Section 18 | **PDS Lab Test 2** | **Oct 29, 2019**
Odd set | **Full marks: $100 \times 4 = 400$** | **Time: 3 hours**

Note:

1. Marks will be deducted for inefficient coding, bad structuring of code, bad indentation, lack of important commenting, and deviation from input and output formats as shown in the examples.
2. You can use any library function unless forbidden in the question.
3. Name and submit your files as `o2a.c, o2b.c, o2c.c, o2d.c`.
4. In each file you must write your name, roll number, and machine number in the beginning as comment lines.

o2a. **(Sum of consecutive odd squares)** Given a positive integer n, check whether it can be expressed as the sum of squares of two consecutive odd integers.
You **cannot** use any loop.

Examples:

```
Enter n: 34      Enter n: 20         Enter n: 38090      Enter n: 38091
3 5              Does not exist.     137 139             Does not exist.
```

o2b. **(Number of lines)** Given a poem in a text file containing "`---End---`" at the end, count the number of lines in the poem.

Examples:

```
./a.out < poem1.txt          ./a.out < poem2.txt          ./a.out < poem3.txt
#lines = 8                   #lines = 8                   #lines = 24
```

o2c. **(Room allotment)** n rooms are to be allotted to $2n$ students so that 2 students occupy a single room. Given n as input, find the number of ways of allotment—both recursively and iteratively.
For example, for $n = 2$, possible allotments to Rooms 1 and 2 (taken in order) are:
$\{(s_1, s_2), (s_3, s_4)\}$, $\{(s_1, s_3), (s_2, s_4)\}$, $\{(s_1, s_4), (s_2, s_3)\}$,
$\{(s_2, s_3), (s_1, s_4)\}$, $\{(s_2, s_4), (s_1, s_3)\}$, $\{(s_3, s_4), (s_1, s_2)\}$;
and so the count is 6.
Note that $\{(s_1, s_2), (s_3, s_4)\}$ and $\{(s_3, s_4), (s_1, s_2)\}$ are two different allotments in this example, because the pair (s_1, s_2) is allotted to Room 1 in the former allotment but allotted to Room 2 in the latter. $(50 + 50)$

Examples:

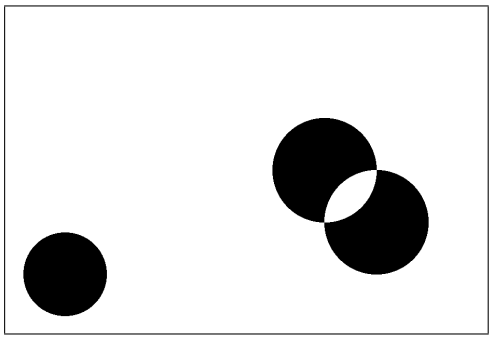
```
Enter #rooms: 1              Enter #rooms: 2              Enter #rooms: 3
#ways (recursive) = 1        #ways (recursive) = 6        #ways (recursive) = 90
#ways (iterative) = 1        #ways (iterative) = 6        #ways (iterative) = 90
```

o2d. **(Overlapping discs)** An input file (`disc1.txt` or `disc2.txt`) contains the information about n discs. Its 1st line contains the value of n, and each of the next n lines contains the (x, y) coordinates of the center and the radius (all are integers) of a disc. Your program has to print on the terminal the areas of all these discs. Assume that $\pi = 3.1416$.

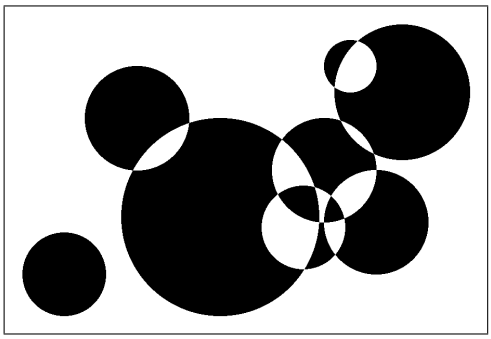
Now prepare an output binary image file named `o2d.pbm`. Image height = 600, width = 900, and each point p of the image is black if and only if p belongs to an odd number of discs. Note that p belongs to a disc of radius r if its distance from the center of the disc is at most $\sqrt{r(r+1)}$.

You **cannot** use math library. $(40 + 60)$

Examples:



```
./a.out < disc1.txt
Areas = 31416.00, 31416.00, 20106.24.
```



```
./a.out < disc2.txt
Areas = 31416.00, 113411.76, 31416.00, 20106.24,
        7854.00, 31416.00, 53093.04, 20106.24.
```

Note: A `pbm` file contains "`P1`" in the 1st line, #columns and #rows in the 2nd line, and then each line represents a row of the image with '1' = black and '0' = white.