Олимпиада школьников «Гранит науки» Шифр работы 038-172
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ (не заполнять)

ВАРИАНТ № 5

Оценка выполнения олимпиадной работы
(заполняется проверяющим)

	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
Первая проверка	Полученный балл	–	3	0	20	18	15					56
	Σ баллов прописью	пятьдесят шесть (56)						Подпись проверяющего		Ивл		
								Подпись проверяющего				
Вторая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	–	3	0	14	18	15					50
	Σ баллов прописью	пятьдесят (50)						Подпись проверяющего				
								Подпись проверяющего				

-6,5

Олимпиада школьников «Гранит науки» Шифр работы И38-172
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ (не заполнять)

Вариант № 5

№2

Стратегия для первого игрока:
На каждом ходу делать общее количество камней делящимся на 3 (это возможно, потому что для первого хода достаточно взять 1 камень из любой кучи, а далее ~~повторять~~ дополнять ход второго игрока до трёх камней). Так как после хода первого игрока всегда остаётся кол-во камней, делящееся на 3, и второй игрок берёт 1 или 2 камня, то второй никогда не возьмёт последний камень, значит победит первый.
При бездиндексной игре выиграет первый (1)(нали)
(номер игрока)

-1 (1)	-1 (1)
-2 (2)	-2 (2)
-1 (1)	-1 (1)
-2 (2)	-2 (2)
-1 (1)	-1 (1)
-2 (2)	-2 (2)
-1 (1)	-1 (1)
-2 (2)	-2 (2)
-1 (1)	-1 (1)
-2 (2)	-2 (2)
-1 (1)	-1 (1)
-2 (2)	-1 (2)
-1 (1)	-1 (1)
первый выиграл	-1 (2)

второй победил, т.к. первый у~~пал~~ в уменьшил до такого числа, которое не делится на 3

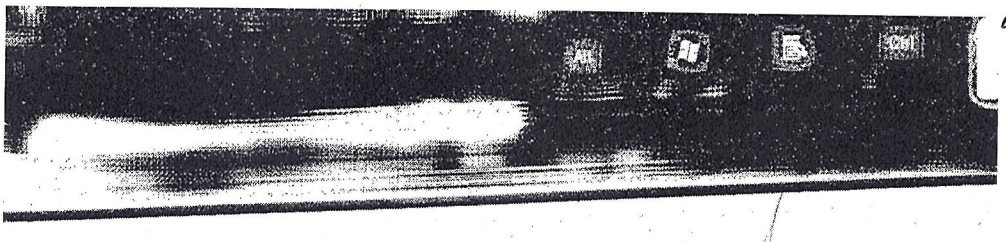
1б + стратегия не верная

1б

1+2 = 3 бал

3б

ЛИСТ 2 из 6



438-172

Олимпиада школьников «Гранит науки»
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ

Шифр работы _____ (не заполнять)

№3

Вариант № 5

60

если $\overline{X1} \to \overline{X2} \oplus X3 \vee X1 = (\overline{X1} \wedge \overline{X2}) \oplus (X3 \wedge X1)$

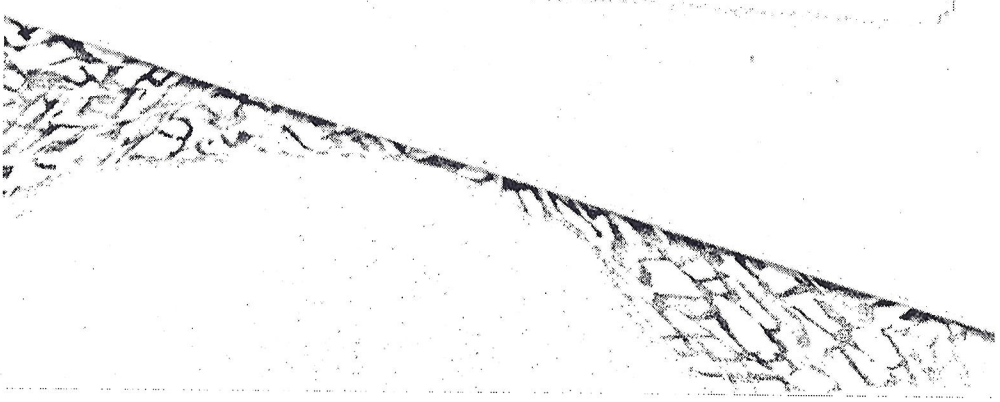
X1 - истинно ⇒ если X3 ложно, то всё значение истинно

Если X1 ложно ⇒ если X2 ложно, то всё значение истинно

В обратном случае значение выражения будет ложным

00

00



Олимпиада школьников «Гранит науки» Шифр работы 0138-172
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ (не заполнять)

Вариант № 5

№4

Алгоритм обработки изображения заключается в изменении порядка использования максимального и минимального значения яркости в каждом канале RGB для создания эффекта смешения цвета внутри квадратов при переходе от первого изображения ко второму

Значения яркости:

Рисунок 1:

Желтый (255,255,0) #FFFF00

Розовый (255,0,255) #FF00FF

Зелёный (0,255,0) #00FF00

Рисунок 2:

Синий (0,0,255) #0000FF

Зелёный (0,255,0) #00FF00

Розовый (255,0,255) #FF00FF

Цвета смешиваются с помощью инверсии каждого отдельного канала RGB

20б

20б

20б

Неверно Т.О. 14б

8б

Олимпиада школьников «Гранит науки» Шифр работы И38-172
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ

Вариант № 5

18 баллов

15 баллов

№5

```
def count_cells(N,R,D):
    MOD = 10**9 + 7
    def multiply_matrices(A,B):
        result = [[0,0],[0,0]]
        for i in range(2):
            for j in range(2):
                for k in range(2):
                    result[i][j] = (result[i][j] + A[i][k] * B[k][j]) % MOD
        return result
    def power_matrix(matrix,n):
        result = [[1,0],[0,1]]
        while n>0:
            if n%2==1:
                result = multiply_matrices(result,matrix)
            matrix = multiply_matrices(matrix,matrix)
            n//=2
        return result
    transition_matrix = [[R, N-R],[1, 0]]
    powered_matrix = power_matrix(transition_matrix, D)
    red_cells = (powered_matrix[0][0] * R + powered_matrix[0][1] *
                 * (N-R)) % MOD
    green_cells = (powered_matrix[1][0] * R + powered_matrix[1][1] *
                   * (N-R)) % MOD
    return red_cells, green_cells
# Ввод данных
N,R,D = map(int,input().split())
red_count, green_count = count_cells(N,R,D)
print(red_count, green_count)
```

Олимпиада школьников «Гранит науки» Шифр работы 4138-172
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ (не заполнять)

Вариант № 5

~~def check_robot_size~~

~ 6

```
def check_robot_size(track_width, track_length, track_layout):
    def is_valid_move(x,y):
        return 0<=x<track_lenght and 0<=y<track_width and track_layout[x][y]==
                                                          =='.'
    def can_reach_end(size):
        visited=[[ False for_in range(track_width)]for_in range(track_length)]
        q=[(0,0)]
        while q:
            x,y=q.pop(0)
            if x==track_length-1 and y=track_width-1:
                return True
            for dx,dy in[(0,size),(size,0),(0,-size),(-size,0)]:
                n_x,n_y=x+dx,y+dy
                if is_valid_move(n_x,n_y) and not visited[n_x][n_y]:
                    visited[n_x][n_y]=True
                    q.append((n_x,n_y))
        return False
    max_side_length=1
    while can_reach_end(max_side_length):
        max_side_length+=1
    return max_side_length-1

#Ввод данных
track_width, track_length=map(int,input().split())
track_layout=[input() for_in range(track_length)]
max_robot_side_length=check_robot_size(track_width,track_length,
                                       track_layout)
print(max_robot_side_length)
```

15б

15 done

15б