

Mini-Project 1:

Fundamentals of Problem Solving with Python

1. Seeing Stars

This program prints left-aligned star pyramids with a size determined by the user. *This program should continue to run until the user quits themselves.* Name your file `stars.py`.

Input

An integer, `n`, taken through `raw_input()` from the user

Output

`n` lines of asterisks where the number of asterisks in each line increases from 1 to `n`, i.e. the first line has one asterisk and the `n`th line has `n` asterisks.

Sample Output

Enter an integer: 4

```
*
**
***
****
```

Enter an integer: 9

```
*
**
***
****
*****
*****
*****
*****
*****
*****
```

Enter an integer: 1

```
*
```

2. Is it a palindrome?

This program determines if a word is a palindrome, i.e. if the word is spelled the same way backwards as it is forwards. *This program should continue to run until the user quits themselves.* Name your file `palindrome.py`.

Note: There is not a built-in function for reversing a String, and if you find `[::-1]` on the Internet and use it, be prepared to explain how that functionality works.

Input

A string from the user via `raw_input()`

Output

An indication of if the word is a palindrome or not.

Sample Output

Enter a word.

A but tuba

A but tuba is a palindrome.

Enter a word.

This class rules

This class rules is not a palindrome.

Enter a word.

Hannah

Hannah is a palindrome.

Enter a word.

See what I did there?

See what I did there? is not a palindrome.

Enter a word.

Acrobats stab orca

Acrobats stab orca is a palindrome.

Enter a word.

A nut for a jar of tuna

A nut for a jar of tuna is a palindrome.

3. Duke Sucks

This program asks the user what their favorite basketball team is until the user comes to Jesus and says their favorite team is Carolina (with a special message for those that say Duke). The program should terminate only after the user has input "Carolina". To terminate a program before it has finished executing, use the function `sys.exit()` within the `sys` module. Name your file `dukesucks.py`.

Input

The user's favorite basketball team (a String via `raw_input()`)

Output

Depending on the user's input, the program will output one of three things:

1. Excellent choice! Goodbye if the user inputs "Carolina"
2. Duke sucks and so do you! if the user inputs "Duke"
3. No it's not. Try again. if the user inputs anything else

Sample Output

What is your favorite basketball team?

NC State

No it's not. Try again.

What is your favorite basketball team?

Duke

Duke sucks and so do you!

What is your favorite basketball team?

Alabama

No it's not. Try again.

What is your favorite basketball team?

Kentucky

No it's not. Try again.

What is your favorite basketball team?

Carolina

Excellent choice! Goodbye

Repl Closed

4. Celestial Bodies

This program takes information about about celestial bodies, e.g. stars, planets, and asteroids, and does various qualitative and quantitative analyses. *This program should continue to run until the user quits themselves.* Name your program `planets.py`.

Input

The program will prompt the user for two values:

1. The celestial object they want to know about
2. Which of seven options they'd like to select to learn something about that object ((0) Radius, (1) Diameter, (2) Circumference, (3) Mass, (4)

Classification, (5) Surface Area, (6) Size compared to Earth (ratio of surface areas))

The data will come from a text file containing the name, mass (in Earth masses), radius (in kilometers), and type of the celestial body separated by a space, e.g:

```
Mercury 0.055 2440 planet
Venus 0.815 6052 planet
Moon 0.012 1737.5 moon
```

and so on. Note that this program should work for *any* text file of this format, not just with the ordering in `objectInfo.txt`. You can assume that the program will not be queried with a name that does not exist in the file.

Output

There will be different functions to produce different outputs as a result of different computations on this data. Your program should use the following functions:

- `getRadius()`
- `getDiameter()`
- `getCircumference()`
- `getSurfaceArea()`
- `getMass()`
- `getType()`
- `compareSurfAreaToEarth()`

Sample Output

Which object would you like to know about?

Mars

What would you like to know about it?

(0) Radius

(1) Diameter

(2) Circumference

(3) Mass

(4) Classification

(5) Surface Area

(6) Size compared to Earth (ratio of surface areas)

0

The radius of Mars is 3390 km.

Which object would you like to know about?

Ceres

What would you like to know about it?

(0) Radius

- (1) Diameter
- (2) Circumference
- (3) Mass
- (4) Classification
- (5) Surface Area
- (6) Size compared to Earth (ratio of surface areas)

4

Ceres is an asteroid

Which object would you like to know about?

Moon

What would you like to know about it?

- (0) Radius
- (1) Diameter
- (2) Circumference
- (3) Mass
- (4) Classification
- (5) Surface Area
- (6) Size compared to Earth (ratio of surface areas)

4

Moon is a moon

Which object would you like to know about?

Neptune

What would you like to know about it?

- (0) Radius
- (1) Diameter
- (2) Circumference
- (3) Mass
- (4) Classification
- (5) Surface Area
- (6) Size compared to Earth (ratio of surface areas)

6

Neptune is 14.9359016011 times larger than Earth.

Submission

You will submit 3 Python files:

1. stars.py
2. palindrome.py
3. dukesucks.py
4. planets.py

Your output should match the sample outputs for at least the provided inputs, i.e. if you run the same inputs that are in the sample output sections for each program, you should have identical output. This includes formatting but not necessarily spelling!

This is due by 5:00PM on Friday, March 6th (aka the start of your Spring Break).