Circles and Squares

Create an enumeration type which represents colors. Name at least 3 colors in this type.
(5 points)

Create two other types: one for representing circles, and another one for representing squares. Both structures should be able to store the center coordinates, furthermore the radius in case of the circle and the side length in case of the square. All these are integer numbers. The shapes have a color too which is given by the enumeration type above.
(5 points)

Create a function which can be given a square and a circle as parameter. The function returns a logical value depending on whether the circle is inside the square. For example the circle (x=1, y=2, r=3) is inside of square (x=2, y=0, l=11) but not inside of square (x=2, y=0, l=2).
(5 points)

Create a function which gets an array of circles and a square. The function should return an array on the heap that contains the circles of this array which are inside of this square. The returned array size goes to the caller through a pointer parameter. Make sure to avoid memory leak in the program.
(10 points)

Create a function which gets an array of circles and a color as parameter and returns a pointer to the smallest circle of that color from the array. In case there is no circle of that color in the array, then return a NULL pointer.
(10 points)

Place the functions to a separate translation unit. Use the "include guard" idiom.
(5 points)

The program can be given a file name as a command line argument. The first line of the file is a number which indicates the number of further lines. Those lines are describing circles with their x, y coordinates and radius respectively. Their color doesn't matter.

```
./my_program circles.txt

circles.txt
-----------
3
1 2 3
0 0 10
-2 5 9
```

Read the data (x, y coordinates, side length) of a square from the keyboard. Print the number of circles inside the square to the screen.
(10 points)

Bonus task:
Indicate the color of circles in the input file and read those too:

```
circles.txt
-----------
3
1 2 3 red
0 0 10 green
-2 5 9 blue
```

(+5 points)