

Вариант № 3

Оценка выполнения олимпиадной работы (заполняется проверяющим)												
Первая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	0	5	5	16	7	7					40
	Σ баллов прописью	сорок (40)						Подпись проверяющего		[Подпись]		
								Подпись проверяющего				
Вторая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	0	7	5	16	7	12					47
	Σ баллов прописью	сорок семь						Подпись проверяющего		[Подпись]		
								Подпись проверяющего				

№2 Внимает наш, деловой 2-ой хог. (Т) Голоса:
 доходить до 3, не было 3) $\Rightarrow$ -1 дойти 2
 после взять отдален
 все хогы 1-ого доходить
 со мной стретин

хог 1-ого
 хог 2-ого
 ...

21
 -1 / -21 -4
 20 19 17
 -2 / -4 -2
 18 15 15
 -2 / -4 -2
 15 12 12 9 9
 $\Rightarrow$

2 хог (2-ого и 2-ого)

12
 -2 / -4 -2
 -2 / -21 -4
 6 6
 -2 / -21 -4
 5 4 2 -2
 3 3 3 3
 3 3 3 3

хог 1-ого
 хог 2-ого

9
 -2 / -4 -2
 -2 / -21 -4
 6 3 3
 -1 / -4 -2
 2 1 хог сделать
 -2 / -4 -2 хог сделать
 3 3 хог сделать

Алгоритм не пишется не алгоритмическом языке, можно
 можно добавить 2 багга исходя из решения

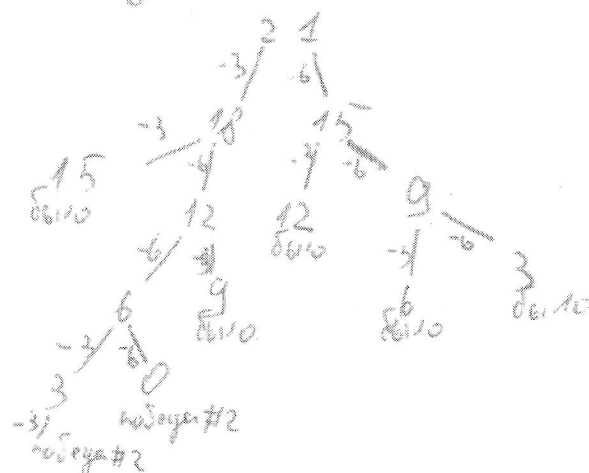
№2 Выигрывает тот, делающий 2-ой ход.
(Примечание: дополнять до 3 (или 3), когда нечего оставить
число = ходу → забрать оставшиеся
с м. след. ст.)



Вариант № 3

Так как 2-ой ход будет доминать по 3-ей-либ-ти,
То кидок ребро числа будет равно ходу 1-ого + 2-ого
дополнение

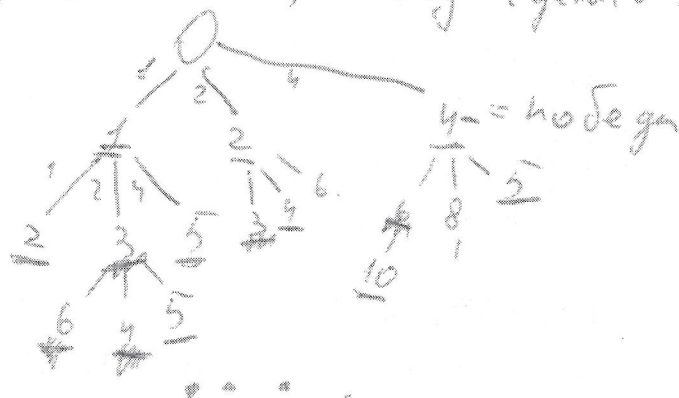
1-ый	2-ой
1	2 = 3
2	1 = 3
2	4 = 6
4	2 = 6



Алгоритм для двух количеств камней:

проигрыш = 0 камней в куче, ~~или~~ Так как есть ход = 1 = 3
если в куче ≥ 1 камней, то ход сделать можно.

последний ход



выигрывают: 4

Те, кто могут привести к: 3

или = ходу. Все остальные =
сам. = проигрыш.

Также в куче может

оказаться любое

кол-во камней: 3

для победы 2-ого.

для победы 1-ого:

Куча должно быть: 3

+ 1 или + 2 или + 4

Например для 1-ого:

23,
23,
25,
24,
22.



Вариант № 3

✓ 5

n = int(input())

m = int(input())

g = int(input())

pl = [i+1 for i in range(n)]

~~for k in range(1, m+1):~~

pr = [2, 3, 5, 7, 11]

while len(pr) < m:

def prost(k):

for i in range(3, int(k**0.5)+1):

if k % i == 0:

l = 0

break

return l

pr = [2, 3, 5, 7, 11]

y = 11

while len(pr) < m:

y += 1

if prost(y):

pr += [y]

for k in range(m):

num = pr[k]

la = pr[0]

pl = pl[1:num+1] + [la] + pl[num+1:]

if pl.index(g) == len(pl)-1:

print(pl[1:-1], pl[-1], sep='')

else:

print(pl[pl.index(g)-1], pl[pl.index(g)+1:], sep='')

проверяет, что ввоз через
функцию и ввоз
вследствие решения

все числа с 1-го н.

проверка
на простое
число

48

Словесное описание
недостаточно.
Дополните себя данными не
известными

78

как написать функцию
простое число

подсчет чисел

если последний
(миллион)



Вариант №

3

✓6

$m, n, k = \text{map}(\text{int}, \text{input}.\text{split}())$

$l, g = [1]$

$\text{for } i \text{ in range}(k):$

$l, g = [1]$

$h, l = \text{map}(\text{int}, \text{input}.\text{split}())$

$l, g = [h, l]$

$m1 = [i \text{ for } i \text{ in range}(1, m+1)]$
 $n1 = [i \text{ for } i \text{ in range}(1, n+1)]$

чтоо получить список
из списка, в котором
координаты.

это будет рекурсивная
функция.

~~$\text{def fun}(light, m, n):$
 $k = 1$
 $\text{if } (m // 2) * 2 == m:$~~

~~$up = 0$~~

~~$low = 1$~~

~~$down = 0$~~

~~$l1 = [1]$~~

~~$\text{for } i \text{ in light}:$~~

~~$\text{if } i[0] < m + (m // 2):$~~

~~$up += 1$~~

~~else~~

~~$low += 1$~~

~~$down += 1$~~

~~$\text{if } down == up:$~~

~~$\text{if } up == 1:$~~

~~$k += 1$~~

~~$\text{else}:$~~

~~$k += \text{fun}(low, m // 2, n1) + \text{fun}(l1, n // 2, n1)$~~

~~$\text{if } (n // 2) * 2 == n:$~~

~~$up = 0$~~

~~$low = 1$~~

~~$down = 0$~~

~~$l1 = [1]$~~

~~$\text{for } i \text{ in light}:$~~

~~$\text{if } i[1] < n + (n // 2):$~~

функцию

считать

визуально
показать
лог.

*

2 куса, в каждом
1 клетка

то же самое только
для разрезов вертикальных





Вариант № 3

```

left += 1
left += [i]
else:
    down += 1
    l = [i, 0], [i] - n/2
; if left left == right:
    left = 1
    k += 2
else:
    k += fun(left, m, n/2) + fun(r, m, n/2)
le + ush k

```

als = []

count = 0

vse = fun(lg, m, n)

for i in vse:

if not(i in als):

count += 1

als += i

print(count)

↑
ответ.

Важно учесть, что куски
в списке могут быть
суперпозиционными (и, следовательно,
повторяться)

def fun(left, m, n):

k = []

if (m//2) % 2 == m[-1]:

up = 0

down = 0

in p = []

11 - 5 - 1

m и n - списки
[1..m] и [1..n]
соответственно



Вариант № 3

for i in left:

if $i \leq \lfloor \frac{n-1}{2} \rfloor$:

up += 1

lup += [i]

else:

down += 1

ld += [i]

if down == up:

if up == 1:

~~k += [[[m // 2, n], [~~

k += [[m : len(m)//2, n], [m : len(m)//2, n]]

else:

k += fun(lup, m : len(m)//2, n) + fun(ld, m : len(m)//2, n)

if $(n-1) // 2 * 2 == n-1$:

left = 0

right = 0

le = []

re = []

for i in left:

if $i \leq \lfloor \frac{n-1}{2} \rfloor$:

left += 1

le += [i]

else:

right += 1

re += [i]

if left == right:

if left == 1:

k += [[m, n : len(m)//2], [m, n : len(m)//2, n]]

сдвиги
+ сдвиги
на координаты.

н/о/е/ка сдвиги

индексная, или
у изначальной
таблицы.

то же самое, но
без таблицы /-эффект



Комментарии могут
быть засчитаны в кое-ве
словесного описания



+ 5





Вариант № 3

else: $k += \text{fun}(lef, m, n, \lfloor \frac{len(mz)}{2} \rfloor) + \text{fun}($
 $\text{return } k.$

№1

исходя из: $A_4; C_4; C_4; C_6 \Rightarrow$ 1-й символ у A_{2n} ($n=2$)

исходя из: $A_3; A_6 \Rightarrow$ последний символ $A_1 = 6(2) + 1(4)$
 $E+P/O = 0$

$\Rightarrow A_1 = 10^*$

исходя из: $(9; C_{10}; D_4; E_4$

$$\text{length}(E_2) - 2 + \text{length}(D_2) - 1 = 9$$

$$\Rightarrow \text{length}(E_2) + \text{length}(D_2) = 11$$

$$(a-1)(b-1) = 20$$

$$ab - a - b + 1 = 20$$

$$ab = 19 + a + b = 19 + 11 = 30$$

Ответ:

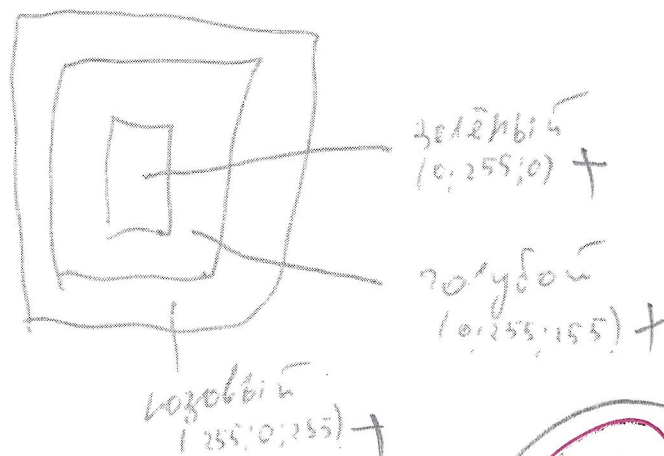
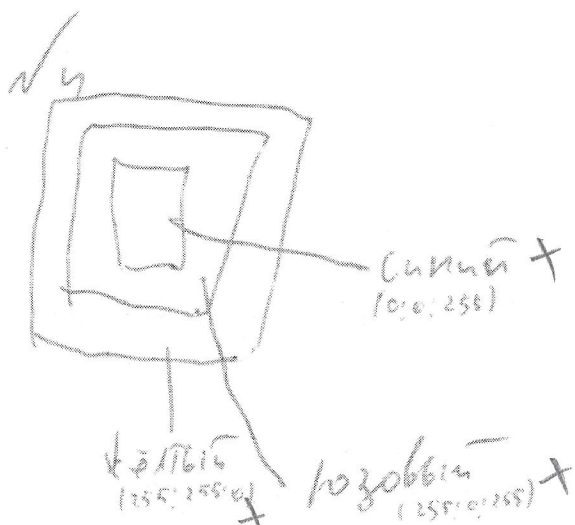
30

↓





Вариант № 3



переходы цветов + значения RGB:

синий $(0;0;255) \rightarrow$ зелёный $(0;255;0)$

голубой $(255;0;255) \rightarrow$ голубой $(0;255;255)$

жёлтый $(255;255;0) \rightarrow$ голубой $(255;0;255)$

$\Rightarrow$ переход значения цвета на левую позицию,

т.е. $R \rightarrow B$ переход $B \rightarrow R$ $G \rightarrow G$ $B \rightarrow R$ $R \rightarrow B$

✓ 3

по столбцу ✓ 5: $z \rightarrow x \Rightarrow$

z	x
0	1
1	1
0	0
1	0

$\Rightarrow 1$

$\Rightarrow$ в 3 и 4-ой строке $x=0; z=1$; g и h подходит только 1-ый столбец.

по столбцу ✓ 3 $x=1; y$

x	y
0	0
0	1
1	0
1	1

(0;1;0;2;1;1) строке или подходит для y 3 столбца

$\Rightarrow$ столбцы: $x|z|y$
(восстановлены)
по 3-м столбцам

	x	z	y
1)	0	0	0
2)	0	0	1
3)	0	1	0
4)	0	1	1
5)	1	0	0

	x	z	y
6)	1	0	1
7)	1	1	0
8)	1	1	1



Вариант № 3

1 ячейки для x может стоять и 1 и 0, т.к.

$\bar{x} \wedge y$; $y=0 \Rightarrow \bar{x} \wedge y = 0$ всегда.

$z \rightarrow x$; $z=0 \Rightarrow z \rightarrow x = 0$ всегда } + н/н-мост
но бо избежания повтора: $x=0$ / любое значение

$F(x, y) \wedge F(x, z) = 0$, н/н $x=0, z=y=1$,
в остальных $x=1$

$\Rightarrow$ между ними (?) стоит ~~некоторое~~, либо

~~н/н~~ н/н $x=y=z=0$ $F(x, y) \wedge F(x, z) = 1$

очевидно, что н/н равных y и z $F(x, y) = F(x, z)$

$\Rightarrow$ ~~н/н~~ $z, x=y$? $z=y, x$, но $x, 0 \neq x, 1$

$\Rightarrow$ ~~н/н~~ $\begin{array}{c|c} F_2 & F_2 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{array} \Rightarrow 1$ или $\begin{array}{c|c} F_2 & F_2 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{array} \Rightarrow 1$
нет решений

ответ: н/н (V)

$F(x, y) \vee F(x, z)$

1 1
2 1
...
8 1

соответственно: $F_{x,y}$ x y
 $\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{array} = \begin{array}{cc} 1 & 1 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{array} \Rightarrow y \rightarrow x$

$\Rightarrow F(y \rightarrow x) \vee F(z \rightarrow x)$

$G(x, y, z)$

$yz=1$, когда $y=z=1$.



одеют

= 1 и 1 (1/1) = 1

Вариант № 3

$(z \rightarrow y) ? (b(x, yz))$

1
0
1
0
1
1

x y z
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

7 =

58

ответ:

x z y
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

~~1~~ ~~1~~ ~~1~~

... F(x, y) $\vee$ F(x, z)

+

F(y, z) $\vee$ b(x, yz)