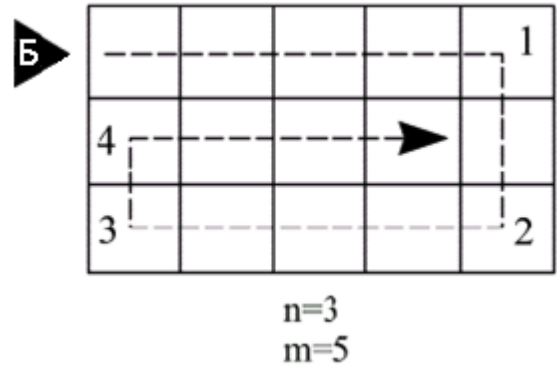


**10 класс**  
**Вариант 1**

1. Бельчонок хочет пройтись по своей комнате. План комнаты представлен в виде прямоугольника с целыми длинами сторон ( $n$  – высота прямоугольника,  $m$  – длина прямоугольника). Бельчонок встаёт в левую верхнюю клетку прямоугольника и начинает движение в направлении «слева - направо», после чего, двигаясь по спирали по часовой стрелке пока не пройдёт все клетки (см. рис.) При этом спираль постепенно «закручивается» вовнутрь, захватывая все клетки прямоугольника.



Напишите программу на любом языке программирования или подробный алгоритм на русском языке, определяющие для заданных исходных данных количество поворотов, которые должен выполнить Бельчонок в процессе обхода.

2. Двое играют в следующую игру. Имеются три кучки орехов: в первой – 9, во второй – 14, а в третьей – 19. За ход разрешается разбить кучку на две меньшие либо съесть один орех, лежащий отдельно, т.е. в кучке из одного ореха; проигрывает тот, кто не может сделать ход. Кто выиграет при оптимальной игре первый или второй игрок?

3. Каждое воскресенье Бельчонок вместе с друзьями играет в футбол. Сперва случайным образом выбираются два капитана, после чего капитаны по очереди выбирают по одному игроку.

Сила команды вычисляется как суммарная сила всех игроков в этой команде, включая капитана.

Оба капитана, конечно же, стремятся максимизировать силу своей команды.

На вход подаётся число  $N$  — количество человек, пришедших на игру,  $N$  всегда чётно. Во второй строке записаны целые числа — сила  $i$ -го игрока.

Выведите единственное целое число — минимально возможную разницу в силе команд.

Напишите программу на любом языке программирования и подробный алгоритм на русском языке (если описан только алгоритм, то задача оценивается в половину баллов).

4. В турнире по настольному теннису участвовали 16 спортсменов. Игры проводились по очереди. Проиграв хотя бы две игры, участник выбывает. По правилам два спортсмена могут встретиться не более одного раза. Турнир заканчивается, когда невозможно больше провести ни одной игры. Какое наибольшее количество участников могло остаться в турнире к моменту его завершения?

5. Бельчонок хочет выбрать на плоскости несколько точек так, чтобы никакие три из них не лежали на одной прямой.

На вход подаётся число  $N$  – число точек, причём  $N \leq 10000$ . Если искомое множество точек можно построить, то выведите  $N$  строк, каждая из которых должна содержать два числа – координаты соответствующей точки. Среди точек не должно быть совпадающих. Все координаты должны быть целыми числами, не превосходящими 10000 по абсолютному значению. Если искомое множество точек нельзя построить, выведите -1.

Напишите программу на любом языке программирования и подробный алгоритм на русском языке (если описан только алгоритм, то задача оценивается в половину баллов).

## Решения

1. Конечно, можно было написать сложную программу обхода прямоугольника по спирали. Но есть и более оптимальное решение.

После каждого поворота Бельчонка уменьшается высота или длина прямоугольника по очереди. Если высота меньше или равна длине, т.е.  $n \leq m$ , то по достижении высоты 1, можно будет совершить ещё только один поворот, уменьшив длину, значит совершенно  $n$  уменьшение высоты и  $n$  уменьшение длины. Получилось  $2n$  поворотов.

Если же высота больше длины, т.е.  $n > m$ , то будет  $m$  уменьшений длины и  $m$  уменьшений высоты, т.к. уменьшение высоты всегда самое первое. В этом случае получилось  $2m$  поворотов.

*Пример программы(на C++).*

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

```
int main(){
    int n,m;
    cin >> n >> m;
    if(n <= m )
        cout << 2*(n-1);
    else
        cout << 2*m-1;
    return 0;
}
```

### Критерии

Описан алгоритм или написана программа, верно решающие задачу для любых  $n, m$  – 10 баллов.

Рассмотрен только один из случаев – 5 баллов.

Мелкие недочёты в программе или алгоритме – снять 2 балла.

2. Существует два типа ходов: съесть орех и разбить кучку. Игра окончится, когда все орехи будут съедены, значит, ходов первого типа будет  $9+14+19=42$ .

А орехи будут съедены, когда все кучки будут разбиты на самые маленькие по одному ореху в кучке. За каждый ход второго типа количество кучек увеличивается на один. Сначала было 3 кучки, а в конце – 42, значит, ходов второго типа будет 39.

Результат игры не зависит от игроков. В любой последовательности будут совершены 42 хода первого типа и 39 ходов второго типа. Всего будет совершено 81 ход, значит, последний ход совершит первый игрок, он и победит.

## Критерии

Верный ответ без объяснений – 1 балл.

Верное решение и ответ – 10 баллов.

3. Отсортировав по убыванию, будем по очереди рассматривать пары игроков. На этапе рассмотрения пары, если первая команда сильнее второй, то из этой пары сильного игрока добавляем во вторую команду, и наоборот, если первая команда слабее, сильного игрока добавляем в первую команду. Таким образом, получится минимальная разница, т.к. на каждом этапе эта разница минимизировалась.

*Пример программы(на C++).*

```
#include <iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    int n, a[10000], s1=0,s2=0;//s1 - сила первой команды, s2 - сила второй  
команды
```

```
    cin >> n;
```

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

```
        cin >> a[i];
```

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

```
        for(int j=0; j<n-i-1; ++j){
```

```
            if (a[j] < a[j+1]) {
```

```
                int b = a[j];
```

```
                a[j] = a[j+1];
```

```
                a[j+1] = b;
```

```
            }
```

```
        }
```

```
    }
```

```
    for(int i=0; i<n; i+=2)
```

```
        if(s1 < s2){ //если вторая команда сильнее
```

```
            s1 += max(a[i], a[i+1]);//первой команде добавляем более сильного  
игрока
```

```
            s2 += min(a[i], a[i+1]);
```

```
        }
```

```
    else{ //если вторая команда сильнее
```

```
        s1 += min(a[i], a[i+1]);
```

```
        s2 += max(a[i], a[i+1]);//второй команде добавляем более сильного
```

```
    }
```

```
    cout << abs(s1-s2);
```

```
    return 0;
```

```
}
```

## Критерии

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

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

4. В момент завершения турнира каждые двое из оставшихся участников должны были встретиться друг с другом, иначе можно было бы провести ещё одну игру – между этой парой участников. Пусть осталось не менее четырёх участников. Выберем любых четырёх из них. Каждый должен был проиграть между ними не более одного раза, значит, общее количество поражений, а значит, и игр между этими спортсменами, не больше 4. В то же время общее количество игр равно 6. Получаем противоречие, значит, может остаться менее 4 участников.

Трое спортсменов могло остаться, если вначале 13 спортсменов встречались только с двумя спортсменами А и В и им постоянно проигрывали. А потом оставшиеся спортсмены А, В и С по кругу одержали победы в играх между собой. А выиграл у В, В – у С, а С – у А.

## Критерии

Доказано, что могло остаться не больше 3 участников – 6 балла.

Приведён пример турнира, в котором осталось 3 участника – 4 балла.

Оба пункта выполнены – 10 баллов.

5. Как известно никакие три точки выбранные на параболе, не лежат на одной прямой. А т.к.  $y = x^2 - 200x + 10000$ , то можно выбрать для абциссы точки любую координату от -100 до 100, тогда ордината будет находиться в пределах от 0 до 10000, что соответствует условию задачи.

*Пример программы(на C++).*

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

```
int main(){
    int n;
    cin >> n;
    for(int i=0; i<n; ++i)
        cout << -100 + i << " " << (-100 + i)*(-100 + i);
    return 0;
}
```

## Критерии

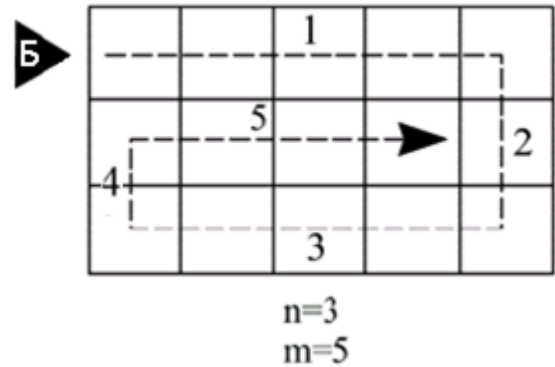
Описан алгоритм и написана программа, верно решающие задачу – 10 баллов.

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

Программа или алгоритм не учитывают ограничение на значение координат – снять 3 балла.

**10 класс**  
**Вариант 2**

1. Бельчонок хочет пройтись по своей комнате. План комнаты представлен в виде прямоугольника с целыми длинами сторон ( $n$  – высота прямоугольника,  $m$  – длина прямоугольника). Бельчонок встанет в левую верхнюю клетку прямоугольника и начинает движение в направлении «слева - направо», после чего, двигаясь по спирали по часовой стрелке пока не пройдёт все клетки (см. рис.) При этом спираль постепенно «закручивается» вовнутрь, захватывая все клетки прямоугольника.



Напишите программу на любом языке программирования или подробный алгоритм на русском языке, определяющие для заданных исходных данных количество прямых участков, которые проходит Бельчонок.

2. Двое играют в следующую игру. Имеются четыре кучки орехов: в первой – 5, во второй – 10, в третьей – 15, а в четвёртой – 20. За ход разрешается разбить кучку на две меньшие либо съесть один орех, лежащий отдельно, т.е. в кучке из одного ореха; проигрывает тот, кто не может сделать ход. Кто выиграет при оптимальной игре первый или второй игрок?

3. Каждое воскресенье Бельчонок вместе с друзьями играет в футбол. Выбраны два капитана, сейчас они должны по очереди выбирать по одному игроку в свою команду.

Сила команды вычисляется как суммарная сила всех игроков в этой команде, включая капитана.

Капитан первой команды, конечно же, стремится максимизировать силу своей команды. А вот капитана второй команды кажется кто-то подкупил и он стремится минимизировать силу своей команды.

На вход подаётся число  $N$  — количество человек, пришедших на игру,  $N$  всегда чётно. Во второй строке записаны целые числа — сила  $i$ -го игрока.

Выведите единственное целое число — максимально возможную разницу в силе команд.

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

4. В турнире по настольному теннису участвовали 16 спортсменов. Игры проводились по очереди. Проиграв хотя бы две игры, участник выбывает. Однако среди спортсменов, есть сын организатора соревнований и у него иммунитет от выбывания, т.е. независимо от результата игр он никогда не выбывает. По правилам два спортсмена могут встретиться не более одного раза. Турнир заканчивается, когда невозможно больше провести ни одной игры. Какое наибольшее количество участников могло остаться в турнире к моменту его завершения?

5. *Перестановкой  $p$*  называется упорядоченный набор чисел  $p_1, p_2, \dots, p_n$  состоящий из  $n$  различных целых положительных чисел, каждое из которых не больше чем  $n$ . Число  $n$  будем называть длиной перестановки

Бельчонок хочет найти такую перестановку  $p$  длины  $n$ , что среди чисел  $p_1, p_2, \dots, p_n$  ровно  $k$  различных.

На вход подаются два целых положительных числа  $n, k$  ( $1 \leq k < n \leq 10^5$ ). Выведите  $n$  целых чисел — искомую перестановку. Если существует несколько ответов, разрешается вывести любой.

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

## Решения

1. Конечно, можно было написать сложную программу обхода прямоугольника по спирали. Но есть и более оптимальное решение.

После каждого поворота Бельчонка уменьшается высота или длина прямоугольника по очереди. Если высота меньше или равна длине, т.е.  $n \leq m$ , то по достижении высоты 1, можно будет совершить ещё только один поворот, уменьшив длину, значит совершенно  $n$  уменьшение высоты и  $n$  уменьшение длины. Получилось  $2n$  поворотов.

Если же высота больше длины, т.е.  $n > m$ , то будет  $m$  уменьшений длины и  $m$  уменьшений высоты, т.к. уменьшение высоты всегда самое первое. В этом случае получилось  $2m$  поворотов.

А количество прямых участков на один больше чем поворотов.

Поэтому для  $n \leq m$  ответ  $2n + 1$ .

А для  $n > m$  ответ  $2m + 1$ .

*Пример программы(на C++).*

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

```
int main(){
    int n,m;
    cin >> n >> m;
    if(n <= m )
        cout << 2*n-1;
    else
        cout << 2*m;
    return 0;
}
```

### Критерии

Описан алгоритм или написана программа, верно решающие задачу для любых  $n, m$  – 10 баллов.

Рассмотрен только один из случаев – 5 баллов.

Мелкие недочёты в программе или алгоритме – снять 2 балла.

2. Существует два типа ходов: съесть орех и разбить кучку. Игра окончится, когда все орехи будут съедены, значит, ходов первого типа будет  $5+10+15+20=50$ .

А орехи будут съедены, когда все кучки будут разбиты на самые маленькие по одному ореху в кучке. За каждый ход второго типа количество кучек увеличивается на один. Сначала было 4 кучки, а в конце – 50, значит, ходов второго типа будет 46.

Результат игры не зависит от игроков. В любой последовательности будут совершены 50 ходов первого типа и 46 ходов второго типа. Всего будет



совершено 96 ходов, значит, последний ход совершит второй игрок, он и победит.

### Критерии

Верный ответ без объяснений – 1 балл.

Верное решение и ответ – 10 баллов.

3. Отсортировав по возрастанию, во вторую команду добавим первых игроков, а оставшихся – в первую команду. Таким образом, получится максимальная разница.

*Пример программы(на C++).*

```
#include <iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    int n, a[10000], s1=0,s2=0;//s1 - сила первой команды, s2 - сила второй  
    команды
```

```
    cin >> n;
```

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

```
        cin >> a[i];
```

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

```
        for(int j=0; j<n-i-1; ++j){
```

```
            if (a[j] > a[j+1]) {
```

```
                int b = a[j];
```

```
                a[j] = a[j+1];
```

```
                a[j+1] = b;
```

```
            }
```

```
        }
```

```
    }
```

```
    for(int i=0; i<n/2; ++i)
```

```
        s2 += a[i];
```

```
    for(int i=n/2; i<n; ++i)
```

```
        s1 += a[i];
```

```
    cout << abs(s1-s2);
```

```
    return 0;
```

```
}
```

### Критерии

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

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

4. Т.к. сын организатора всегда остаётся в конце турнира, предположим, что он проиграл во всех играх.

Рассмотрим остальных 15 игроков.

В момент завершения турнира каждые двое из оставшихся участников должны были встретиться друг с другом, иначе можно было бы провести ещё одну игру – между этой парой участников. Пусть осталось не менее четырёх участников. Выберем любых четырёх из них. Каждый должен был проиграть между ними не более одного раза, значит, общее количество поражений, а значит, и игр между этими спортсменами, не больше 4. В то же время общее количество игр равно 6. Получаем противоречие, значит, может остаться менее 4 участников.

Трое спортсменов могло остаться, если вначале 12 спортсменов встречались только с двумя спортсменами А и В и им постоянно проигрывали. А потом оставшиеся спортсмены А, В и С по кругу одержали победы в играх между собой. А выиграл у В, В – у С, а С – у А.

Таким образом, в конце турнира осталось 4 спортсмена.

### Критерии

Доказано, что могло остаться не больше 4 участников – 6 балла.

Приведён пример турнира, в котором осталось 4 участника – 4 балла.

Оба пункта выполнены – 10 баллов.

5. Если \_\_\_\_\_ ответом будет, например, вывод такой последовательности:

$n, 1, n-1, 2, n-2, 3, n-3, \dots,$

в которой на нечётных местах стоит убывающая последовательность, а на чётных возрастающая. Таким образом, получатся все возможные различные разницы от 1 до \_\_\_\_\_.

Если же \_\_\_\_\_, то можно вывести первые \_\_\_\_\_ членов такой последовательности, получив \_\_\_\_\_ различных разниц. А оставшиеся числа вывести так, чтобы соседние разности были равны единице.

*Пример программы(на C++).*

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

```
int main(){
    int n,k,t=0,l=1;
    cin>>n>>k;
    for(int i=0;i<k;++i){//вывод k первых членов последовательности
        t+=(n-i)*l;
        l*=-1;
        cout<<t<<" ";
    }
}
```

```
if(k%2==1)// в зависимости от чётности k оставшиеся члены нужно будет  
ВЫВЕСТИ  
for(int i=n-k/2-1;i>=k/2+1;--i)//в порядке убывания  
cout<<i<<" ";  
else  
for(int i=k/2+1;i<=n-k/2;++i)// или возрастания  
cout<<i<<" ";  
  
return 0;  
}
```

### **Критерии**

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

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