



Олимпиада школьников «Гранит науки» Шифр работы ИИ17-57
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ
(не заполнять)

Вариант № 7

Оценка выполнения олимпиадной работы (заполняется проверяющим)											
№ задания	1	2	3	4	5	6	7	8	9	10	Σ
Полученный балл	—	5/10	3/12	—	9/12	5/5	9			8	12/66
I проверка	Фамилия И.О. проверяющего	Заблоцкий В.Р.	Подпись	<i>В.Р.</i>	Σ баллов прописью		восемь				
II проверка	Фамилия И.О. проверяющего	Лонгинов И.И.	Подпись	<i>И.И.</i>	Σ баллов прописью						
	Фамилия И.О. проверяющего	Сошинская А.А.	Подпись	<i>А.А.</i>	Σ баллов прописью		шестнадцать				
	Фамилия И.О. проверяющего	Саранчукова	Подпись	<i>Саранчукова</i>	Σ баллов прописью		шестнадцать				

Задача 2.

- 1) При 20 камнях: первый игрок первыми ходами берёт 4 камни, а затем каждый своим ходом берёт $(5-k)$ камней, где k - число камней, взятых вторым игроком за последние его ходы. Тогда второй игрок будет вынужден взять последний камень.

ход	1	2	1	2	1	2	1	2
сколько взял	4	k_1	$5-k_1$	k_2	$5-k_2$	k_3	$5-k_3$	1

Антонина 10 баллов

- 2) При n камнях (n - чётные). Пусть q -остаток от деление n на 5. Если $q=1$, то победит второй игрок, используя схему „дополнение до 5“. Иначе первому надо взять первым ходом X камней:

если $q=2$, то $X=1$; если $q=3$, то $X=2$; если $q=4$, то $X=3$;
если $q=0$, то $X=4$.

Далее первому надо снова каждым ходом брать $(5-k)$ камней, где k - число камней, взятых вторым. Тогда первый победит.

(Этот алгоритм работает при любых комбинаторных чётных n , т. е. $n \geq 2$). *Биномиальная схема?*

Задача 5.

Поскольку списки упорядочены, то искать элементы по значению можно с помощью бинарного поиска. Код на C++:



Алешин
25.5.21 год

```
* #include <iostream>
# include <vector>
# include <string>
using namespace std;
int bin_search ( string val , vector<string> & vec ) { // Бинарный поиск
    int left = 0 , right = vec.size() , mid; // элемента по массиву строк
    while (right - left > 1) { // реализован на полуинтервалах
        mid = (left + right) / 2;
        if (vec[mid] < val) left = mid;
        else right = mid;
    }
    if (vec[left] == val) return left; // возвращаем индекс исходного эл.
    else return -1; // или -1, если такого нет.
}
int main () {
    vector<string> a , b; // а - коды зарегистрированных, б - участников
    /* но условия эти данные уже загружены в память, поэтому
     * этого действия в программе не будет */
    vector<string> ans; // сюда будем записывать those, кто не принёс
    for (int i = 0; i < a.size(); ++i) // передираем зарегистрированных
        if (bin_search (a[i] , b) == -1) // если он не принёс
            ans.push_back (a[i]); // записываем его код в ответ
    /* теперь в ans хранятся нужные коды, можно вывести их
     * на экран, например: */
    for (int i = 0; i < ans.size(); ++i)
        cout << ans[i] << endl;
    return 0;
}
```

Пример. Учсть зareгистрированысъ участники с кодами
"ABC1", "DEF2", "GHI3", "JKL4"; а принесли "ABC1" и "JKL4".
В функции main передираются коды зарегистрированных,
bin_search ("ABC1", {"ABC1", "JKL4"}) вернёт 0, bin_search ("DEF2", {"ABC1", "JKL4"})
вернёт -1, bin_search ("GHI3", {"ABC1", "JKL4"}) вернёт -1,
bin_search ("JKL4", {"ABC1", "JKL4"}) вернёт 1. Тогда получимся
ans = {"DEF2", "GHI3"}, это верный ответ.



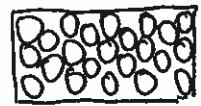
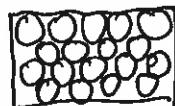
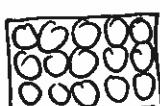
~~Задача 6.~~ ~~Анализ задачи~~ ~~ИЧУЮЩАЯ~~

Из условия однозначно понятно, что ставится в приоритет если в приоритет ставится размер шарика (то есть надо сделать шарика самого большого из возможных размеров), то ответ прост. Пусть a и b - размеры икуры. Тогда $W = \max(a, b)$, $h = \min(a, b)$. Тогда количество таких шариков равно $k = \text{int}(w/h)$ (целочисленное деление), их диаметр равен $d = h$.

Также в условии не сказано являемся ли размеры икуры и искомый диаметр шариков целыми числами. И если все - шарик в приоритет ставится количество шариков, то ответ - бесконечно много шариков бесконечно малого диаметра. В предположении, что числа все - шарик целые, задачу можно решить. Диаметр равен 1 и шарики должны быть расположены так:

Ответ снова очень прост. Количество таких шариков равно $a * b$, где a и b - размеры икуры. (Ведь пытаться разместить шарики диаметром 1 ~~бесконечно~~ получится только меньше). И при этом, если диаметр увеличить, то количество уменьшится.

Если же не диаметр, и количество не являются максимально возможными, то понятно, что считать ответом на задачу. Возможен иной способ решения ввиду максимально эффективного использования площади икуры. Т.е. есть найти такие k и d , что значение $k \cdot \frac{\pi d^2}{4}$ - максимально (k -количество, d -диаметр). Иначе говоря максимальный суммарный размер. Чтобы решить такую задачу будем перебирать значение d от 1 до $\min(a, b)$, где a и b - размеры икуры, и будем считать k . При максимально эффективном способе расположения залоговок:



← Этим отпадает, т.к. при такой ориентации прямугольника от пристыковки вправо.



В первом способе: $k = \text{int}(a/d) * \text{int}(b/d)$, ~~суммарная площадь~~:

Во втором способе: суммарная площадь максимальна, когда $k \cdot d^2$ максимальна:

$$\frac{\pi}{4} = k \cdot d^2.$$

Во втором способе: пусть $w = \max(a, b)$, $h = \min(a, b)$,
 и ширину d не высоте, т.к. ширина (возможно в лёгких
 случаях будем $m+1$ по ширине). Из геометрии:

$$d + (n-1) \cdot \frac{d\sqrt{3}}{2} \leq h; n \leq \frac{2(h-d)}{d\sqrt{3}} + 1, \text{ т.е. } n = \text{int}\left(\frac{2(h-d)}{d\sqrt{3}} + 1\right)$$

Также, ~~если~~ $d \cdot m + \frac{d}{2} \leq w$ или $d \cdot m \leq w$ (т.е. m или $m+1$
 ширина по ширине). $m \leq \frac{w}{d} - \frac{1}{2}$ или $m \leq \frac{w}{d}$, т.е. ~~ширина~~

т.е. $m = \min(\text{int}(w/d - 0.5), \text{int}(w/d))$. Тогда, если $m = \text{int}(w/d)$,
 то ~~площадь~~ $k = n \cdot m$, иначе $k = (n + n \% 2) / 2 * (m + 1) + (n - n \% 2) / 2 * m$.

Получаемся, что надо заполнить том бармаки $(k; d)$, где
 $k \cdot d^2$ максимальна. И это нужно проделать для всех d от 1 до h .

Код на C++, если применим на размер:

```
#include <iostream>
using namespace std;
int main(){
    int a, b;
    cin >> a >> b; // ввод размеров
    cout << int(max(a,b)/min(a,b)) << ' ' << min(a,b) << endl; // вывод k и d
    return 0;
}
```

Код на C++, если применим на количество:

```
#include <iostream>
using namespace std;
int main(){
    int a, b;
    cin >> a >> b; // ввод размеров
    cout << a * b << ' ' << 1 << endl; // вывод k и d
    return 0;
}
```



• Код на C++, если применить на суммарный размер:

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    int a, b;
    cin >> a >> b; // ввод размеров
    int w = max(a, b), h = min(a, b);, ans_k = w * h, ans_d = 1, s = w * h;
    /* ans_k, ans_d - для хранения будущего ответа, s - для перестёма ответа,
    for (int d=2; d <= h; ++d) { // ал. решение
        int n = 2 * (h-d)/d / sqrt(3)+1; // по высоте
        int m = min(w/d-0,s, w/d); // по ширине
        if (m == int(w/d)) k = n * m;
        else k = (n+n%2)/2 * (m+1)+(n-n%2)/2*m;
        if (k * d * d > s) {
            s = k * d * d;
            ans_k = k;
            ans_d = d;
        }
    }
    cout << ans_k << endl << ans_d << endl; // вывод k и d.
    return 0;
}
```

Например, если есть шкура 2^*3 , то $w=3$, $h=2$, $ans_k=6$, $ans_d=1$, $s=6$. В цикле, при $d=2$: $n=1$, $m=1$, $k=1$, что соответствует такой картины: $\boxed{2}$. $k * d * d = 4 < 6$, условие ложно. Выводится: 6, 1 — верно. Это соответствует картинке $\boxed{\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}}$.

Задача 3.

Пусть переменные a, b, c, d отвечают за красную, жёлтую, зелёную и дополнительную жёлтую секции. Т.о. условию возможно, что горят: $a; b; c; ab; d; acd; ad$ (7 вариантов). Построили таблицу истинности:

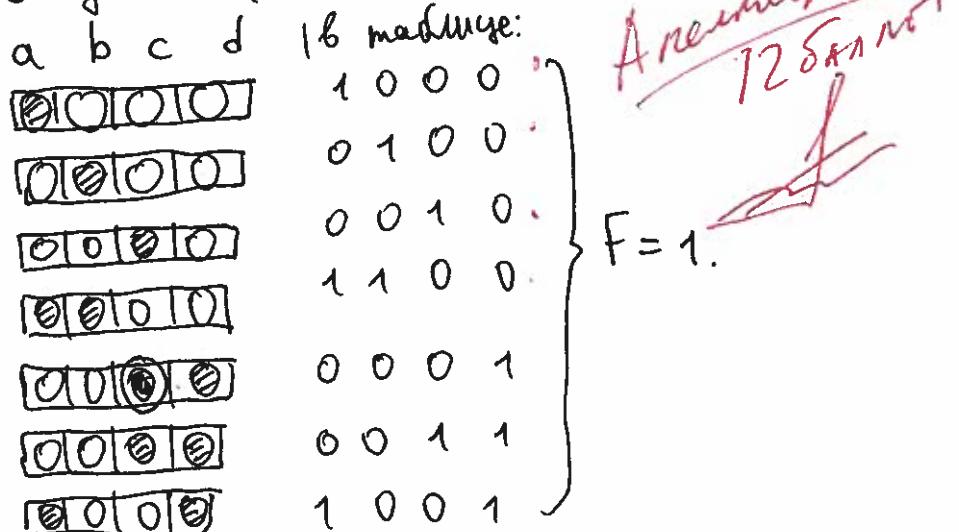


Олимпиада школьников «Гранит науки» Шифр работы ЧЧ17-57
 БЛАНК ОЛИМПИАДНОЙ РАБОТЫ (не заполнять)

a	b	c	d	F
0	0	0	0	0
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

F - светофор

Рисунки: (\odot - горит, 0 - не горит).



Когда можно составить выражение для F . Там будет 4 слагаемых, разделённых „или“, каждое из которых отвечает за одну из возможных ситуаций:

$$F(a, b, c, d) = a \cdot b \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot b \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot b \cdot c \cdot \bar{d} + a \cdot b \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d + a \cdot \bar{b} \cdot \bar{c} \cdot d. (\cdot = "и" , + = "или" , | = "не").$$

Например, слагаемое $a \cdot b \cdot \bar{c} \cdot \bar{d}$ получилось так: а - горит, б - горит, с - не горит, д - не горит.

~~Можно записать и по другому. $F(a, b, c, d)$ равно 1 тогда, когда одно значение из чисел a, b, c, d равно единице, а при этом не ($a=1, c=1$) и не ($b=1, d=1$) и не обе ($a=1, b=1$) или ($c=1, d=1$) или ($a=1, d=1$):~~

~~$F(a, b, c, d) =$~~

Запомни, что второе и четвёртое слагаемое можно объединить в $b \cdot \bar{c} \cdot \bar{d}$; первое и третье - в $a \cdot \bar{b} \cdot \bar{c}$, третью и шестую - в $\bar{a} \cdot \bar{b} \cdot c$. Итак: $F(a, b, c, d) = b \cdot \bar{c} \cdot \bar{d} + a \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d$.