Олимпиада школьников «Гранит науки» Шифр работы 127-72
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ
(не заполнять)

ВАРИАНТ № 2

Оценка выполнения олимпиадной работы
(заполняется проверяющим)

	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
Первая проверка	Полученный балл	–	9	7	–	7	7					30
	Σ баллов прописью	Тридцать (30						Подпись проверяющего		25		
								Подпись проверяющего				
Вторая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	–	9	7	–	15	15					46
	Σ баллов прописью							Подпись проверяющего				
								Подпись проверяющего				

№2.

• Заметим, что мы могли делать ходы 1, 2, 4; которые в свою очередь дают остаток 1, 2, 1 по модулю 3 соответственно.

• Позиция 3 является ~~выигрышной~~ проигрышной, т.к. ведёт в выигрышные позиции 1 или 2.

• Придумаем стратегию для первого игрока, как „дополнить" количество камней в куче до нуля по модулю трёх.

• ✗ На примере:
Первый игрок, первым ходом дополняет количество камней в куче до 0 по mod 3 => ходит либо 1 ~~или~~ либо 4.

Вариант № 2

	I	II
1	15 (-4)	—
2		13 (-2)
3	12 (-1)	8 (-4) ход после 3 ход 4.
4	6 (-2)	
5	—	5 (-1)
6	3 (-2)	—
7	—	2 (-1)
8	0 (-2)	Гранит

	I	II
1	15 (-4)	—
2	—	13 (-2)
3	12 (-1)	—
4	—	8 (-4)
5	6 (-2)	—
6	—	5 (-1)
7	3 (-2)	—
8	—	2 (-1)
9	0 (-2)	Гранит

• Второй игрок может ответить любым ходом, которые дают в итоге 1 или 2 остатка по mod 3 => ред в куче число камней было кратно трём, то станет остаток 1 или 2, а дальше первый игрок опять сможет добить остаток до нуля.

• Последним ходом, первый добьёт количество камней в куче до нуля => 0 по mod 3

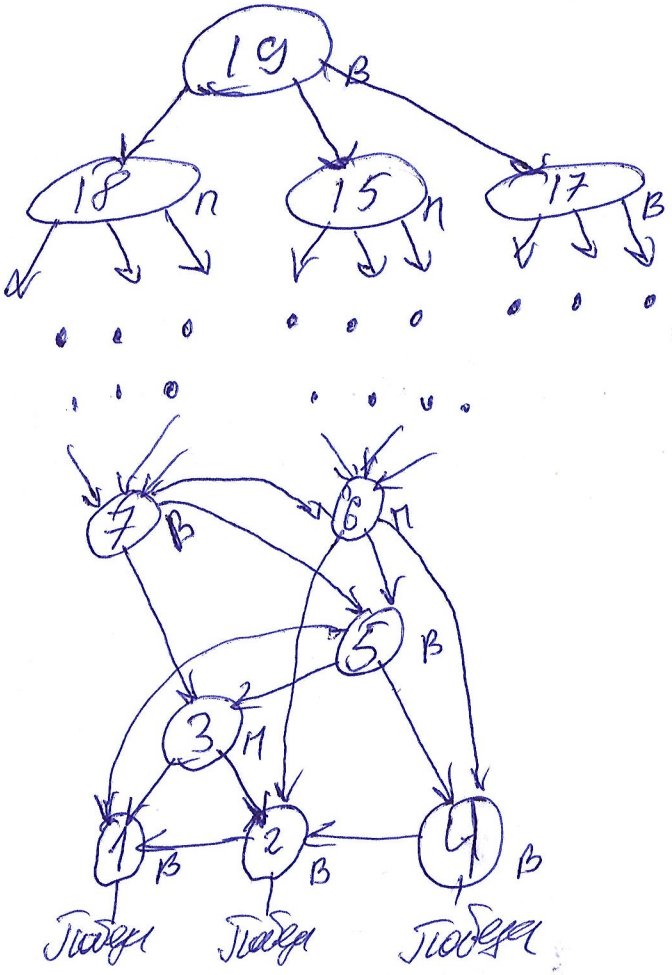
Представим алгоритмически ходы второго игрока

Вариант № 2

~~n = 19 // количество камней в куче~~
~~while n~~

```
n:=19 // - количество камне в куче
ПОКА n!=0: // - цикл ходов
| a:= n % 3 // - остаток числа n на 3
| if (a==2): // - если остаток 2, то отнимем 2 камня
|   |n:=n-2
| if (a==1): // - если остаток 1, то отнимем 1 камень
|   |n:=n-1
```

Представим ориентированный граф выигрышных / проигрышных позиций (частично), т.к. вариантов много



98
Не указано для какого кол-ва камней

Олимпиада школьников «Гранит науки» Шифр работы И27-72
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ
(не заполнять)

Вариант № 2

№2.

N	z	x	y	$x \wedge \bar{y}$	$z \vee \bar{x}$	$y \leftrightarrow z$	F(x,y) ? F(x,z)	F(y,z) ? G(x,y,z)
1	0	0	0	0	1	1	0	1
2	0	0	1	0	1	0	0	0
3	0	1	0	1	0	1	0	1
4	0	1	1	0	0	0	1	1
5	1	0	0	0	1	0	0	0
6	1	0	1	0	1	1	0	1
7	1	1	0	1	1	0	1	1
8	1	1	1	0	1	1	0	1

1. $x \wedge \bar{y}$ истино когда $x=1$ и $\bar{y}=0$ (логическое „и"); Можно однозначно заполнить x, y столбцы 1, 7.

2. Столбцу $y \leftrightarrow z$ можно однозначно заполнить зная хотя бы 2 из трех столбцов: y; z; $y \leftrightarrow z$. №3, 4, 5 - однозначно заполнены

3. Остались следующие пропуски:

	z	x	y
2	0	-	1
6	-	0	

4. №2 столбик: $z \vee \bar{x} = 1$ при $z=0$, также при $x=0$.

5. №6 столбик $y \leftrightarrow z = 1$; то $y=0$ и $z=0$, либо $y=1$ и $z=1$, т.е. два случая:

I. $y=0$ и $z=0$; $x=0$.
$x \wedge \bar{y} = 0$; $z \vee \bar{x} = 1$, $y \leftrightarrow z = 1$. Подходит

II. $y=1$; $z=1$; $x=0$
$x \wedge \bar{y} = 0$; $z \vee \bar{x} = 1$, $y \leftrightarrow z = 1$. Подходит

Вариант № 2

• Т. к. значения $x=0, y=0, z=0$ были в ТИ, то может быть только вариант $z=1; x=0; y=1$.

$F(x,y)?F(x,z) = (\bar{z} \wedge x \wedge y) \vee (z \wedge x \wedge \bar{y})$ = из таблицы истинности.

$= x \wedge (y \wedge \bar{z} \vee z \wedge \bar{y}) \underset{*}{=} x \wedge (y \leftrightarrow z) \overset{!}{=}$

$*$ - Если биты одинаковые $\Rightarrow 1$, иначе 0.

$\overset{!}{=} x \wedge (\bar{y} \vee z) = x \wedge \bar{y} \oplus x \wedge z$ - логическая и.

~~$\exists F(x,y)$~~

• Во втором случае логическое или.

Ответ: 1.) $\wedge$; 2) $\vee$.

7/5

Вариант № 2

№5.

Представим код на C++:

```
#include <bits/stdc++.h> // - импорт STL
using namespace std;
#define ll long long

int main(){
    ll n, m, q;
    cin >> n >> m >> q;
    vector<ll> primes; // - список простых
    for(int i=0; i<100000000 && primes.size() < m; i++){
        for(int j=0; j*j<=i; j++){ // - проверка на простое.
            if(i%j==0) goto next; // - если не подходит, не добавляем
        }
        primes.push_back(i);
        next:;
    }
// - получив список простых, начнем делать круги по этому.
    for(int i=0; i<primes.size(); i++){
        for(int j=0; j<primes[i]; j++){
            ll t = v[j];
            v[j] =
            ll t = v[j%n]; // - обмен
            v[j%n] = v[(j+1)%n]; // - по модулю, т.к. у нас круг.
            v[(j+1)%n] = t;
        }
    }
}
```

* vector<ll> v;
for(int i=0; i<n; i++) v.push_back(i+1);

* т.к. не знаем заранее, что будем добавлять в primes, пока размер не будет m - кол-во кругов

Олимпиада школьников «Гранит науки» Шифр работы И 27-72 (не заполнять)
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ

Вариант № 2

```
for(int i=0; i<n; i++){ // поиск элемента с номером g
    if(v[i]==g){
        cout<<v[((i-1)+n)%n]<<" "<<v[(i+1)%n];
    }
}
~~return 0;~~
}
return 0;
}
```

По критериям оценки:
+5б за словесное описание
+10б за решение с синтаксич. ошибками

15б

~~10~~

Словесное описание.

1. Составим список простых из первых m простых, проверив каждое число a за $O(\sqrt{a})$ на простоту.
2. Сделаем m кругов, где в каждом будем обменивать значения поэтапно, количество обменов будет равно текущему простому числу в цикле.

* Для того, чтобы не выйти за границы, будем обменивать числа в массиве по модулю длины массива. (Чтобы при длине массива n, V[n-1], обменялся с V[0].)

3. Линейным поиском найдём номер g, и выведем номера его соседей опять же, по модулю.

Вариант № 2

N6

Напишем программу на C++ (рекурсивную)

```
#include <bits/stdc++.h> // - имеет STL
#define ll long long
using namespace std;
// ans = 0;

// count(int x1, int x2, int y1, int y2, vector<vector<int>> v){
    // cnt = 0;
    for(int i = x1; i < x2; i++){
        for(int j = y1; j < y2; j++){
            if(v[i][j] == 0) cnt++;
        }
    }
}
void rec(int x1, int x2, int y1, int y2, vector<vector<int>> v){
    if((x2 - x1 + 1) % 2 == 0){ // - горизонтальный разрез
        int cnt1 = count(x1, (x2 - x1 + 1)/2, y1, y2, v);
        int cnt2 = count((x2 - x1 + 1)/2, x2, y1, y2, v);
        if(cnt1 != cnt2) return;
        if(cnt1 == 1){
            ans += 2;
            return;
        }
        rec(x1, (x2 - x1 + 1)/2, y1, y2, v);
        rec((x2 - x1 + 1)/2, x2, y1, y2, v);
    }
```

Вариант № 2

```
~~if ((y2 - y1 + 1) % 2 == 0;~~
else return;
if ((y2 - y1 + 1) % 2 == 0) {
    ~~int cntleft = (x1, x2, (y2 - y1 + 1)/2, y2, v)~~
    ~~int cntright = (x1, x2, (y2 - y1 + 1)/2~~
    int cntl = count(x1, x2, y1, (y2 - y1 + 1)/2, v);
    int cntr = count(x1, x2, (y2 - y1 + 1)/2, y2, v);
    if (cntl != cntr) return;
    if (cntl == 1) {
        ans += 2;
        return;
    }
    rec(x1, x2, y1, (y2 - y1 + 1)/2, v);
    rec(x1, x2, (y2 - y1 + 1)/2; y2, v);
}
else return;
}
int main() {
    int n, m, k;
    cin >> n >> m >> k;
    vector<vector<int>> v(n, vector<int>(m, 1))
    for (int i = 0; i < k; i++) {
        int x, y;
        cin >> x >> y;
        v[x][y] = 0;
    }
```

Олимпиада школьников «Гранит науки» Шифр работы И 27-72
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ (не заполнять)

Вариант № 2

```
rec(0, n-1, 0, m-1, v);
cout<<ans;
}
```

В соответствии с критерием: 10+5 = 15б

Словесное описание.

~~будем~~ Заведём функцию для подсчёта на прямоугольнике.

7б

- Будем рекурсивно ~~разбивать~~ разбивать сначала вертикальный разрез (если кол-во столбцов чётно), считаем у левого куска и правого количество торта, если одинаково разрезаем дальше, иначе заканчиваем рекурсию
- если кол-во тортов 1, то считаем в ответ +=2.
- Гор. разрез аналогично.