Assignment 2 Computer Science I Fall 2018

Instructions Follow instructions *carefully*, failure to do so may result in points being deducted. Hand in all your source code files through webhandin and make sure your programs compile and run by using the webgrader interface. You can grade yourself and re-handin as many times as you wish up until the due date.

Naming Instructions Place your code in source files with the file names `rainfall.c`, `agm.c`, and `isotope.c` respectively. Do some rudimentary input validation and exit the program with an error message on any erroneous input.

Programs

1. Implement a program to solve the classic "Rainfall Problem" which has been used in CS education studies to assess programming knowledge and skills. Write an *interactive* program that repeatedly prompts the user to enter an integer which represents the amount of rainfall on a particular day. The program should continue prompting and reading figures until the user enters the integer 99999. After 99999 is entered, it should compute and print the *correct* average rainfall. That is, it should not count the final 99999. In addition, negative values should be ignored but not cause an error or termination of the program. For example, if the user entered the sequence `4 0 -1 10 99999` the output should look something like the following.

   ```
   Average rainfall: 4.66
   ```

2. The *arithmetic-geometric* mean of two positive numbers x, y, denoted $M(x, y)$ (or $\text{agm}(x, y)$) can be computed iteratively as follows. Initially, $a_1 = \frac{1}{2}(x + y)$ and $g_1 = \sqrt{xy}$ (i.e. the normal arithmetic and geometric means). Then, compute

   $$\begin{aligned} a_{n+1} &= \tfrac{1}{2}(a_n + g_n) \\ g_{n+1} &= \sqrt{a_n g_n} \end{aligned}$$

   The two sequences will converge to the same number which is the arithmetic-geometric mean of x, y. Obviously we cannot compute an infinite sequence, so we compute until $|a_n - g_n| < \epsilon$ for some small number ϵ.

   Write a program that computes the arithmetic-geometric mean of two positive numbers by reading x, y and ϵ as *command line arguments* and outputs the AGM to the standard output.

3. The rate of decay of a radioactive isotope is given in terms of its half-life H, the time required for the isotope to decay to one-half of its original mass. For example, the radioactive isotope of Caesium, Cs-137 has a half-life of 30.17 years. If we start with 10kg of Caesium-137 then 30.17 years later you would expect to have only 5kg of Caesium-137 (and 5kg of Barium-137 a stable isotope which is a decay product of Caesium-137).

   Write a program that takes the following inputs as command line arguments:

   - Atomic Number (integer)
   - Element Name

- Element Symbol
- H (half-life in years of the element)
- m, an initial mass in grams

Your program will then produce a table detailing the amount of the element that remains after each year until less than 50% of the original amount remains. This amount can be computed using the following formula:

$$r = m \times \left(\frac{1}{2}\right)^{(y/H)}$$

where y is the number of years elapsed, and H is the half-life of the isotope in years.

For example, using your program on Caesium-137 (symbol: Cs, atomic number: 56) with a half-life of 30.17 years and an initial amount of 10 grams would produce a table something like the following.

```
Caesium (56-Cs)
Year Amount
---------------
--      10.000g
 1       9.773g
 2       9.551g
 3       9.334g
 4       9.122g
...
29       5.136g
30       5.020g
31       4.906g
```