Олимпиада школьников «Гранит науки» Шифр работы 113Я-245
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ (не заполнять)

ВАРИАНТ № 3

Оценка выполнения олимпиадной работы
(заполняется проверяющим)

	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
Первая проверка	Полученный балл	—	0	10	—	7	7					24
	Σ баллов прописью	Двадцать четыре (24)						Подпись проверяющего				
								Подпись проверяющего				
Вторая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	—	0	10	-	18	18					46
	Σ баллов прописью	сорок шесть (46)						Подпись проверяющего				
								Подпись проверяющего				

Задание 3.

x	z	y	$\bar{x} \wedge y$	$z \to x$	$\bar{y} \vee z$	$F(x,y) \to F(x,z)$	$F(y,z) \wedge F(x,y,z)$
0/1	0	0	0	1	1	1	1
0	0	1	1	1	0	1	0
0	1	0	0	0	1	1	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	1	1
1	0	1	0	1	0	1	0
1	1	0	0	1	1	1	1
1	1	1	0	1	1	1	1
+	+	+					

108

Вариант № 3

Определим порядок переменных в столбцах. Z не может быть в первом столбце, т.к. в 3 строке Z будет равно 0 и при любом X выражение $Z \to X$ будет давать 1, но в этой строке оно равно 0, что невозможно. Пусть переменной Z соответствуют 2 столбец, тогда по строкам 3 и 5 переменной X соответствует 1 столбец. Проставив пропущенные значения, понимаем, что действительно переменной X соответствует 1 столбец, переменной Z – 2-ой, переменной Y – 3-ий. Заметим, что значение X в первой строке может равняться как 0, так и 1, т.к. $0 \to 0 = 1$, $0 \to 1 = 1$.

Определим оператор между функциями $F(X,Y)$? $F(X,Z)$, где $F(X,Y) = \overline{X} \wedge Y$, $F(X,Z) = Z \to X$. Это оператор логического следования, т.к. в столбце 7 единственное значение 0 ($1 \to 0 = 0$). То есть, $F(X,Y) \to F(X,Z)$.

Определим оператор между функциями $G(X,Y,Z)$? $F(Y,Z)$, где $F(Y,Z) = \overline{Y} \vee Z$, $G(X,Y,Z) = F(X,Y) \to F(X,Z)$. Это оператор логического умножения, т.к. в столбце 8 значение 1 получается только тогда, когда оба значения функций равны 1. Т.е., $F(Y,Z) \wedge G(X,Y,Z)$.

Олимпиада школьников «Гранит науки» Шифр работы ИЗФ-245
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ (не заполнять)

Вариант № 3

Задание 5.

```
#include <iostream>
#include <vector>
#include <algorithm>
#define ll long long
using namespace std;
// m;
vector <bool> a;
vector <ll> primes;
int resheto() {
    for (ll i=2; i*i<m+1; i++){
        if (not(a[i])) {
            for (ll j=i*2; j<m+1; j+=i) {
                a[j]=true;
            }
        }
    }
    for (ll h=2; h<m+1; h++) {
        if (not(a[h])) {
            primes.push_back(h);
        }
    }
}
int main() {
    ll n, g, p, temp, new_k;
    cin>>n>>m>>g;
    vector<bool> new_a (m+1, false);
    swap (a, new_a);
    resheto();
    vector<ll> krug (n+1);
}
```

// реализуем решето Эратосфена

// создадим вектор простых чисел

// обмен векторов, чтобы задать вектору a (глобальному) // размер и заполнить значением false

// создаем вектор, в котором будем реализовывать перемещения

В данной задаче реализовано эффективное и прав. решение на языке программирования, что соотв. 18 баллам.

К сожалению, нет словесного описания (комментарии не являются описанием) и ответа и форматирования к исходным данным.

185

Вариант № 3

```
for (ll i=1; i<n+1; i++){
    krug[i]=i; // заполняем вектор числами от 1 до n включительно
}
for (ll j=0; j<m; j++){
    p=primes[j]; //определяем j-ое простое число
    for (ll k=1; k<p+1; k++){ // запускаем цикл перемещений
        if (k==n){ //если человек последний в круге, то меняем его с первым
            temp=krug[k];
            krug[k]=krug[1];
            krug[1]=temp;
            break;
        }
        else{
            temp=krug[new_k];
            krug[new_k]=krug[new_k+1];
            krug[new_k+1]=temp;
        }
    }
}
for (ll q=1; q<n+1; q++){
    if (krug[q]==q){ //если человек последний в круге, то его правый сосед - 1 человек в круге.
        if (q+1>n){
            cout<<krug[1]<<" "<<krug[q-1];
            break;
        }
        else{if (q-1==0){ //если человек первый, то его левый сосед - последний в круге
            cout<<krug[q+1]<<" "<<krug[n];
            break;
        }
        else{
            cout<<krug[q+1]<<" "<<krug[q-1];
            break;
        }
    }
}
```

if(k>n){
new_k=k%n;
}
//если k больше n, то вычисляем остаток от деления k на n и меняем человека с этим номером
else{
new_k=k;
}

Вариант № 3

```
}
```

Задание 6.

```
#include <iostream>
#include <vector>
#define ll long long
int main(){
  ll m,n,k, x,y;   ll ans=0;
  cin>>m>>n>>k;
  vector<vector<ll>> a(n, vector<ll>(m, 0));
  for(int i=0; i<m; i++){
      for(int j=0; j<n; j++){
  for (int i=0; i<k; i++){
      cin>>x>>y;
      a[x][y]=1; //обозначим все свечи единицами
  }
//реализуем алгоритм рекурсивно, будем вызывать
функцию пока n!=0
bool v_razrez(ll n, ll m){   ll sum1=0
                             ll sum2=0     //проверка можно ли совершить вертик. разрез
  for(int i=0; i<n; i++){
      for(int j=0; j<m/2; j++){
          sum1+=a[i][j];
      }
  }
  for(int i=0; i<n; i++){
      for(int j=m/2; j<m; j++){
          sum2+=a[i][j];
      }
  }
```

75

Вариант № 3

```
if (sum1 == sum2) {
    return true)
else { return false}
bool g_razrez (ll n, ll m) {
    for (int i=0; i<n/2; i++){
        for (int j= 0; j<m; j++){
            sum1 += a[i][j];
        }
    }
    for (int i=n/2; i<n; i++){
        for (int j= 0; j<m; j++){
            sum2 += a[i][j];
        }
    }
if (sum1 == sum2) {
    return true;
}
else{ return false;}
```

// проверка можно ли сделать горизонтальный разрез

Согласен с автором в эффективности реализации: 185

```
ll sum1 = 0;
ll sum2 = 0;
while (n != 1 and m != 1) {
    if (v_razrez (n, m) and g_razrez (n, m)) { // если можно сделать оба разреза, то сделаем горизонтальный.
        n /= 2;
        ans++;
    }
    else {
        if (v_razrez(n,m)) {
            m /= 2;
            ans++;
        }
    }
```

Вариант № 3

```
        else {
            if (g-razrez(n, m)) {
                n /= 2;
                ans++;
            } else { break; } // если нельзя разрезать, то заканчиваем цикл.
        }
    }
    cout << ans;
}
```

Задание 2.

Выигрывает тот, делающий второй ход, он должен делать макс. ход (-4).

Алгоритм будет работать только с четн. кол-вом камней.

08