# **1,2,3 – это условие заданий**

# **Первый ход делает Петя, второй ход – Ваня.**

# **За один ход в кучу можно положить 1 или 4 камня или увеличить кучу в 2 раза. (S+1 или S+4 или S\*2)**

# **Победителем считается игрок, сделавший последний ход, т. е. первым получивший кучу, в которой будет 48 или больше камней.**

# №19 Найдите минимальное значение S, при котором Ваня может выиграть на своем первом ходе.

№20 Найдите два значение S, при которых выполняется два условия

— Петя не может выиграть за один ход;

— Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

№21 Найдите минимально значение S, при котором выполняется два условия

— у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;

— у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

# **Решение при помощи таблицы**

**Решение задачи №19**

Ходом Вани должно быть «\*2», т.к. ему нужно выиграть на своем первом ходе, следовательно, рассматривать другие возможные ходы бесполезно.

48/2 = 24 (Это выигрышная позиция для Вани)

24-1 = 23 (Это проигрышная позиция для Пети)

Минимально значение S, с которого Ваня может получить выигрышную позицию 🡪 24/2=12 – это ответ на эту задачу.

Составим таблицу, в которой «В1» обозначает выигрыш за один ход, «П1» обозначает проигрыш за один ход.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | … | 47 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | П1 | В1 | … | В1 |

**Решение задачи №20**

Для того, чтобы Петя смог выиграть своим вторым ходом, ему нужно своим первым ходом перевести игру в проигрышную (для Вани) позицию, т.е., получить 23 камней. Он может сделать это при S = 19 (ходом «+4») или при S = 22 (ходом «+1»).

Ответы на эту задачу – 19 и 22.

В таблице обозначим эти позиции как выигрышные за 2 хода.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | … | 47 |
|  |  |  |  |  |  |  |  |  |  |  | В2 |  |  | В2 | П1 | В1 | … | В1 |

**Решение задачи №21**

Нужно найти такие позиции, из которых **все возможные ходы** ведут в выигрышные позиции, помеченные как В1 или В2; это позиция S = 18 при S = 18 можно получить 19, 22 или 36 камней, все эти позиции выигрышные;

Ответ на эту задачу – 18.

## **Решение при помощи программирования (Java&Python)**

**Решение задачи №19**

1. Создаем функцию, которая определяет, является ли позиция выигрышная за один ход.

**(JAVA)**

*static boolean win1(int s) {  
 return s+1 >= 47 || s+4 >= 47 || s\*2 >= 47;  
}*

**(PYTHON)**

*def win1(s):  
 return s + 1 >= 47 or s + 4 >= 47 or s \* 2 >= 47*

1. Создаем функцию, которая определяет, является ли позиция проигрышная. Учитываем только умножение на 2, т.к. Ване нужно выиграть своим первым ходом. И нам не нужно учитывать другие его ходы (по условию задачи).

**(JAVA)**

*static boolean loss1(int s) {  
 return (!win1(s)) && win1(s\*2);  
}*

**(PYTHON)**

*def loss1(s):  
 return not (win1(s)) and win1(s \* 2)*

1. Делаем перебор значений через цикл.

**(JAVA)**

*public static void main(String[] args) {*

*for (int i = 0; i < 48; i++)   
 if (loss1(i)) {  
 System.out.println(i);*

*break;  
 }  
}*

**(PYTHON)**

*for i in range (0, 48):*

*if loss1(i):*

*print(i)*

*break*

Получаем ответ – 12.

**Решение задачи №20**

1. Дополняем функцию “loss1”, добавляя в нее прибавление 1 и 4 камней.

**(JAVA)**

*static boolean loss1(int s){  
 return (!win1(s)) && win1(s\*2) && win1(s+1) && win1(s+4);  
}*

**(PYTHON)**

*def loss1(s):*

*return not (win1(s)) and win1(s \* 2) and win1(s + 1) and win1(s + 4)*

1. Создаем функцию, которая определяет, является ли позиция выигрышная за два хода.

**(JAVA)**

*static boolean win2(int s) {  
 return loss1(s+1) || loss1(s+4) || loss1(s\*2);  
}*

**(PYTHON)**

*def win2(s):*

*return loss1(s \* 2) or loss1(s + 1) or loss1(s + 4)*

1. Делаем перебор значений через цикл.

**(JAVA)**

*public static void main(String[] args) {  
 for (int i = 0; i < 48; i++)   
 if (win2(i))   
 System.out.println(i);  
}*

**(PYTHON)**

*for i in range(0, 48):*

*if win2(i):*

*print(i)*

Получаем ответы – 19 и 22

**Решение задачи №21**

1. Создаем функцию, которая определяет, является ли позиция проигрышной за 1 или 2 хода.

**(JAVA)**

*static boolean loss12(int s) {  
 return win1(s+1) && win2(s+4) && win2(s\*2) ||  
 win1(s+4) && win2(s+1) && win2(s\*2) ||  
 win1(s\*2) && win2(s+1) && win2(s+4);  
}*

**(PYTHON)**

*def loss12(s):*

*return win1(s + 1) and win2(s + 4) and win2(s \* 2) or win1(s + 4) and win2(s + 1) and win2(s \* 2) or win1(s\* 2) and win2(s + 1) and win2(s + 4)*

1. Делаем перебор значений через цикл.

**(JAVA)**

*public static void main(String[] args) {  
 for (int i = 0; i < 48; i++)   
 if (loss12(i))  
 System.out.println(i);  
}*

**(PYTHON)**

*for i in range(0, 48):*

*if loss12(i):*

*print(i)*

Получаем ответ – 18.

## **Решение при помощи** **Excel**

**Решение задачи №19**

1. Делаем три столбца – S, Петя, Ваня. (Приложение 1)
2. В столбце Петя прописываем возможные ходы этого игрока. (Приложение 2)
3. В столбце Ваня прописываем его максимальный ход, т.к. ему надо выиграть. (Приложение 3)
4. В столбце S перебираем значения, до того момента, пока Ваня не выиграет хотя бы один раз – это будет минимально значение S.

Получаем ответ – 12. (Приложение 4)

**Решение задачи №20**

1. Добавляем еще один столбец – Петя
2. Создаем 3 пути развития событий. (Приложение 5)
3. Перебираем числа и рассматриваем возможные пути. Нам нужно найти путь, в котором во всех исходах побеждает только Петя.

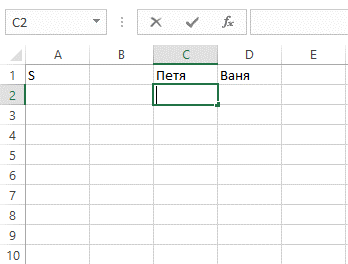
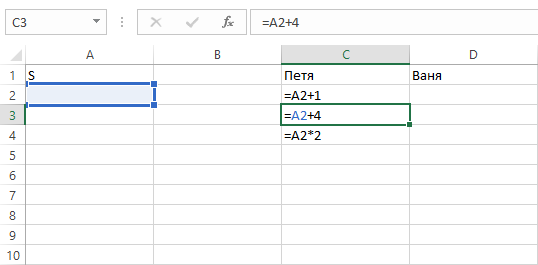
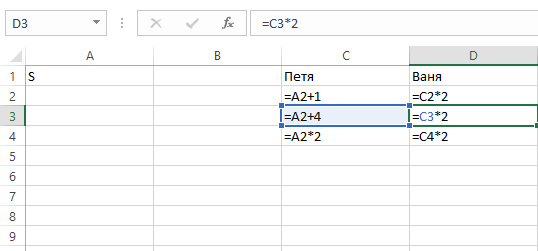
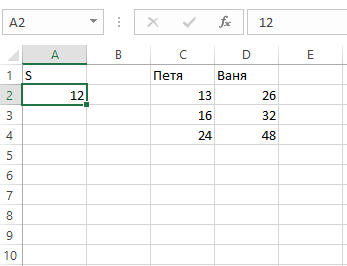
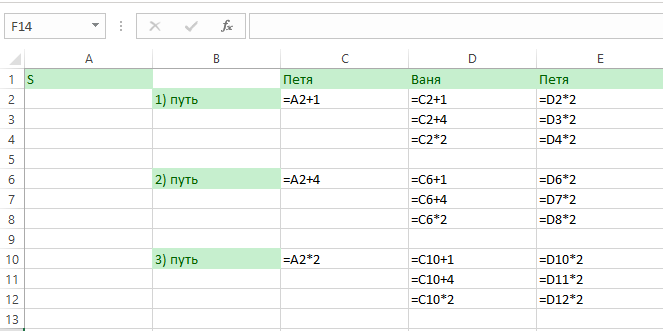
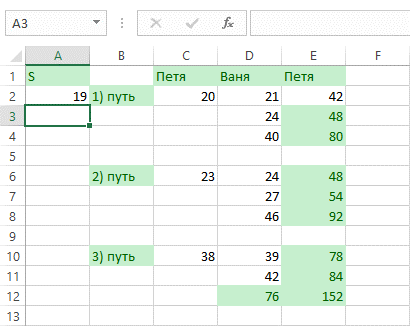
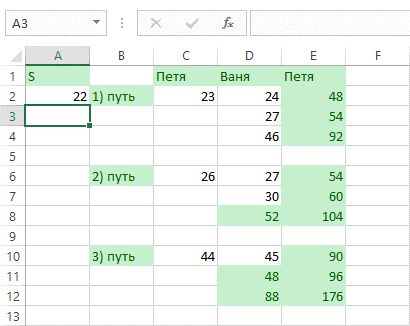
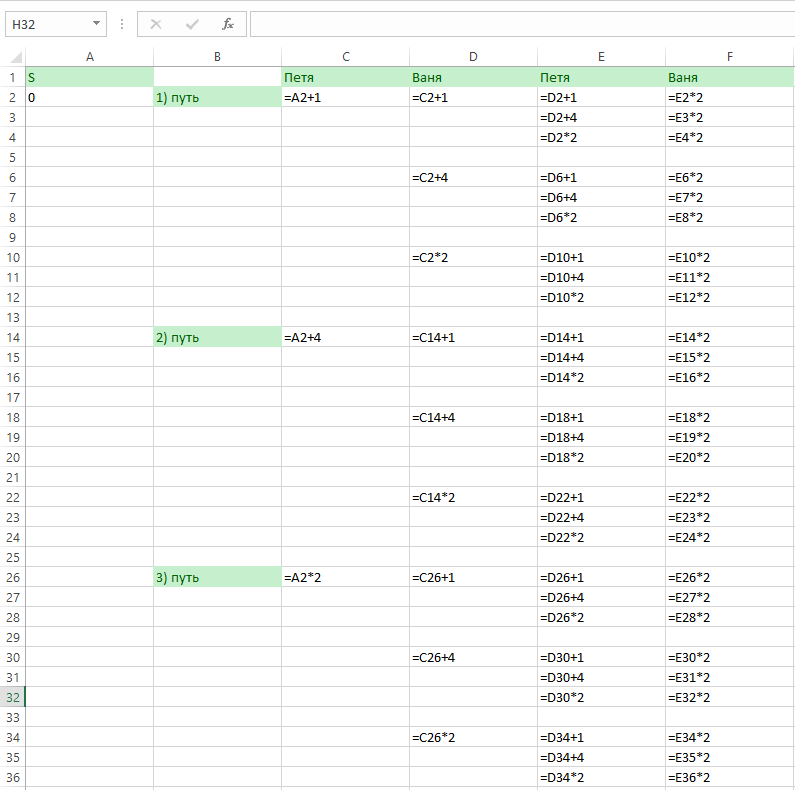
Получаем ответ - 19 и 22. (Приложение 6 и 7)

**Решение задачи №21**

1. Добавляем еще один столбец – Ваня.
2. Оставляем три пути для Пети и к этим трем путям добавляем пути для Вани. (Приложение 8)
3. Перебираем и числа и ищем такое значение, при котором в каждом пути, будет хоть один выигрышный под путь для Вани.

Получаем ответ – 18. (Приложение 9)

Приложения для экселя

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 