



Олимпиада школьников «Гранит науки»
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ

Шифр работы 1138-135
(не заполнять)

ВАРИАНТ № 6

Оценка выполнения олимпиадной работы (заполняется проверяющим)												
Первая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	—	10	7	12	7	18					54
	Σ баллов прописью	Пятидесят четыре (54)						Подпись проверяющего		ШШР -35		
								Подпись проверяющего				
Вторая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	—	4	7	12	7	18					51
	Σ баллов прописью	Пятнадцать один (51)						Подпись проверяющего		ШШР		
								Подпись проверяющего				



Вариант № 6

2. Каким образом разойдутся 13 и 9 камней. Разность этих чисел $= 4$ (четная). Рассмотрим, что будет в шах-х, если у нас останутся камни (четное; 0) и (нечетное; 1).

Пусть об.: 2 0 у х-го игрока 1; у 3-го игрока 1 0 у
0 0 $\Rightarrow$ победит игрок у.

Если об.: 3 0 у х-го
2 0
1 0
0 0 $\Rightarrow$ победит игрок х.

$\Rightarrow$ выигрывает всегда 1 игрок, если будем сокращать количество разности количества камней в 1 и 2 игроках.

Будем обозначать I, если на данном этапе ходит игрок 1, и II, если ходит игрок 2; r-разность;

13 9 r=4

I 12 8 r=4

II (11; 8) r=3 (12; 7) r=5 (11; 7) r=4

I (10; 8) (12; 6) (10; 6)

II - любой ход. $r \% 2 = 0$

I - сокращает $r \% 2 = 0$

III в результате после хода 1 игрока остается позиция (четное; 0)

Примем: 4 0

II 3 0

I 2 0

II 1 0

I 0 0

$\Rightarrow$ 1-й побеждает.



Вариант № 6

2. Если изначально $1 \times 2 = 1$, то игрок 1 можно считать выигравшим из 1 игрок 1 выигрывает, а далее продолжать соревнование.

Приведем примеры:

13	9	
12	8	I
12	7	II
11	6	I
10	6	II
9	5	I
8	4	II
7	3	I
6	3	II
5	2	I
5	1	II
4	0	I
3	0	II
2	0	I
1	0	II
0	0	I

$\Rightarrow$ победил 1 игрок.

12	4	
12	6	I
11	5	II
10	4	I
9	4	II
9	3	I
8	3	II
8	2	I
7	1	II
6	0	I

$\Rightarrow$ победил 1 игрок.

Вывод: При безразличии, игрок выигрывает, если выигрывает 1 ход.

10 баллов

108

75

-35

Вариант № 6

5. Нужно написать в языке C++ программу, которая выводит на экран все возможные комбинации из заданных чисел, сумма которых равна заданному числу.

Пример программы на языке C++:

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
signed main() {
    int n, r, d;
    cin >> n >> r >> d;
    int z = 0;
    int red = 0;
    for (int i = 1; i <= d; ++i) {
        z += (n - r) * red;
        red = r * red;
    }
    cout << red << " " << z;
}
```

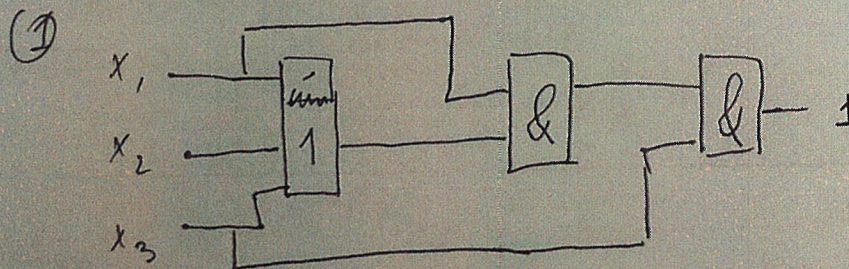
75

7. Sumel

78

по условию задан
ответ нужно
предоставить
помогая 1077

3. Схема:



Выражение

②

$$x_4 \rightarrow \bar{x}_2 \wedge x_3 \oplus x_1$$

Нужно найти значения x_1, x_2, x_3 , при которых
схема истинна, подставить их в выражение
и проверить истинность.



Вариант № 6

3. Преобразуем схему, т.е. в «исключающее или» входы 3 значения.

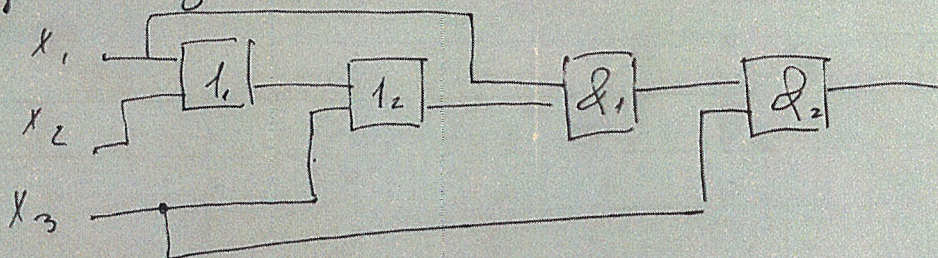


Таблица ист-ти:

x_1	x_2	x_3	1	1_2	2	2_2
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	1	1	0	0
0	1	1	1	0	0	0
1	0	0	1	1	1	0
1	0	1	1	0	0	0
1	1	0	0	0	0	0
1	1	1	0	1	1	1

← подводит.

~~Преобразуем выражение:~~

Преобразуем выражение и подставим в него значения:

$$(\bar{x}_1 + (\bar{x}_2 \wedge x_3)) \neq x_1$$

$$(0 + (0 \wedge 1)) \neq 1$$

↓
1 - истинное.

Ответ: 1.

105 750
i-1

70

Вариант № 6

3. Можно в задании найти быть решение в том, что в нем или "входит" это.

Решение:

Табл. ист-ты:

x_1	x_2	x_3	1	$\&_1$	$\&_2$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	0	0

— порядок.

Решение задачи: 1; 0; 1 — порядок.

$$(0 + (111)) \neq 1$$

↓

Ошибки: при решении в решении

4. Если в кодировке K B B кодифицирует эти значения. В условии указано, что в кодифицировании в кодифицировании можно использовать или минимальная ошибка. При этом максимальная за 1, минимальная за 0,4. Рассмотрим все случаи по рисунку.

рис. 2

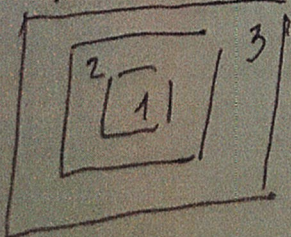
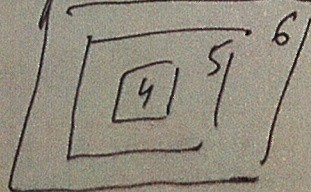


рис. 2



125



Вариант № 6

```

6. for x in range(n-fat+1):
    for y in range(m-fat+1):
        # none p-ke fat x fat
        field = [map-split#col[x+dx][y+dy] for
dx in range(fat) for dy in range(fat)]
        # проверка символов '1' в поле
        rs_on = all(map(lambda x: x == '1', field))
        if rs_on:
            good.append((x, y))
        used = set() # используемые.
        # обход в поиску
        def dfs(x, y, fat, used, good):
            used.add((x, y))
            if y == m-fat:
                return True
            for dx, dy in zip([+1, 0, -1, 0], [0, -1, 0, +1]):
                if (0 <= x+dx <= n-fat) and
(0 <= y+dy <= m-fat):
                    if (x+dx, y+dy) not in used and (x+dx, y+dy) in good:
                        a = dfs(x+dx, y+dy,
fat, used, good)
                        if a:
                            return True
            return False

```


Вариант № 6

Возможно есть и другие решения.

flag = False

for start_x in range(1 - fat + 1):

if (start_x, 0) in good and (start_x, 0)

not in used:

flag |= dfs(start_x, 0, fat, used, good)

Если есть ответ, выведем размер
квадрата и периметр. числа.

if flag:

print(fat)

break

(где именно p-р квадрата ищем все
возм-е квадратов по периметру)

Вместе с else: n, m = map(int, input().split(),

map - splitted = []

for i in range(1, n):

map - splitted.append(perim[i])

~~Уважаемый участник. Действительно код будет
работать корректно, что составит 18 баллов.~~~~Однако, комментарию не хватает описания реше-
ния в словесном или в виде блок-схемы, что
не даст дополнитель-
ных 5 баллов.~~

185