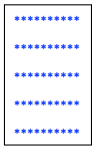
CSCI 2202 Computer Modelling for Scientists Lab 3 Due: 25 Jan 21

Lists and Introduction to Projectile Motion

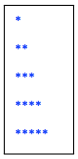
Lists and More Turtles:

1. (a) Create a program `staRect.py`. The program takes two integers, R and C, as inputs and prints an R C rectangle array of asterisks (*). For example, R = 5, C = 10 should produce:



Note: When you use `print("A")` *it adds a newline character (*`''\n''`*). To avoid the newline character, use* `print("A", end='')`

   (b) Modify `staRect.py` to accept a single integer R as input and produce a triangle with a single star in the first row, two in the second, etc. Save the program as `sTri.py`.



2. In this exercise, we explore projectile motion, using the `turtle` module. Consider the motion of a projectile, launched at a velocity v_0 at an angle θ to the horizontal (x-axis). The simplest approximation to the motion of the projectile is when the *only* force acting on the projectile is its own weight: mg where g is the acceleration due to gravity ($9.81 m/s^2$). Using the equations of motion ($v(t)$ - velocity; s - displacement, t - time):

$$v = v_0 - g * t \tag{1}$$

$$s = s_0 + v_0 * t - \frac{1}{2} g * t^2 \tag{2}$$

The motion of the projectile can be broken down into two independent components of the motion:

(i) Along the x-axis, with an acceleration $a_x = 0$ and an initial velocity $v_{0x} = v_0 * \cos\theta$ and *(ii)* Along the y-axis, with an acceleration $a_y = -g$ (the negative sign indicates that it is pointing downwards.) and the initial velocity: $v_{0y} * \sin\theta$ Along the horizontal and vertical directions:

$$v_x = v_0 * \cos\theta \qquad \text{and} \qquad x = x_0 + v_0 * \cos\theta * t. \tag{3}$$

$$v_y = v_0 * \sin\theta - g * t \qquad \text{and} \qquad y = y_0 + v_0 * \sin\theta * t - \frac{1}{2} g * t^2 \tag{4}$$

Using the x and y equations above, plot the trajectory of a turtle launched at an angle θ, and initial velocity v_0 (entered by the user). To compute cos, sin etc. `import math`, *convert the angle entered in degrees, θ, into radians ($\theta * \pi/180$). Use as:* `math.cos`*($\theta * \pi/180$). The math module expects angles in radians). Fire the projectile from* $(x_0, y_0) = (-200, 0)$.

- Pick a `travel_time` that will (at least) cover the entire range of the x motion (experiment to find the right `travel_time`).
- Compute the (x, y) position at small time increments (using the x and y equations above) making up the total `travel_time`. A convenient way to do this is to use the `range` function in the form: `range(start, stop, step)`. The `step parameter` allows you to control the increment (the default increment is 1).
  (See: `https://docs.python.org/3/library/functions.html#func-range`).
  Use the `turtle stamp()` function to leave an impression of the turtle at each time-increment.
- Create lists for x, y and *time* and `append` each new x, y and *time* value to the respective lists. Print the lists (side-by-side) after the turtle has finished moving.
- Modify your program to compute *an estimate of* and print the max. distance the turtle travels in the x-direction (the "range"). The *range* is found by finding the *time* at which the y position of the turtle first becomes ≤ 0, and then using that time in the x equation.
- Modify your program to find the maximum height the projectile reaches. When the projectile is at its maximum height, $v_y = 0$. You can *approximate* this using the v_y equation: Find the time when v_y switches from positive to negative. Use the time in the y equation to find the max. height.

- Finally, save a copy of your program and change it to create a list of firing angles θ from 20 to 80 degrees, in steps of 5 degrees (with fixed value for v_0) and make a list of the max. heights reached and a list of the max. x-value (range) reached, for each value in the list of firing angles.
  Below is the turtle trajectory for $x_0 = -200,\ y_0 = 0$, $\theta = 60$, $v_0 = 70m/s$