You have to program several functions. In each case, few lines of code are enough.

1. Write a function \texttt{drawH}(n) that given an odd integer \( n \geq 3 \) prints a letter \( \text{H} \) of size \( n \) formed with symbol \( * \). Follow the pattern of the example below.

2. Write a function \texttt{area\_circle}(r) that given a float number \( r \geq 0 \) returns, rounded to the hundredth, the area of the circle of radius \( r \). Use the \( \pi \) constant defined in the math module.

3. Write a function \texttt{slow\_pi\_aprox}(n) that given a non negative integer \( n \) computes \( 4 \sum_{k=0}^{n} \frac{(-1)^k}{2k+1} \) rounded to the hundredth.

4. Write a function \texttt{is\_univariate\_number}(n) that given a non negative integer returns a boolean pointing out whether \( n \) is represented using only one digit. For instance 22222 is univariate but 22322 is not.

5. Write a boolean function \texttt{is\_univariate\_word}(s) that given an string \( s \) returns \texttt{True} if and only if \( s \) is formed using only one letter. For instance word \texttt{xxxXXXx} is univariate but \texttt{xxxXxy} is not. We assume \( s \) is non empty and all characters of \( s \) are letters.

**Scoring**

Every function counts 20 points.

**Sample session**

```python
>>> drawH(5)
**
**
*****
**
**
>>> area_circle(2.5)
19.63
>>> slow_pi_aprox(50)
3.16
>>> is_univariate_number(22322)
False
>>> is_univariate_word("xxXxXXx")
True
```

**Problem information**

Author : Jorge Castro
Generation : 2016-09-20 09:23:35

http://www.jutge.org