

11 класс
1 вариант

1. У Бельчонка есть N целых чисел. Он хочет выбрать некоторые из них, чтобы получить максимально возможную чётную (то есть, делящуюся на 2) сумму. Обратите внимание, что если Бельчонок не выберет ни одного числа, то сумма будет равна чётному числу 0.

На вход подаётся число N ($1 \leq N \leq 100\,000$) — количество чисел. В следующей строке записаны N целых чисел, имеющих у Бельчонка. Все эти числа не меньше 1 и не больше . Выведите максимально возможную чётную сумму, которую можно получить, используя каждое из данных чисел не более одного раза.

Напишите программу на любом языке программирования или подробный алгоритм на русском языке

2. Двое по очереди ломают шоколадку 7×9 . За ход разрешается разломать любой кусок вдоль линии углубления либо съесть кусочек 1×1 , если такой кусочек был ранее отломан; проигрывает тот, кто не может сделать ход. Кто выиграет при оптимальной игре первый или второй игрок?

3. У Бельчонка есть N слонов на шахматной доске 100 на 100 . Строки и столбцы пронумерованы от 1 до 100, при этом строки пронумерованы сверху вниз, а столбцы пронумерованы слева направо.

Бельчонок считает, что два слона атакуют друг друга, если они стоят на одной диагонали. Обратите внимание, в отличие от настоящих шахмат, наличие третьего слона между ними не является помехой. Теперь Бельчонок хочет посчитать количество пар слонов, которые атакуют друг друга.

На вход подаётся число N ($1 \leq N \leq 2\,000$) — количество слонов на шахматной доске. Каждая из следующих N строк содержит два целых числа x_i и y_i ($1 \leq x_i, y_i \leq 100$) — номер строки и номер столбца, в которых расположен i -й слон. Гарантируется, что позиции всех слонов различны. Выведите количество пар слонов, которые атакуют друг друга.

Напишите программу на любом языке программирования и подробный алгоритм на русском языке.

4. Бельчонок нашёл различных натуральных чисел и при тщательном изучении этих чисел заметил, что из любых трёх чисел можно выбрать два

так, что сумма этих двух чисел – нечётное число. При каком наибольшем это возможно?

5. У Бельчонка украли его любимый орех. Для поимки вора Бельчонок позвал всех своих друзей. Никто не знает, где находится вор, но Бельчонок знает, как вор движется.


Лес можно представить как сетку размера 100×2 , где каждая ячейка представляет одно дерево. У каждого дерева есть свое уникальное название “(X, Y)”, где X и Y— это координаты на сетке. Вор движется следующим образом:

Каждую минуту вор покидает дерево (X, Y), на котором он на данный момент прячется, и переходит на одно из деревьев:

$(X - 1, Y), (X + 1, Y), (X - 1, Y - 1), (X - 1, Y + 1), (X + 1, Y - 1), (X + 1, Y + 1)$

при условии, что это дерево существует в Лесу.

Ниже приведен пример возможных движений вора, если он расположен в районе (7,1):

(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)	(7, 2)	(8, 2)	(9, 2)	...
(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)		(8, 1)	(9, 1)	...

У Бельчонка достаточно друзей, так что **каждую минуту** он может выбрать любые **два** дерева в Лесу и полностью обыскать их, удостоверившись, что если вор расположен на одном из них, то он будет пойман. Бельчонок очень голоден, поэтому необходимо поймать вора **не более** чем за 240 минут. У этой задачи нет ввода. В первой строке выведите целое число N — продолжительность расследования в минутах. Каждая из следующих N строк должна содержать ровно по 4 целых числа, , , , разделенных пробелами, обозначающих 2 дерева (,), (,), которые должны быть обысканы в i -ую минуту. Должно быть верно, что вор будет гарантированно пойман не более чем за 240 минут, **независимо от начального положения и перемещений вора.**

Напишите программу на любом языке программирования и подробный алгоритм на русском языке.

Решения

1. Просуммируем все, числа, если сумма чётна, то её и выводим, если сумма нечётна, то необходимо вычесть наименьшее нечётное число.

Пример программы(на C++).

```
#include <iostream>
using namespace std;

int main(){
    long long ans = 0, ne = 0;
    int n, d, mn = 1000000000;
    cin >> n;
    for(int i=0; i<n; ++i){
        cin >> d;
        ans += d;
        if(d%2){
            mn = min(mn, d);
            ++ne;
        }
    }
    if(ne%2){
        ans -= mn;
    }
    cout << ans;
}
```

Критерии

Описан алгоритм или написана программа, верно решающие задачу – 10 баллов.

Не учтена возможность переполнения типа int – снять 3 балла.

Мелкие недочёты в программе или алгоритме – снять 2 балла.

2. Существует два типа ходов: съесть кусочек шоколадки и разломать кусок шоколадки. Игра окончится, когда вся шоколадка будет съедена, значит, ходов первого типа будет $7 \times 9 = 63$.

А шоколадка будет съедена, когда все возможные разломы будут совершены. За каждый ход второго типа количество кусочков шоколадки увеличивается на один. Сначала был 1 кусочек, а в конце – 63, значит, ходов второго типа будет 62.

Результат игры не зависит от игроков. В любой последовательности будут совершены 63 хода первого типа и 62 хода второго типа. Всего будет совершено 125 ходов, значит, последний ход совершит первый игрок, он и победит.

Критерии

Верный ответ без объяснений – 1 балл.

Верное решение и ответ – 10 баллов.

3. Рассмотрим все диагонали: диагонали проходящие в ↘ направлении определяются значением $x+y$, т.е все слоны с одинаковым значением $x+y$ лежат на одной диагонали. Диагонали в ↗ направлении определяются значением $x-y$, т.е все слоны с одинаковым значением $x-y$ также лежат на одной диагонали. Зная количество слонов на каждой диагонали, можно найти количество пар слонов, которые атакуют друг друга. Если на диагонали k слонов, то количество пар будет равно $\frac{k(k-1)}{2}$.

Пример программы(на C++).

```
#include <iostream>
using namespace std;
```

```
int main(){
    int n, d, x, y, v[300], r[300]; //массив v для первого вида диагоналей, r - для
    //второго типа
    long long ans =0;
    cin >> n;
    for(int i=0; i<n; ++i){
        cin >> x >> y;
        ++v[99+x-y];
        ++r[x+y];
    }
    for(int i=0; i<300; ++i){
        ans += v[i]*(v[i]-1)/2; //подсчёт количества пар на каждой диагонали
        ans += r[i]*(r[i]-1)/2;
    }
    cout << ans;
    return 0;
}
```

Критерии

Описан алгоритм или написана программа, верно решающие задачу – 10 баллов.

Не учтена возможность переполнения типа int – снять 3 балла.

Описан алгоритм, но не написана программа – не более 5 баллов.

Неверный расчёт пар для каждой диагонали – не более 2 баллов.

4. Если среди чисел есть три нечётных, то из этих трёх нельзя никак выбрать два, сумма которых нечётна. Аналогичная ситуация, если есть три чётных числа.

Покажем, что для чисел, среди которых два чётных и два нечётных, условие задачи выполняется. Выбрав три числа, существует две ситуации: либо два чётных и одно нечётное, тогда сумма чётного и нечётного нечётна, либо одно чётное и два нечётных, тогда сумма чётного и нечётного нечётна.

Критерии

Доказано, что не больше 4 – 6 балла.

Приведён пример для – 4 балла.

Оба пункта выполнены – 10 баллов.

5. Легко заметить, что чётность координаты меняется каждый раз, когда вор движется. Предположим, что в самом начале координата позиции вора нечётная, и проверяем деревья (1,1) и (1,2). В следующую минуту проверяем деревья (2,1) и (2,2) и так далее до деревьев (100,1) и (100,2). Таким образом, если начальная чётность была такой, как предполагалось, вор никогда бы не перепрыгнул на дерево с координатой на день , следовательно, он не мог бы перепрыгнуть через поисковый отряд и был бы пойман. Если он не был пойман, то начальная координата позиции вора чётная. Тогда пропуская одну минуту, в 102 минуту снова проверяем деревья (1,1) и (1,2) и так далее до деревьев (100,1) и (100,2). Пропустив одну минуту, мы позволили вору сменить чётность координаты и при втором проходе вор обязательно будет пойман. Важно, что при пропуске одной минуты, надо вывести координаты любой точки

Пример программы(на C++).

```
#include <iostream>
using namespace std;
```

```
int main(){
    cout<<"201\n";
    for(int i=1; i<=100; ++i)
        cout<<i<<" 1 "<<i<<" 2 "<<endl;
    cout<<" 1 1 1 2 "<<endl;
    for(int i=1; i<=100; ++i)
        cout<<i<<" 1 "<<i<<" 2 "<<endl;
    return 0;
}
```

Критерии

Описан алгоритм и написана программа, верно решающие задачу – 10 баллов.

Описан алгоритм, но не написана программа – не более 5 баллов.

Программа или алгоритм не учитывают ограничение на время поиска вора –
снять 3 балла.

11 класс
Вариант 2

1. У Бельчонка есть N целых чисел. Он хочет выбрать некоторые из них, чтобы получить максимально возможную нечётную сумму. Гарантируется, что среди данных чисел найдётся хотя бы одно нечётное.

На вход подаётся число N ($1 \leq N \leq 100\,000$) — количество чисел. В следующей строке записаны N целых чисел, имеющихся у Бельчонка. Все эти числа не меньше 1 и не больше . Выведите максимально возможную чётную сумму, которую можно получить, используя каждое из данных чисел не более одного раза.

Напишите программу на любом языке программирования или подробный алгоритм на русском языке

2. Двое по очереди ломают шоколадки 6×8 и 7×9 . За ход разрешается разломать любой кусок вдоль линии углубления либо съесть кусочек 1×1 , если такой кусочек был ранее отломан; проигрывает тот, кто не может сделать ход. Кто выиграет при оптимальной игре первый или второй игрок?

3. У Бельчонка есть N ферзей на шахматной доске 100 на 100 . Строки и столбцы пронумерованы от 1 до 100, при этом строки пронумерованы сверху вниз, а столбцы пронумерованы слева направо.

Бельчонок считает, что два ферзя атакуют друг друга, если они стоят на одной диагонали, на одной горизонтали или на одной вертикали. Обратите внимание, в отличие от настоящих шахмат, наличие третьего ферзя между ними не является помехой. Теперь Бельчонок хочет посчитать количество пар ферзей, которые атакуют друг друга.

На вход подаётся число N ($1 \leq N \leq 2\,000$) — количество ферзей на шахматной доске. Каждая из следующих N строк содержит два целых числа x_i и y_i ($1 \leq x_i, y_i \leq 100$) — номер строки и номер столбца, в которых расположен i -й ферзь. Гарантируется, что позиции всех ферзей различны. Выведите количество пар ферзей, которые атакуют друг друга.

Напишите программу на любом языке программирования и подробный алгоритм на русском языке.

4. Бельчонок нашёл различных натуральных чисел и при тщательном изучении этих чисел заметил, что из любых четырёх чисел можно выбрать

два так, что сумма этих двух чисел – нечётное число. При каком наибольшем это возможно?

5. У Бельчонка украли его любимый орех. Для поимки вора Бельчонок позвал всех своих друзей. Никто не знает, где находится вор, но Бельчонок знает, как вор движется.


Лес можно представить как сетку размера 100×3 , где каждая ячейка представляет одно дерево. У каждого дерева есть свое уникальное название “(X, Y)”, где X и Y— это координаты на сетке. Вор движется следующим образом:

Каждую минуту вор покидает дерево (X, Y), на котором он на данный момент прячется, и переходит на одно из деревьев:

$(X - 1, Y), (X + 1, Y), (X - 1, Y - 1), (X - 1, Y + 1), (X + 1, Y - 1), (X + 1, Y + 1)$

при условии, что это дерево существует в Лесу.

Ниже приведен пример возможных движений вора, если он расположен в районе (7,1):

(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)	(7, 2)	(8, 2)	(9, 2)	...
(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)		(8, 1)	(9, 1)	...
(1, 0)	(2, 0)	(3, 0)	(4, 0)	(5, 0)	(6, 0)	(7, 0)	(8, 0)	(9, 0)	...

У Бельчонка достаточно друзей, так что **каждую минуту** он может выбрать любые **три** дерева в Лесу и полностью обыскать их, удостоверившись, что если вор расположен на одном из них, то он будет пойман. Бельчонок очень голоден, поэтому необходимо поймать вора **не более** чем за 240 минут.

У этой задачи нет ввода. В первой строке выведите целое число N — продолжительность расследования в минутах. Каждая из следующих N строк должна содержать ровно по 4 целых

числа, , , , , разделенных пробелами, обозначающих 3 дерева (,), (,), (,), которые должны быть обысканы в i -ую минуту. Должно быть верно, что вор будет гарантированно пойман не более чем за 240 минут, **независимо от начального положения и перемещений вора.**

Напишите программу на любом языке программирования и подробный алгоритм на русском языке.

Решения

1. Просуммируем все, числа, если сумма нечётна, то её и выводим, если сумма чётна, то необходимо вычесть наименьшее нечётное число.

Пример программы(на C++).

```
#include <iostream>
using namespace std;

int main(){
    long long ans = 0, ne = 0;
    int n, d, mn = 1000000000;
    cin >> n;
    for(int i=0; i<n; ++i){
        cin >> d;
        ans += d;
        if(d%2){
            mn = min(mn, d);
            ++ne;
        }
    }
    if(ne%2 == 0){
        ans -= mn;
    }
    cout << ans;
}
```

Критерии

Описан алгоритм или написана программа, верно решающие задачу – 10 баллов.

Не учтена возможность переполнения типа int – снять 3 балла.

Мелкие недочёты в программе или алгоритме – снять 2 балла.

2. Существует два типа ходов: съесть кусочек шоколадки и разломать кусок шоколадки. Игра окончится, когда вся шоколадка будет съедена, значит, ходов первого типа будет $6 \times 8 + 7 \times 9 = 111$.

А шоколадка будет съедена, когда все возможные разломы будут совершены. За каждый ход второго типа количество кусочков шоколадки увеличивается на один. Сначала было 2 кусочка, а в конце – 111, значит, ходов второго типа будет 109.

Результат игры не зависит от игроков. В любой последовательности будут совершены 111 ходов первого типа и 109 ходов второго типа. Всего будет совершено 220 ходов, значит, последний ход совершит второй игрок, он и победит.

Критерии

Верный ответ без объяснений – 1 балл.

Верное решение и ответ – 10 баллов.

3. Рассмотрим все диагонали: диагонали проходящие в ↘ направлении определяются значением $x - y$, т.е все ферзи с одинаковым значением $x - y$ лежат на одной диагонали. Диагонали в ↗ направлении определяются значением $x + y$, т.е все ферзи с одинаковым значением $x + y$ также лежат на одной диагонали. Зная количество ферзей на каждой диагонали, вертикали и горизонтали, можно найти количество пар слонов, которые атакуют друг друга. Если на линии k ферзей, то количество пар будет равно $\frac{k * (k - 1)}{2}$

Пример программы(на C++).

```
#include <iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    int n, d, x, y, v[300], r[300], q[300], w[300]; //массив v для первого вида  
    диагоналей, r - для второго типа, q – для вертикалей, w – для горизонталей
```

```
    long long ans = 0;
```

```
    cin >> n;
```

```
    for(int i=0; i<n; ++i){
```

```
        cin >> x >> y;
```

```
        ++v[99+x-y];
```

```
        ++r[x+y];
```

```
        ++w[x];
```

```
        ++q[y];
```

```
    }
```

```
    for(int i=0; i<300; ++i){
```

```
        ans += v[i]*(v[i]-1)/2; //подсчёт количества пар на каждой диагонали
```

```
        ans += r[i]*(r[i]-1)/2;
```

```
        ans += w[i]*(w[i]-1)/2;
```

```
        ans += q[i]*(q[i]-1)/2;
```

```
    }
```

```
    cout << ans;
```

```
    return 0;
```

```
}
```

Критерии

Описан алгоритм или написана программа, верно решающие задачу – 10 баллов.

Не учтена возможность переполнения типа int – снять 3 балла.

Описан алгоритм, но не написана программа – не более 5 баллов.

Неверный расчёт пар для каждой диагонали – не более 2 баллов.

4. Если среди чисел есть 4 нечётных, то из этих четырёх нельзя никак выбрать два, сумма которых нечётна. Аналогичная ситуация, если есть 4 чётных числа.

Покажем, что для чисел, среди которых три чётных и три нечётных, условие задачи выполняется. Выбрав 4 числа, существует три ситуации: либо три чётных и одно нечётное, либо одно чётное и три нечётных, либо два чётных и два нечётных, во всех случаях можно выбрать чётное и нечётное, тогда сумма чётного и нечётного нечётна.

Критерии

Доказано, что не больше 6 – 6 балла.

Приведён пример для – 4 балла.

Оба пункта выполнены – 10 баллов.

5. Легко заметить, что чётность координаты меняется каждый раз, когда он движется. Предположим, что в самом начале координата позиции вора нечётная, и проверяем деревья (1,0), (1,1) и (1,2). В следующую минуту проверяем деревья (2,0), (2,1) и (2,2) и так далее до деревьев (100,0), (100,1) и (100,2). Таким образом, если начальная чётность была такой, как предполагалось, вор никогда бы не перепрыгнул на дерево с координатой на день, следовательно, он не мог бы перепрыгнуть через поисковый отряд и был бы пойман. Если он не был пойман, то начальная координата позиции вора чётная. Тогда пропуская одну минуту, в 102 минуту снова проверяем деревья (1,0), (1,1) и (1,2) и так далее до деревьев (100,0), (100,1) и (100,2). Пропустив одну минуту, мы позволили вору сменить чётность координаты и при втором проходе вор обязательно будет пойман. Важно, что при пропуске одной минуты, надо вывести координаты любой точки

Пример программы(на C++).

```
#include <iostream>
using namespace std;
```

```
int main(){
    cout<<"201\n";
    for(int i=1; i<=100; ++i)
        cout<<i<<" 0 "<<i<<" 1 "<<i<<" 2 "<<endl;
    cout<<"1 0 1 1 1 2 "<<endl;
    for(int i=1; i<=100; ++i)
        cout<<i<<" 0 "<<i<<" 1 "<<i<<" 2 "<<endl;
    return 0;
}
```