Олимпиада школьников «Гранит науки» Шифр работы И31-08
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ (не заполнять)

ВАРИАНТ № 6

Оценка выполнения олимпиадной работы
(заполняется проверяющим)

	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
Первая проверка	Полученный балл	–	7	2	–	25	13					47
	Σ баллов прописью	сорок семь (47)						Подпись проверяющего				
								Подпись проверяющего				
Вторая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	–	7	2	–	25	13					47
	Σ баллов прописью	сорок семь (47)						Подпись проверяющего				
								Подпись проверяющего				

1

Ответ: ~~да~~ нет Если в одной кучке 13 камней, а в другой 9, то выигрывает 1.

Рассмотрим выигрышные варианты.

ч - чётное количество камней

н - нечётное количество камней.

Проигрышный вариант это ч ч (0;0).

В вариант ч;ч можно попасть из вариантов (н;н), (ч;н); (н;ч). ⇒ Следовательно если в кучках ~~нечетно~~ во время какого хода хотя бы одно количество камней нечётное то это выигрышная позиция.

Если изначально в кучках (н;н), (н;ч); (ч;н) количество камней то выигрывает первый игрок, если (ч;ч) то выигрывает второй игрок.

Олимпиада школьников «Гранит науки» Шифр работы И31-08 (не заполнять)
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ

Вариант № 6

x - количество камней в 1 кучке
y - количество камней во 2ой кучке.

Начало
x, y
↓
Если x ⋮ 2 и y ⋮ 2 —да→ Выводим: „Выиграет 2ой гном"
нет ↓
Выводим: „Выиграет 1ый гном."

7б

7б

4+3=7 баллов
нет графа

2. 3

x_1	x_2	x_3
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

из этих вариантов верными исходя из схемы будут:

x_1	x_2	x_3
1	0	1
1	1	1

2б

2б

Подставим их в выражение $x_1 \rightarrow \overline{x2} \wedge x3 \vee x1$:

$1 \rightarrow 1 \wedge 1 \vee 1 = 1$

$1 \rightarrow 0 \wedge 1 \vee 1 = 1$

2 балла за возможно верную запись решения.

Ответ: Результирующий столбец выражения = 2.

2. 5

Ответ: 1024 1364 при входных данных ~~8 4 5~~ 8 4 5.

(Продолжение задачи на 3 листе).

Вариант № 6

```
n, r, d = map(int, input().split())  # вводим входные данные.
ar = 1  # начальное количество красных клеток.
ag = 0  # начальное количество зелёных клеток.
for i in range(d):  # повторяем операцию d раз.
    ag += ar * (n-r) % 1000000007  # находим количество добавившихся зелёных клеток и добавляем их в сумму (так как они не разместились)
    ar = ar * r % 1000000007  # находим количество красных клеток, но не суммируем их с предыдущим значением, так как они уже после размножения пропадают. И находим остаток от деления на 1000000007, чтобы не было переполнения.
print(ar, ag%1000000007)  # выводим два полученных ответа.
```

Количество красных клеток это R^D.

Количество зелёных клеток это $R^1 \cdot (N-R) + R^2 \cdot (N-R) + R^3 \cdot (N-R) + \dots$

25 баллов

25

Вариант № 6

№6

Ответ: 1 при входных данных.

Рассмотрим путь как матрицу.
Можно заметить, что существуют некоторые точки старта S и некоторые точки финиша.
Робот может ходить во все стороны, но есть смысл ходить только: вверх, вниз, вправо. Алгоритм по нахождению его пути - это граф.
Нам подойдёт алгоритм: BFS на сетке, с его помощью можно рассмотреть все возможные маршруты из начала в конец. ~~Также~~ Сначала размер робота будет равен 1 и BFS запустится от одного, если такой маршрут будет найден, то нужно будет запустить BFS от двух и так далее пока маршрут будет найден. В противном случае (если маршрут не будет найден) нужно будет вывести максимальный размер робота.

```
n, m = map(int, input().split())
g = []
for i in range(n):
```

Вариант № 6

```
~~for j in range(m):~~
    L = [int(el) for el in input().split()]
    g.append(L)
g.
start = []
dict = [[0] * n*m]
used = [[0] * n*m]
a = [1, 0, 0]
b = [0, 1, -1]
for i in range(n):
    if g[n][0] == ".":
        start.append(n)
finish = []
for i in range(n):
    if g[n][-1] == ".":
        finish.append(n)
dict[0] = 1.
used[0] = 1.
~~for i in range(~~
~~while~~
~~print~~ j = 1.
for i in range(0, 999999999):
    dict = [[0] * n*m]
    used = [[0] * n*m]
    ~~for~~ j in range():
```

135

5+1+7 = 138

5+7 = 6 баллов