буде лише 31.

Таким чином, однією з схем розв'язку могла бути наступна. На кожному проміжку від числа **N**=$2^k + 1$ до **N**=2^{k+1} дивимось, скільки чисел послідовності 2 він містить і відповідно враховуємо це при формуванні рахунку разом з послідовністю 1. Звичайно, останній проміжок буде йти лише до числа **N**.

Описаний варіант буде мати таке рішення на мові програмування С++:

```
#include <iostream>
using namespace std;
int main()
{
   long long N;
   cin >> N;
   long long p = 1, q = 2, Choriv = 0;
   while (2 * p + 1 <= N)
   {
      p = 2 * p + 1;
      if (p > 4 * q - 2)
      {
         long long pp = p;
         while (pp > 4 * q - 3)
```

```
            pp = (pp - 1) / 2;
         pp = 2 * (4 * q - 3 - pp) - 1;
         if (pp > q) Choriv += (pp - q) / 2 + 1;
         q = 4 * q - 2;
      }
      Choriv += (p - q) / 2 + 1;
   }
   if (N >= 4 * q - 2)
      {
         long long pp = p;
         while (pp > 4 * q - 3)
            pp = (pp - 1) / 2;
         pp = 2 * (4 * q - 3 - pp) - 1;
         if (pp > q) Choriv += (pp - q) / 2 + 1;
         q = 4 * q - 2;
      }
   p = 2 * (N - p) - 1;
   if (p > q) Choriv += (p - q) / 2 + 1;
   cout << Choriv << " " << N - Choriv - 1 << endl;
}
```

Але продовживши дослідження, можна отримати простішу схему розв'язання розглянутої задачі.

Але продовживши дослідження, можна отримати ще простішу схему розв'язання розглянутої задачі.

Відмітимо, що в цій грі лише при значенні **N**=1 обидва гравці зіграють в нічию. При всіх інших значеннях **N** виграє або перший, або другий. Тому переведемо результати гри у формат 0-1. Запишемо послідовність в якій *i*-тий елемент рівний 1, якщо другий гравець виграє і 0 – в іншому випадку, починаючи з гри для **N**=2:

```
1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1,
      0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1,
      1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
      1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
```

Далі запишемо кількості однакових чисел, що йдуть підряд:

`1, 1, 1, 1, 1, 1, 3, 5, 3, 3, 5, 5, 11, 21, 11, 11, 21, 21, ...`

Наступним кроком можна розділити цю послідовність на блоки по 6 елементів і помітити, що кожен блок складається з двох послідовних чисел Якобшталя

`1, 1, 3, 5, 11, 21, 43, 86, 171, ...`

Тобто новою схемою розв'язання розглядуваної задачі буде така. Кожен раз, вибираючи по два значення послідовності Якобшталя, будемо формувати поточний рахунок. Звісно, при формуванні остаточного рахунку, будемо використовувати лише ті значення в останній шістці, які обмежуються кількістю партій **N**.

Наприклад, обчислимо рахунок при кількості партій **N**=40. Відмітимо, що для цього знадобиться використати перші дві шістки та частину першого значення третьої шістки. Рахунок після 31 партії буде описуватись підсумовуванням елементів перших двох шісток. Для підрахунку кількості виграних партій першим гравцем рахуємо елементи на парних позиціях 1+1+1+5+3+5=16, а для підрахунку кількості виграних партій другим гравцем рахуємо елементи на непарних позиціях 1+1+1+3+3+5=14. Потім врахуємо, що третя шістка починається з 11 перемог другого гравця, з яких нам потрібно врахувати 9. Тобто, завершальний рахунок буде 16 : 23 на користь другого гравця.

Описаний розв'язок на мові програмування С++ буде мати досить простий і компактний вигляд:

```
#include <iostream>
using namespace std;
long long J[61], N, Choriv, Shchek, rem;

void Score (long long &X, long long Y)
{
```

```
    if (rem > Y) { X += Y; rem -= Y;}
        else { X += rem; rem = 0;}
}

int main()
{
    cin >> N;
    int i = 1;
    J[1] = 1; J[2] = 1;
    while (Shchek + Choriv + 3 * (J[i] + J[i + 1]) < N)
    {
        Shchek += 2 * J[i] + J[i + 1];
        Choriv += J[i] + 2 * J[i + 1];
        i += 2;
        J[i] = J[i - 1] + 2 * J[i - 2];
        J[i + 1] = J[i] + 2 * J[i - 1];
    }
    rem = N - 1 - Shchek - Choriv;
    Score(Shchek, J[i]);
    Score(Choriv, J[i + 1]);
    Score(Shchek, J[i]);
    Score(Choriv, J[i]);
    Score(Shchek, J[i + 1]);
    Score(Choriv, J[i + 1]);
    cout << Choriv << " " << Shchek;
}
```

Відмітимо, що в даній задачі кількість задіяних чисел Якобшталя не перевищує 60. Процедура Score дозволяє реалізувати кінцівку в цій задачі, коли останній блок не буде задіяний повністю.

Таким чином розглянута задача (підвищеної складності), яка використовувалась на міжнародному змаганні, є поєднанням зовсім під новим ракурсом, який досі не використовувався, двох відомих задач. Наведено два варіанти її розв'язання. У більш гнучкому і ефективному варіанті достатньо було знайти закономірність, яка задається елементами послідовності Якобшталя.

Завдання до лабораторної роботи №7

Завдання 1. Потрібно для заданого числа **N ($1<N<10^{18}$)** визначити рахунок гри Щека та Кия.

Завдання 2. Потрібно для заданого числа **N ($1<N<10^{18}$)** визначити рахунок гри Хорива та Кия.

Завдання 3. Потрібно для заданого числа **N ($1<N<10^{18}$)** визначити рахунок гри, коли одночасно грають троє (!!!) Щек, Хорив та Кий.