

## Choosing a Fuse

A fuse melts when a current in an electrical device exceeds the fuse's rating, breaking the circuit and preventing the heat from building up too much (which can cause a fire). The ideal fuse to choose is higher than the device's current output, yet as close as possible to it as well. Given a list of fuse ratings, and the de ...

algorithms numbers

Medium

## Choosing a Fuse

Published by Joshua Señoron in Python

algorithms numbers

A fuse melts when a current in an electrical device exceeds the fuse's rating, breaking the circuit and preventing the heat from building up too much (which can cause a fire). The ideal fuse to choose is **higher** than the device's current output, yet **as close as possible** to it as well.

Given a list of *fuse ratings*, and the *device's current output*, return which of the fuses is the best for the device.

### Examples

```
choose_fuse(["3V", "5V", "12V"], "4.5V") → "5V"
choose_fuse(["5V", "14V", "2V"], "5.5V") → "14V"
choose_fuse(["17V", "15V", "12V", "9V"], "9V") → "12V"
```

### Notes

- You will be given three possible ratings in voltage.
- Fuses may not be in a sorted order.
- Assume that there is a valid fuse in every test case

ทำให้โปรแกรมเลือกฟิวส์ ที่ใกล้เคียงและมากกว่ากับฟิวส์ที่ตั้งขึ้น เพื่อป้องกันไฟไหม้ โดยการหาตัวที่มีค่ามากกว่าและใกล้เคียง

จากโค้ดจะเห็นได้ว่า ตัวตั้ง 4.5 โปรแกรมจะหาค่าที่มากกว่าซึ่งจะมีเพียง 2 ค่า คือ 5 และ 12 แต่ตัวที่ใกล้เคียงมากที่สุดก็คือ 5 จึงได้ผลลัพธ์เป็น 5

```
#Choosing a Fuse
def choose_fuse(fuses, current):
    arr = sorted(fuses + [current], key=lambda x: float(x[:-1]))
    return arr[arr.index(current) + 1]

print(choose_fuse(["3V", "5V", "12V"], "4.5V"))
```

```
PS C:\Users\Heart
rs/Heart Admin/Des
5V
PS C:\Users\Heart
```

Published by [Joshua Señoron](#) in [Python](#) ▾

Given a list of strings (depicting a skyline of several buildings), return in meters the height of the tallest building. Each line in the list represents 20m.

```

tallest_building_height{
  10  ##"
  20  ##"
  30  ##"
  "" ## ##"
  "" ## ## ##"
  "" ## ## ##"
  "" ## ## ## ##"
} → "140m"

```

```

tallest_building_height{
  10
  20
  30
  40
  50
  60
  70
  80
  90
  100
  110
  120
  130
  140
  150
  160
  170
  180
  190
  200
  210
  220
  230
  240
  250
  260
  270
  280
  290
  300
  310
  320
  330
  340
  350
  360
  370
  380
  390
  400
  410
  420
  430
  440
  450
  460
  470
  480
  490
  500
  510
  520
  530
  540
  550
  560
  570
  580
  590
  600
  610
  620
  630
  640
  650
  660
  670
  680
  690
  700
  710
  720
  730
  740
  750
  760
  770
  780
  790
  800
  810
  820
  830
  840
  850
  860
  870
  880
  890
  900
  910
  920
  930
  940
  950
  960
  970
  980
  990
  1000
  1010
  1020
  1030
  1040
  1050
  1060
  1070
  1080
  1090
  1100
  1110
  1120
  1130
  1140
  1150
  1160
  1170
  1180
  1190
  1200
  1210
  1220
  1230
  1240
  1250
  1260
  1270
  1280
  1290
  1300
  1310
  1320
  1330
  1340
  1350
  1360
  1370
  1380
  1390
  1400
  1410
  1420
  1430
  1440
  1450
  1460
  1470
  1480
  1490
  1500
  1510
  1520
  1530
  1540
  1550
  1560
  1570
  1580
  1590
  1600
  1610
  1620
  1630
  1640
  1650
  1660
  1670
  1680
  1690
  1700
  1710
  1720
  1730
  1740
  1750
  1760
  1770
  1780
  1790
  1800
  1810
  1820
  1830
  1840
  1850
  1860
  1870
  1880
  1890
  1900
  1910
  1920
  1930
  1940
  1950
  1960
  1970
  1980
  1990
  2000
  2010
  2020
  2030
  2040
  2050
  2060
  2070
  2080
  2090
  2100
  2110
  2120
  2130
  2140
  2150
  2160
  2170
  2180
  2190
  2200
  2210
  2220
  2230
  2240
  2250
  2260
  2270
  2280
  2290
  2300
  2310
  2320
  2330
  2340
  2350
  2360
  2370
  2380
  2390
  2400
  2410
  2420
  2430
  2440
  2450
  2460
  2470
  2480
  2490
  2500
  2510
  2520
  2530
  2540
  2550
  2560
  2570
  2580
  2590
  2600
  2610
  2620
  2630
  2640
  2650
  2660
  2670
  2680
  2690
  2700
  2710
  2720
  2730
  2740
  2750
  2760
  2770
  2780
  2790
  2800
  2810
  2820
  2830
  2840
  2850
  2860
  2870
  2880
  2890
  2900
  2910
  2920
  2930
  2940
  2950
  2960
  2970
  2980
  2990
  3000
  3010
  3020
  3030
  3040
  3050
  3060
  3070
  3080
  3090
  3100
  3110
  3120
  3130
  3140
  3150
  3160
  3170
  3180
  3190
  3200
  3210
  3220
  3230
  3240
  3250
  3260
  3270
  3280
  3290
  3300
  3310
  3320
  3330
  3340
  3350
  3360
  3370
  3380
  3390
  3400
  3410
  3420
  3430
  3440
  3450
  3460
  3470
  3480
  3490
  3500
  3510
  3520
  3530
  3540
  3550
  3560
  3570
  3580
  3590
  3600
  3610
  3620
  3630
  3640
  3650
  3660
  3670
  3680
  3690
  3700
  3710
  3720
  3730
  3740
  3750
  3760
  3770
  3780
  3790
  3800
  3810
  3820
  3830
  3840
  3850
  3860
  3870
  3880
  3890
  3900
  3910
  3920
  3930
  3940
  3950
  3960
  3970
  3980
  3990
  4000
  4010
  4020
  4030
  4040
  4050
  4060
  4070
  4080
  4090
  4100
  4110
  4120
  4130
  4140
  4150
  4160
  4170
  4180
  4190
  4200
  4210
  4220
  4230
  4240
  4250
  4260
  4270
  4280
  4290
  4300
  4310
  4320
  4330
  4340
  4350
  4360
  4370
  4380
  4390
  4400
  4410
  4420
  4430
  4440
  4450
  4460
  4470
  4480
  4490
  4500
  4510
  4520
  4530
  4540
  4550
  4560
  4570
  4580
  4590
  4600
  4610
  4620
  4630
  4640
  4650
  4660
  4670
  4680
  4690
  4700
  4710
  4720
  4730
  4740
  4750
  4760
  4770
  4780
  4790
  4800
  4810
  4820
  4830
  4840
  4850
  4860
  4870
  4880
  4890
  4900
  4910
  4920
  4930
  4940
  4950
  4960
  4970
  4980
  4990
  5000
  5010
  5020
  5030
  5040
  5050
  5060
  5070
  5080
  5090
  5100
  5110
  5120
  5130
  5140
  5150
  5160
  5170
  5180
  5190
  5200
  5210
  5220
  5230
  5240
  5250
  5260
  5270
  5280
  5290
  5300
  5310
  5320
  5330
  5340
  5350
  5360
  5370
  5380
  5390
  5400
  5410
  5420
  5430
  5440
  5450
  5460
  5470
  5480
  5490
  5500
  5510
  5520
  5530
  5540
  5550
  5560
  5570
  5580
  5590
  5600
  5610
  5620
  5630
  5640
  5650
  5660
  5670
  5680
  5690
  5700
  5710
  5720
  5730
  5740
  5750
  5760
  5770
  5780
  5790
  5800
  5810
  5820
  5830
  5840
  5850
  5860
  5870
  5880
  5890
  5900
  5910
  5920
  5930
  5940
  5950
  5960
  5970
  5980
  5990
  
```

```
tallest_building_height([
  60,
  61,
  62,
  63,
  64,
  65,
  66,
  67,
  68,
  69,
  70,
  71,
  72,
  73,
  74,
  75,
  76,
  77,
  78,
  79,
  80,
  81,
  82,
  83,
  84,
  85,
  86,
  87,
  88,
  89,
  90,
  91,
  92,
  93,
  94,
  95,
  96,
  97,
  98,
  99,
  100])
```

ใส่ข้อความจำพวกสตริง เพื่อหาความสูงของตึกที่สูงที่สุดโดยกำหนดเป็น  
# แทน 20 เมตรหากไม่มีก็ถือว่าไม่นับ

```
#Height of the Tallest Building
def tallest_building_height(lst):
    return str(20*sum('#' in h for h in lst))+ 'm'

print(tallest_building_height([
    "      ##",
    "      ##",
    "      ##",
    "###   ##",
    "###   ##",
    "###   ##",
    "###   ##"
]))
```

```
PS C:\Users\Heart Admin\140m
```

- There may be some open sky above buildings (can't just find the length of the list).
- There may be multiple tallest buildings (see example #2).

●●●

[algorithms](#) [arrays](#) [strings](#)

Medium

## Domino Chain

Published by [Mubashir Hassan](#) in [Python](#)

algorithms

strings

validation

Mubashir was playing with dominos. He concluded that:

- If the first domino is pushed over, it will **keep tipping next dominos to its right**.
- Reaction will stop if a domino is **already tipped over**, or if there is an **empty space**.



Create a function which takes a string of current status of the `dominos` and returns the string after `dominos chain` reaction.

- `|` represents a standing domino.
- `/` represents a tripped domino.
- represents an empty space.

จากโจทย์ให้สร้าง โปรแกรมโดยมีเงื่อนไข

`|` แสดงถึงโดมิโนที่ยืนอยู่

`/` แสดงถึงโดมิโนที่ล้ม

หมายถึงพื้นที่ว่าง

โดยจากโปรแกรมจะเห็นว่า หากมี `/` หรือ  จะไม่ทำให้ `|` กลายเป็น `/` ดังคำตอบ

## Domino Chain

Mubashir was playing with dominos. He concluded that: If the first domino is pushed over, it will keep tipping next dominos to its right. Reaction will stop if a domino is already tipped over, or if there is an empty space. Mubashir Create a function which takes a string of current status of the dominos and returns the st ...

algorithms

strings

validation

Medium

```
#Domino Chain
```

```
def domino_chain(dominos):  
    falling, res = True, ''  
  
    for i in dominos:  
        if i == '|' and falling:  
            res += '/'  
        else:  
            res += i  
            falling = False  
    return res
```

```
print(domino_chain("| | | | | / | /"))
```

```
PS C:\Users\Heart  
rs\Heart Admin\Des  
/// |||/// |
```

## A Simple Modification

Published by [Mubashir Hassan](#) in [Python](#) ▾

algorithms

interview

numbers

strings

**Mubashir** needs your help to learn Python Programming. Help him by modifying a given string `txt` as follows:

- Reverse the string given.
- Replace each letter to its position in the alphabet for example (a = 1, b = 2, c = 3, ...).
- Join the array and convert it to a number.
- Convert the number to binary.
- Convert the string back to a number.

See below example for more understanding :

`modify("hello")` → `111001101011101101101010`

```
"hello" = "olleh"
"olleh" = ['15', '12', '12', '5', '8']
['15', '12', '12', '5', '8'] = 15121258
15121258 = "111001101011101101101010"
"111001101011101101101010" = 111001101011101101101010
```

## A Simple Modification

Mubashir needs your help to learn Python Programming. Help him by modifying a given string `txt` as follows: Reverse the string given. Replace each letter to its position in the alphabet for example (a = 1, b = 2, c = 3, ...). Join the array and convert it to a number. Convert the number to binary. Convert the string back to ...

algorithms

interview

numbers

strings

Medium

### #A Simple Modification

```
def modify(t):
    return int(bin(int(''.join([str(ord(i)-96) for i in t[::-1]])))[2:])

print(modify("hello"))
```

```
PS C:\Users\Heart Admin\Desktop> python problem.py
111001101011101101101010
PS C:\Users\Heart Admin\Desktop>
```

การเปลี่ยนค่า ข้อความที่ป้อนโดยมีเงื่อนไข

- ย้อนกลับข้อความ > แทนลำดับตัวอักษร A-Z 1-26 > แปลงเป็นตัวเลข > แปลงตัวเลขเป็นไบนารี > แปลงสตริงกลับเป็นตัวเลข
- ดังคำตอบของโปรแกรม

# Digital Cipher

Published by [Mubashir Hassan](#) in [Python](#) ▾

[algorithms](#) [cryptography](#) [logic](#)

In **Digital Cipher**, encoding is done by the simple addition of numbers in the key and the corresponding characters on a string input.

Create a function that takes two arguments; a positive integer and a string and returns an encoded list of integers as message.

Assign a unique number to each letter of the alphabet.

```
a b c d e f g h i j k l m
1 2 3 4 5 6 7 8 9 10 11 12 13
n o p q r s t u v w x y z
14 15 16 17 18 19 20 21 22 23 24 25 26
```

There are some variations on the rules of encipherment. One version of the cipher rules are outlined below:

```
message = "scout"
key = 1939

digital_cipher(message, key) → [20, 12, 18, 30, 21]
```

Write the corresponding number against each character:

```
s c o u t
19 3 15 21 20
```

Add to each obtained digit consecutive digits from the key:

```
s c o u t
19 3 15 21 20
+ 1 9 3 9 1
-----
20 12 18 30 21
```

See the below example for a better understanding:

```
message = "masterpiece"
key = 1939

digital_cipher(message, key) → [14, 10, 22, 29, 6, 27, 19, 18, 6, 12, 8]

m a s t e r p i e c e
13 1 19 20 5 18 16 9 5 3 5
+ 1 9 3 9 1 9 3 9 1 9 3
-----
14 10 22 29 6 27 19 18 6 12 8
```

## Digital Cipher

In Digital Cipher, encoding is done by the simple addition of numbers in the key and the corresponding characters on a string input. Create a function that takes two arguments; a positive integer and a string and returns an encoded list of integers as message. Assign a unique number to each letter of the alphabet. a b ...

[algorithms](#) [cryptography](#) [logic](#)

Medium

```
#Digital Cipher
def digital_cipher(message, key):
    k=list(map(int,str(key)))
    l=len(k)
    return [ord(x)-96+k[i%l] for i,x in enumerate(message)]

print(digital_cipher("scout", "1939"))
```

```
20, 12, 18, 30, 21]
```

สร้างฟังก์ชันที่รับสองอาร์กิวเมนต์ จำนวนเต็มบวกและสตริงและส่งกลับรายการจำนวนเต็มทีี่เข้ารหัสเป็นข้อความ จากนั้น กำหนดหมายเลขเฉพาะให้กับตัวอักษรแต่ละตัว(รูปด้านบน) จากนั้นให้แทนตัวอักษรนั้นๆด้วยตัวเลขและ + ด้วย คียตามหลักจะได้คำตอบดังรูปด้านล่าง

# ABACABADABACABA

Published by [mkjlong](#) in [Python](#) ▾

algorithms

recursion

Create a function that follows the "ABACABADABACABA" rule up to a certain letter.

The *abacabadabacaba* pattern is where you start with the first letter (a), and with each new letter, you add the letter in the middle and the others at the start and end.

For instance:

```
A → **A**
B → A**B**A
C → ABA**C**ABA
D → ABACABA**D**ABACABA
E → ABACABADABACABA**E**ABACABADABACABA
F → ABACABADABACABAEABACABADABACABA**F**ABACABADABACABAEABACABADABACABA

# And so on ...
```

## ABACABADABACABA

Create a function that follows the "ABACABADABACABA" rule up to a certain letter. The *abacabadabacaba* pattern is where you start with the first letter (a), and with each new letter, you add the letter in the middle and the others at the start and end. For instance: A → A B → ABA C → ABACABA D → ABACABADABACABA E → ABACABA ...

algorithms

recursion

Medium

```
#ABACABADABACABA
def ABA(s):
    res = ''
    for i in range(ord('A'), ord(s)+1):
        res = res + chr(i) + res
    return res

print(ABA("B"))
```

```
PS C:\Users\Heart A
eart Admin/Desktop/
ABA
PS C:\Users\Heart A
```

การสร้างตัวอักษรขึ้นมา โดยเริ่มจาก A วนไปเรื่อยๆจนถึงตัวที่ต้องการโดยการวน ABACABAD จะเห็นได้ว่ามันจะวนเพิ่มทีละตัวจนกว่าจะถึงตัวที่ต้องการดังภาพ



# Coin Co-Operation

Published by [Joshua Señoron](#) in [Python](#)

algorithms

interview

language\_fundamentals

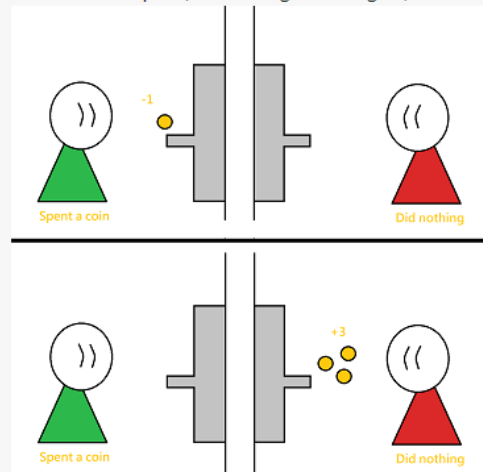
loops

Let's say that there exists a machine that gives out free coins, but with a twist!

Separating two people is a wall, and this machine is placed in such a way that both people are able to access it. Spending a coin in this machine will give the person on the other side **3 coins** and vice versa.

If both people continually spend coins for each other (SHARING), then they'll both gain a net profit of **2 coins per turn**. However, there is always the possibility for someone to act selfishly (STEALING): they spend no coins, yet they still receive the generous **3 coin** gift from the other person!

Here's an example of Red taking advantage of Green!



การแยกคนสองคนเป็นกำแพงและเครื่องนี้ถูกวางไว้ในลักษณะที่ทั้งสองคนสามารถเข้าถึงได้ การใช้เหรียญในเครื่องนี้จะให้คนที่อยู่อีกด้าน 3 เหรียญและในทางกลับกัน

หากทั้งสองคนใช้เหรียญให้กันและกันอย่างต่อเนื่อง (แบ่งปัน) ทั้งคู่จะได้กำไรสุทธิ 2 เหรียญต่อเทิร์น อย่างไรก็ตามมีความเป็นไปได้ที่ใครบางคนจะทำตัวเห็นแก่ตัว (ขโมย): พวกเขาไม่ใช้เหรียญ แต่พวกเขายังคงได้รับของขวัญ 3 เหรียญจากอีกฝ่ายสมมติว่าทั้งสองคนเริ่มต้นด้วยเหรียญ 3 เหรียญให้สร้างฟังก์ชันที่คำนวณจำนวนเหรียญสุดท้ายของทั้งสองคน คุณจะได้รับสองรายการของสตริงกับแต่ละสายเป็นคำหรือ 'share' 'steal'

## The Challenge

Assuming that both people **start with 3 coins each**, create a function that calculates both people's final number of coins. You will be given two lists of strings, with each string being the words `'share'` or `'steal'`.

## Coin Co-Operation

Let's say that there exists a machine that gives out free coins, but with a twist! Separating two people is a wall, and this machine is placed in such a way that both people are able to access it. Spending a coin in this machine will give the person on the other side 3 coins and vice versa. If both people continually spend ...

algorithms

interview

language\_fundamentals

loops

Medium

```
#Coin Co-Operation
def get_coin_balances(lst1, lst2):
    a, b = 3, 3
    for i, j in zip(lst1, lst2):
        if i == 'share': a, b = a-1, b+3
        if j == 'share': a, b = a+3, b-1
    return [a, b]

print(get_coin_balances(["share"], ["share"]))
```

```
eart Admin/Des
[5, 5]
PS C:\Users\He
```

# The Karaca's Encryption Algorithm

Published by [Furkan Karaca](#) in [Python](#)

algorithms

cryptography

formatting

Make a function that encrypts a given input with these steps:

Input: "apple"

Step 1: Reverse the input: "elppa"

Step 2: Replace all vowels using the following chart:

```
a => 0
e => 1
i => 2
o => 2
u => 3

# "1lpp0"
```

Step 3: Add "aca" to the end of the word: "1lpp0aca"

## The Karaca's Encryption Algorithm

Make a function that encrypts a given input with these steps: Input: "apple" Step 1: Reverse the input: "elppa" Step 2: Replace all vowels using the following chart: a => 0 e => 1 i => 2 o => 2 u => 3 "1lpp0" Step 3: Add "aca" to the end of the word: "1lpp0aca" Output: "1lpp0aca" Examples encrypt("banana") → "OnOn0ba ...

complete

algorithms

cryptography

formatting

Hard

```
#The Karaca's Encryption Algorithm
def encrypt(word):
    return word[::-1].translate(str.maketrans('aeou', '0123')) + 'aca'

print(encrypt("banana"))
```

```
earc Admin/De
0n0n0baca
PS C:\Users\H
```

อัลกอริทึมการเข้ารหัสของ Karaca

สร้างฟังก์ชันที่เข้ารหัสอินพุตที่กำหนดด้วยขั้นตอนเหล่านี้:

อินพุต: "apple"

ขั้นตอนที่ 1: ย้อนกลับอินพุต: "elppa"

ขั้นตอนที่ 2: แทนที่สระทั้งหมดโดยใช้แผนภูมิต่อไปนี้ (ทางด้านบน)

ขั้นตอนที่ 3: เพิ่ม "aca" ต่อท้ายคำ: "1lpp0aca"



## Binary Tree Nodes

Published by [Matt in Python](#)

[algorithms](#) [arrays](#) [conditions](#) [interview](#)

We have two lists `N` and `P`, where `N` represents the value of a node in Binary Tree, and `P` is the parent of `N`.

N	P
1	2
3	2
6	8
9	8
2	5
8	5
5	-1

Write a function to find the node type of the node within this Binary Tree, ordered by the value of the node. Output one of the following:

- `Root`: If node is root node.
- `Leaf`: If node is leaf node.
- `Inner`: If node is neither root nor leaf node.
- `Not exist`: If node not exist.

```
node_type([1, 3, 6, 9, 2, 8, 5], [2, 2, 8, 8, 5, 5, -1], 5) → "Root"
node_type([1, 3, 6, 9, 2, 8, 5], [2, 2, 8, 8, 5, 5, -1], 6) → "Leaf"
node_type([1, 3, 6, 9, 2, 8, 5], [2, 2, 8, 8, 5, 5, -1], 2) → "Inner"
node_type([1, 3, 6, 9, 2, 8, 5], [2, 2, 8, 8, 5, 5, -1], 10) → "Not exist"
```

## Binary Tree Nodes

We have two lists `N` and `P`, where `N` represents the value of a node in Binary Tree, and `P` is the parent of `N`. `N | P --- | -`  
`-- 1 | 2 3 | 2 6 | 8 9 | 8 2 | 5 8 | 5 5 | -1` Write a function to find the node type of the node within this Binary Tree, ordered by the value of the node. Output one of the following: `Root`: If node is ...

[algorithms](#) [arrays](#) [conditions](#) [interview](#)

Hard

```
#Binary Tree Nodes
def node_type(nodes, parents, value):
    if value not in nodes:
        return 'Not exist'
    if parents[nodes.index(value)] == -1:
        return 'Root'
    if value not in parents:
        return 'Leaf'
    return 'Inner'

print(node_type([1, 3, 6, 9, 2, 8, 5], [2, 2, 8, 8, 5, 5, -1], 5))
```

eart Admin/De  
Root  
PS C:\Users\H

เขียนฟังก์ชันเพื่อค้นหาประเภทโหนดของโหนดภายใน Binary Tree นี้โดย  
เรียงลำดับตามค่าของโหนด เอาท์พุต

Root: ถ้าโหนดเป็นโหนดราก

Leaf: ถ้าโหนดเป็นโหนดลีฟ

Inner: ถ้าโหนดไม่ใช่โหนดรากหรือโหนดลีฟ

Not exist: หากไม่มีโหนด

ทั้งหมดของ `N` ไม่ซ้ำกัน จึงได้ค่าออกมาดังรูป

# Filter Primes from a List

Published by [Matt](#) in [Python](#) ▾

algorithms

arrays

numbers

Create a function that takes a list and returns a new list containing only prime numbers.

## Examples

```
filter_primes([7, 9, 3, 9, 10, 11, 27]) → [7, 3, 11]
```

```
filter_primes([10007, 1009, 1007, 27, 147, 77, 1001, 70]) → [10007, 1009]
```

```
filter_primes([1009, 10, 10, 10, 3, 33, 9, 4, 1, 61, 63, 69, 1087, 1091, 1093, 1097]) → [1009, 3, ...]
```

## Notes

- New list must maintain the order of primes as they first appear in the original list.
- Check the **Resources** tab for help.

### Filter Primes from a List

Create a function that takes a list and returns a new list containing only prime numbers. Examples `filter_primes([7, 9, 3, 9, 10, 11, 27]) → [7, 3, 11]` `filter_primes([10007, 1009, 1007, 27, 147, 77, 1001, 70]) → [10007, 1009]` `filter_primes([1009, 10, 10, 10, 3, 33, 9, 4, 1, 61, 63, 69, 1087, 1091, 1093, 1097]) → [1009, 3, ...]`

algorithms

arrays

numbers

Hard

```
#Filter Primes from a List
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if not n%i:
            return False
    return True

def filter_primes(num):
    return [i for i in num if is_prime(i)]

print(filter_primes([7, 9, 3, 9, 10, 11, 27]))
```

```
earth Admin/De
[7, 3, 11]
PS C:\Users\H
```

กรองราคาจากรายการ สร้างฟังก์ชันที่รับ  
รายการและส่งกลับรายการใหม่ที่มีเฉพาะ  
จำนวนเฉพาะ และรักษารายการแรกเอาไว้ ดัง  
รูปข้างๆ