

Задача 1. Две цифры

(Время: 1 сек. Память: 16 Мб Баллы: 100)

Сколько N-значных чисел можно составить, используя цифры 5 и 9, в которых три одинаковые цифры не стоят рядом?

Входные данные

В единственной строке входного файла INPUT.TXT записано одно число N ($1 \leq N \leq 30$).

Выходные данные

В единственную строку выходного файла OUTPUT.TXT нужно вывести одно число - количество чисел с указанным свойством.

Пример

№	INPUT.TXT	OUTPUT.TXT
1	3	6

Разбор

Задача решается методом динамического программирования. Обозначим через $f55_n$, $f59_n$, $f95_n$, $f99_n$ – количество n-значных чисел, заканчивающихся на 55, 59, 95, 99 соответственно. При $n=2$ имеем $f55_2=f59_2=f95_2=f99_2=1$. Также заметим, что $f55_n=f95_{n-1}$, $f59_n=f55_{n-1}+f95_{n-1}$, $f95_n=f59_{n-1}+f99_{n-1}$, $f99_n=f59_{n-1}$. Выписанные формулы позволяют последовательно находить введённые величины. Ответом будет величина $f55_N+f59_N+f95_N+f99_N$, а с учётом выписанных выше равенств $f55_{N-1}+2f59_{N-1}+2f95_{N-1}+f99_{N-1}$. Также надо учесть, что при $N=1$ искомым чисел 2, а при $N=2$ – 4.

Программа

```
var
  n, i : integer;
  nn, n55, n59, n95, n99 : longint;
  f1, f2 : text;
begin
  assign(f1, 'input.txt'); reset(f1);
  assign(f2, 'output.txt'); rewrite(f2);
  read(f1, n);
  if n<3 then write(f2, 2*n) else begin
    n55:=1; n59:=1; n95:=1; n99:=1;
    for i:=3 to n-1 do begin
      nn:=n59; n59:=n55+n95; n55:=n95; n95:=nn+n99; n99:=nn
    end;
    nn:=n55+2*n59+2*n95+n99;
    write(f2, nn);
  end;
  writeln(f2); close(f2)
end.
```

Задача 2. Турнир

(Время: 1 сек. Память: 16 Мб Баллы: 100)

Турнир проходит по олимпийской системе. В каждом матче участвуют два игрока. Проигравший игрок выбывает из турнира, а победитель проходит в следующий тур. Матчи продолжаются до тех пор, пока в турнире не останется один участник, который становится обладателем золотой медали. Серебро достаётся его оппоненту в финальном матче. Если количество участников больше трёх, то назначается дополнительный матч для определения бронзового победителя. В нём участвуют два игрока, выбывшие из турнира последними, не считая финалистов.

Напишите программу определения минимального количества матчей, которые необходимо сыграть участникам турнира, чтобы определить из них тех, кто получит медали.

Входные данные

В единственной строке входного файла INPUT.TXT записано одно целое число N ($0 \leq N \leq 2147483647$) – количество участников турнира.

Выходные данные

В единственную строку выходного файла OUTPUT.TXT нужно вывести одно число - минимальное количество матчей в этом турнире.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	3	2
2	4	4

Разбор

Если участников нет или один, то матчи не проводятся. При двух участниках проводится один матч, а при трёх участниках – два. При большем количестве участников количество матчей равно количеству участников. Докажем это. Для этого обозначим через $f(n)$ – количество матчей для определения только обладателя золотой медали. При чётном количестве участников в туре $n=2k$ в следующий тур проходит k из них, при нечётном количестве $n=2k+1$ – $k+1$. При этом играется k матчей. Таким образом, имеем $f(2k)=k+f(k)$, $f(2k+1)=k+f(k+1)$, $f(1)=0$. Подстановкой в каждое из полученных соотношений легко проверить, что $f(n)=n-1$. Так как проводится ещё дополнительный матч для определения бронзового победителя, то общее количество матчей равно количеству участников.

Программа

```
var
  n : longint;
begin
  assign(input, 'input.txt'); reset(input);
  assign(output, 'output.txt'); rewrite(output);
  read(n);
  if n<2 then write(0) else
    if n<4 then write(n-1) else write(n);
  close(output)
end.
```

Задача 3. Укладка плитки

(Время: 1 сек. Память: 16 Мб Баллы: 100)

Бригаде строителей поручили уложить квадратной плиткой пол на кухне в виде шахматного узора. Но строители работали не очень слаженно, и когда весь пол уже был уложен, оказалось, что в некоторых местах плитки одинакового цвета граничат друг с другом.

По заданному замощению определите, какое минимальное число строителей могло укладывать плитку.

Входные данные

Входной файл INPUT.TXT содержит восемь строк, состоящих из восьми символов W и B – полученное замощение. Символ W обозначает плитку белого цвета, а символ B – чёрную.

Выходные данные

В единственную строку выходного файла OUTPUT.TXT нужно вывести одно число - искомое число строителей.

Пример

№	INPUT.TXT	OUTPUT.TXT
1	WBWBWBWB BWBBWBWB WBWWBWBW WBWWBWWB BWBBWBWB WBWBWWBW BWBWBWBW WBWBWWBW	4

Разбор

Просматривая заданное замощение, находим ещё не удалённые плитки и, найдя одну из них, увеличиваем количество строителей на один, а также удаляем её и все плитки, положенные правильно относительно неё. Для осуществления последнего действия применяем рекурсию.

Программа

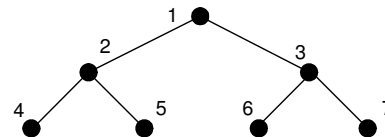
```
var
  i, j, k : integer;
  a : array [0..9,0..9] of char;
procedure U(i,j:integer);
  var c : char;
begin
  c:=a[i,j]; a[i,j]:=' ';
  if (c='W')and(a[i+1,j]='B') then U(i+1,j);
  if (c='B')and(a[i+1,j]='W') then U(i+1,j);
  if (c='W')and(a[i-1,j]='B') then U(i-1,j);
  if (c='B')and(a[i-1,j]='W') then U(i-1,j);
  if (c='W')and(a[i,j+1]='B') then U(i,j+1);
  if (c='B')and(a[i,j+1]='W') then U(i,j+1);
  if (c='W')and(a[i,j-1]='B') then U(i,j-1);
  if (c='B')and(a[i,j-1]='W') then U(i,j-1);
end;
begin
  assign(input,'input.txt'); reset(input);
  assign(output,'output.txt'); rewrite(output);
  for i:=1 to 8 do begin
    for j:=1 to 8 do read(a[i,j]);
    readln
  end;
  for i:=1 to 8 do begin
    a[0,i]:=' '; a[9,i]:=' ';
    a[i,0]:=' '; a[i,9]:=' '
  end;
  k:=0;
  for i:=1 to 8 do
    for j:=1 to 8 do
      if a[i,j]<>' ' then begin
        k:=k+1; U(i,j)
      end;
  write(k);
  close(output)
end.
```

Задача 4. Эволюция

(Время: 1 сек. Память: 16 Мб Баллы: 100)

Во время исследований, посвященных появлению жизни на планете Олимпия, учеными было сделано несколько сенсационных открытий:

1. Все живые организмы планеты происходят от бактерии Bitozoria Programulis.
2. Эволюция происходила шаг за шагом (по предположению ученых – во время изменения климата на планете).
3. На каждом шаге эволюции из каждого вида образовывались ровно два подвида, а предыдущий вид исчезал.
4. Если считать появление бактерии Bitozoria Programulis первым шагом эволюции, то существующие сейчас живые организмы находятся на N -ом шаге.



Чтобы не придумывать названия во время исследований, ученые пронумеровали все виды организмов, которые когда-либо существовали на планете. Для этого они нарисовали дерево эволюции с корнем Bitozoria Programulis, которая получила номер 1. Далее нумеровали виды каждого шага эволюции слева направо. Таким образом непосредственные подвиды Bitozoria Programulis получили номера 2 и 3. Следующими были занумерованы виды третьего шага эволюции – подвиды вида 2 получили номера 4 и 5, а вида 3 – номера 6 и 7, и т.д.

Напишите программу, которая по номерам двух видов вычислит номер вида их ближайшего общего предка в дереве эволюции.

Входные данные

В первой строке входного файла INPUT.TXT записано целое число N ($1 \leq N \leq 60$) – количество этапов эволюции, которые произошли на планете Олимпия до текущего времени. Вторая и третья строки содержат по одному натуральному числу, которые представляют номера видов, для которых требуется найти номер их ближайшего общего предка.

Выходные данные

В единственную строку выходного файла OUTPUT.TXT нужно вывести одно натуральное число – номер ближайшего предка для двух видов.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	4 15 12	3
2	18 233016 233008	14563

Разбор

Описанный в задаче способ нумерации видов позволяет легко находить по номеру подвида номер вида, из которого произошёл этот подвид. Достаточно номер подвида нацело поделить на два. Таким образом, будем перемещаться по дереву эволюции, пока не найдем совпадение номеров видов-предков для двух заданных видов.

Программа

```

var
  n, a, b : int64;
begin
  assign(input, 'input.txt'); reset(input);
  assign(output, 'output.txt'); rewrite(output);
  read(n, a, b);

```

*Муниципальный этап Всероссийской олимпиады школьников по информатике
Ханты-Мансийский автономный округ – Югра
2010-2011 учебный год
9-11 классы*

```
while a<>b do
  if a>b then a:=a div 2 else b:=b div 2;
write(a);
close(output)
end.
```