DSA Introduction Class

Logic building with plain English and pseudocode

This short handout focuses on clear thinking. We will solve three simple problems using everyday language and neat pseudocode. For each one we go through the idea, simple reasoning, examples, and a final plan you can code in any language.

1) Strobogrammatic Number

A number is strobogrammatic if it looks the same after a 180 degree rotation. Use this mapping: 0->0, 1->1, 6->9, 8->8, 9->6. Digits 2, 3, 4, 5, 7 do not work.

Important rule for this class

If the string has more than one character and starts with 0, we treat it as not valid. So leading zero means false here.

How to think about it

Put one pointer on the left (L) and one on the right (R). At each step check if the left digit maps to the right digit by the mapping above. Move L forward and R backward until they cross. If all pairs match, it is strobogrammatic.

Examples

```
A) s = "818"
Pairs: (8,8) ok, middle 1 ok. Answer: true.

B) s = "69"
Pairs: (6,9) ok. Answer: true.

C) s = "960"
Pairs: (9,0). 9 maps to 6 not 0. Answer: false.

D) s = "101"
Pairs: (1,1) ok, middle 0 ok. Answer: true.

E) s = "2"
2 is not in the mapping. Answer: false.

F) s = "010"
Starts with 0 and length > 1. By our rule, Answer: false.
```

Pseudocode

```
function isStrobogrammatic(s):
    # reject leading zero
    if len(s) > 1 and s[0] == '0':
        return False

    map = { '0':'0', '1':'1', '6':'9', '8':'8', '9':'6' }
    L = 0
    R = len(s) - 1
    while L <= R:
        if s[L] not in map: return False
        if map[s[L]] != s[R]: return False
        L = L + 1
        R = R - 1
    return True
```

Notes

Time is O(n) because we scan from both ends once. Space is O(1). Single character is true only for 0, 1, and 8.

2) Find all vowels in a string

Walk through a string and record each vowel and its index. Vowels are a, e, i, o, u. We ignore case so A is also a vowel.

How to think about it

Go from left to right. For each character, check if it is a vowel. If yes, store the character and its position. This is a simple pass.

Examples

```
A) "Hello World!"
Vowels: e@1, o@4, o@7

B) "AEiou"
Vowels: A@0, E@1, i@2, o@3, u@4

C) "sky"
Vowels: none
```

Pseudocode

```
function listVowelsWithIndices(s):
    vowels = set('aeiouAEIOU')
    result = []
    for i from 0 to len(s)-1:
        if s[i] in vowels:
            append (s[i], i) to result
    return result
```

Notes

Time is O(n). If you only need unique vowels, keep a seen set and skip repeats.

3) Pattern printing

We will print two centered star patterns exactly as shown. The plan is to decide star counts per line and add the right number of spaces on the left.

Pattern A (n = 7)

```
Target:
   *
  ***
 *****

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

Plan

Use width = 7. Print lines with 1, 3, and 5 stars, then a blank line, then 5, 3, and 1 stars. Left spaces are (width - stars) / 2.

Pseudocode

```
function pattern_n_equals_7():
    width = 7
    for stars in [1, 3, 5]:
        spaces = (width - stars) / 2
        print ' ' repeated spaces + '*' repeated stars
    print empty line
    for stars in [5, 3, 1]:
        spaces = (width - stars) / 2
        print ' ' repeated spaces + '*' repeated stars
```

Pattern B (n = 5)

```
Target:
    *
   ***
  *****
 *******
```

Plan

Use width = 7. Print four lines with 1, 3, 5, and 7 stars. Left spaces are (width - stars) / 2.

Pseudocode

```
function pattern_n_equals_5():
    width = 7
    for i from 1 to 4:
        stars = 2*i - 1
        spaces = (width - stars) / 2
        print ' ' repeated spaces + '*' repeated stars
```

Wrap up

We focused on simple thinking and clean steps. Try turning the pseudocode into your favorite language, and test with short and long strings. As you practice, explain your plan out loud in plain English before you write code. This habit makes you both clear and fast.