For the following questions, please organize your solutions so that each question can be tested independently of the other questions. You can organize the solutions in any way that makes sense to you. Don't feel you need to finish every question in three hours. Instead, do each question to a level of quality you would be proud of releasing to a client.

1. Given two database tables:

```
    table students:
    | id | name | address |

    table applications:
    | id | student_id | score |
```

Write a query to list the number of applications each student has. Report 0 for students with no applications. List students first by number of applications, then alphabetically by name.

2. Write a program that reads a CSV file with information about different regular polygons (**all sides equal**), and circles. Each row of the file contains the name of the polygon and its side length, in order. Acceptable polygons include:

- triangle
- square
- pentagon

Note also that the shape "circle" should be accepted. Of course, a circle has no side length; instead, the dimension refers to its radius.

Here is an example CSV file:

```
triangle,3.5
circle,2
```

Given the input above, your program should output:

```
A triangle with side length 3.5 u has a perimeter of 10.5 u and an area of 5.30 u^2
A circle with radius 2 u has a perimeter of 12.57 u and an area of 12.57 u^2
```

Please write your solution using classes and objects as appropriate.

3. Given a raw string of html with no formatting like:
`<html><body><div><a></a></div></body></html>`

Write a program to produce the following output:

```
<html>
```

```
  <body>
    <div>
      <a>
      </a>
    </div>
  </body>
</html>
```

If the html is not valid (ie, there are mismatched or missing tags), print an error. Thus, the following two strings should produce errors:

```
<html><body><div></a></body></html>
<html><body><div><a></div></a>
```

Read in strings from stdin. Return a non zero exit code on error.

4. For the array of letters (a, b, c, d, e, f), write a **recursive** function that outputs the following structure:

```
<a>
 <b>
  <c>
   <d>
    <e>
     <f>
     </f>
    </e>
   </d>
  </c>
 </b>
</a>
```