

## Идея решения

Искать число нужно не в миллиарде, а в миллион + 1, что легко доказать.

Сначала рассмотрим отрезок  $[1, 1000001]$ . В этом отрезке обязательно содержится число, которое мы ищем. Запомним начало отрезка в переменной  $Left$ , конец –  $Right$ , середина отрезка –  $Middle (= (Left + Right) \div 2)$ . Далее определим, в какой половине этого отрезка находится это число. Определим, все ли числа из левой половины (от  $Left$  до  $Middle$  включительно) выбраны. Для этого считаем длину этого отрезка и просматриваем файл.

1. Если мы количество чисел  $X$  в файле, удовлетворяющих  $Left \leq X \leq Middle$ , меньше длины  $Middle - Left$ , то в этой половине и находится искомое число. Далее переопределяем границы –  $Left$  остаётся,  $Right = Middle$ ,  $Middle = (Left + Right) \div 2$ , т.е. мы выделили левую половину предыдущего отрезка. Потом рассматриваем новый отрезок точно так же, как предыдущий.
2. Иначе: переопределяем границы –  $Left = Middle + 1$ ,  $Right$  остаётся,  $Middle = (Left + Right) \div 2$ , т.е. мы выделили правую половину предыдущего отрезка. Потом рассматриваем новый отрезок точно так же, как предыдущий.

Отрезки рассматриваются, пока количество найденных чисел в файле из текущего отрезка не станет 0. Тогда левая граница и будет ответом.

## Алгоритм:

1. Установим правую изначальную границу  $Right = 1\ 000\ 001$
2. Устанавливаем левую границу  $Left = 1$
3. Установим середину  $Middle = (Left + Right) \div 2$
4. Связываем файловую переменную с файлом с исходными данными
5. Количество найденных чисел = 1 (для того, чтобы цикл по условию не остановился сразу)
6. Пока количество найденных чисел  $> 0$  делать
  - 6.1. Открываем файл для чтения
  - 6.2. Количество найденных чисел = 0
  - 6.3. Пока не конец файла, делать
    - 6.3.1. Считываем число  $x$
    - 6.3.2. Если  $Left \leq x \leq Middle$ , то
      - 6.3.2.1. Количество найденных чисел = Количество найденных чисел + 1
  - 6.4. Если Количество найденных чисел  $<$  длины левой половины отрезка
    - 6.4.1.  $Right = Middle$
    - 6.4.2.  $Middle = (Left + Right) \div 2$
  - 6.5. Иначе:
    - 6.5.1.  $Left = Middle + 1$
    - 6.5.2.  $Middle = (Left + Right) \div 2$
  - 6.6. Закрываем файл
7. Выводим число  $Left$

## Программа на языке Си

```
#include <stdlib.h>
#include <conio.h>
#include <iostream.h>
#include <stdio.h>
void main() {
```

```

unsigned long n,left,middle,right,col
,searched,x;
FILE *f;

clrscr();
n = right = 1000001;
left = 1;
middle = ldiv((left+right),2).quot;
col = middle-left+1;
cout <<"[" <<left <<"..." <<middle <<"..." <<right <<"]" <<"\n";
searched=10;
while (searched>0) {
    searched=0;
    f=fopen("C:/TEMP/input.TXT","r");
    for (int i=1; i<=10; i++) {
        fscanf(f,"%d\n",&x);
        if ((x>=left)&&(x<=middle))
            searched++;
    }
    if (searched<col) {
        right = middle;
        middle = ldiv((left + right),2).quot;
    }
    else
    {
        left = middle+ 1;
        middle = ldiv((left + right),2).quot;
    }
    cout <<searched <<" found ";
    col = middle-left + 1;
    cout <<"[" <<left <<"..." <<middle <<"..." <<right <<"]" <<"\n";

    fclose(f);
}
cout <<"ANSWER = " <<left;
getch();
}

```