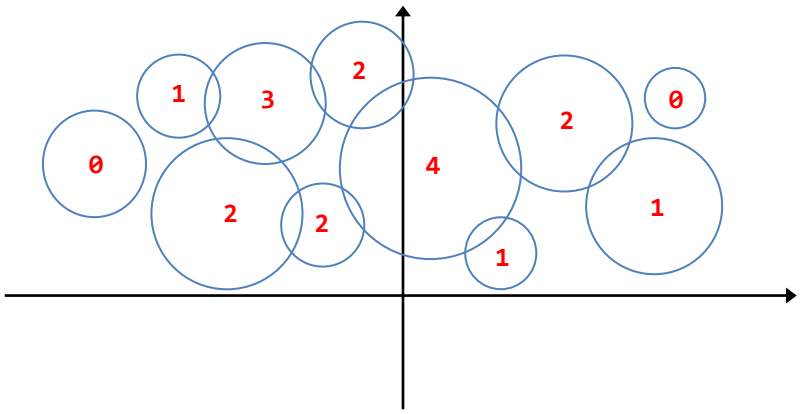
Задание №6. Создайте и инициализируйте динамический массив объектов `clCircle`. Найдите круги, которые пересекаются с другими кругами из данного массива. Выведите список таких кругов, отсортировав его по количеству пересечений.




```
=== Задание 6 =================
Количество пересекающихся кругов = 26
Упорядоченный список пересекающихся кругов:
    X         Y        R
  -6.95 |   -4.34 |  8.73 |  количество пересечений =  6
 -15.15 |    5.91 |  5.96 |  количество пересечений =  5
 -15.80 |    4.10 |  5.32 |  количество пересечений =  5
 -17.97 |    2.56 |  4.77 |  количество пересечений =  4
  -4.91 | -13.65 |  8.24 |  количество пересечений =  4
   1.05 | -25.60 |  6.24 |  количество пересечений =  4
  12.87 |  34.44 |  8.86 |  количество пересечений =  3
 -10.53 |   6.35 |  2.85 |  количество пересечений =  3
 -21.57 |   0.17 |  5.08 |  количество пересечений =  3
  -5.10 | -20.91 |  5.43 |  количество пересечений =  3
  -4.55 | -38.47 |  8.16 |  количество пересечений =  3
  -2.70 | -13.48 |  6.12 |  количество пересечений =  3
   6.38 | -34.08 |  5.81 |  количество пересечений =  3
  21.33 |  30.93 |  2.54 |  количество пересечений =  2
  20.51 |  34.77 |  2.50 |  количество пересечений =  2
  11.17 |  34.03 |  4.00 |  количество пересечений =  1
-778.29 | 201.47 |  4.65 |  количество пересечений =  1
-767.16 | 147.55 |  4.99 |  количество пересечений =  1
-774.96 | 206.76 |  4.00 |  количество пересечений =  1
-770.22 | 140.68 |  2.90 |  количество пересечений =  1
 -20.54 | -38.14 |  7.84 |  количество пересечений =  1
-806.92 |-756.26 |  3.44 |  количество пересечений =  1
-804.29 |-758.36 |  3.88 |  количество пересечений =  1
  11.87 | -37.07 |  3.16 |  количество пересечений =  1
  37.47 | -36.43 |  3.22 |  количество пересечений =  1
  36.46 | -33.82 |  7.17 |  количество пересечений =  1
```

Для выполнения задания добавьте в класс `clCircle` (см. задание №4) следующие методы.

1) Перегрузка метода `set()` вводит значения полей `x`, `y`, `radius` из строки `s`, используя в качестве разделителей символы строки `sep`:

```
void set(const char* s, const char* sep) {
   char* p=(char*) s; p+=myspn(p, sep); //p -> на первое слово
   x = ato(p);                          //чтение первого поля
   p+=myspn(p, sep); p+=myspn(p, sep);  //p -> на второе слово
   y = ato(p);                          //чтение второго поля
   p+=myspn(p, sep); p+=myspn(p, sep);  //p -> на третье слово
   radius = ato(p);                     //чтение третьего поля
}
```

2) Метод `get5(int sw, int sp)` выводит значения полей и поясняющий текст (см. рисунок). Параметры `sw`, `sp` – значения манипуляторов.
3) Метод `intersection()` возвращает количество кругов из массива `crc` (`ncrc` – размер массива), которые пересекаются с кругом данного объекта (`x`, `y`, `radius`):

```
int intersection(clCircle *crc, int ncrc) {
   int n=0;
   for(int i=0; i<ncrc; i++)
      if(crc[i].dist(x,y)<=radius) n++;
   return n;
}
```

Порядок выполнения.

1) Создаем динамическую строку нужного размера и вводим в нее исходные данные задания:

```
char* s = new char[4096]; int nc = myinput(s, "925.txt");
```

2) Создаем массив объектов `clCircle` и инициализируем его с помощью перегрузки `set()` (какие разделители должны быть в строке `sep`?).
3) Создаем массив указателей на объекты `clCircle` и числовой массив для значений, по которым выполняется сортировка.
4) Находим пересекающиеся круги и заносим в массивы их адреса и значения для сортировки.
5) Выполняем сортировку.
6) С помощью метода `get5()` выводим отсортированный список.
7) Освобождаем память, выделенную для динамических массивов.