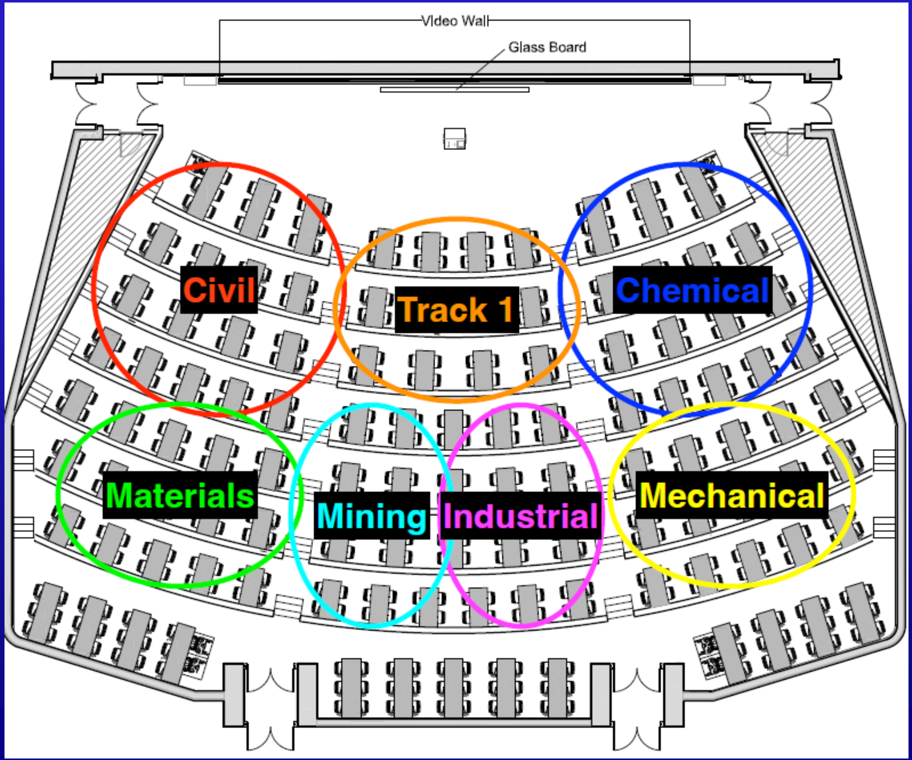
APS106

MIDTERM REVIEW


Video Wall
Glass Board
Civil
Track 1
Chemical
Materials
Mining
Industrial
Mechanical

RULES OF THE GAME

- ➢ Must use button on the table to answer question verbally and explain your answer – you will be called on to answer by one of the instructors
- ➢ Must tell us which discipline you're in before you answer for points (we'll rely on the honour system)
- ➢ If you get the answer right, you pick the next category
- ➢ If you get the answer wrong, the next person whose hand is up can steal
- ➢ Everyone is here to learn and review for the midterm so be kind to everyone who answers!

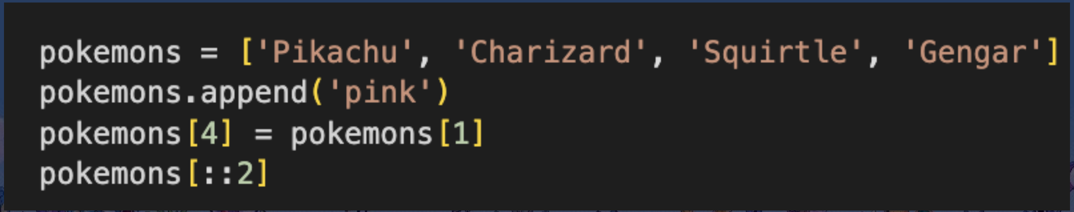
PANEL

POKÉMON GO: CATCH ‘EM ALL	SUPER MARIO: COIN RUSH!	MINECRAFT: MASTERING RESOURCES & TRADES	SUPER CONTRA: POWER-UP TACTICS
$250	$250	$250	$250
$500	$500	$500	$500
$1000	$1000	$1000	$1000

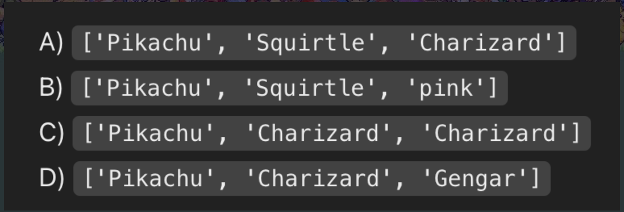
FINAL JEOPARDY

➢ REARRANGING POKÉMON STATS· $250

Which is the correct output?



```
pokemons = ['Pikachu', 'Charizard', 'Squirtle', 'Gengar']
pokemons.append('pink')
pokemons[4] = pokemons[1]
pokemons[::2]
```



A) `['Pikachu', 'Squirtle', 'Charizard']`

B) `['Pikachu', 'Squirtle', 'pink']`

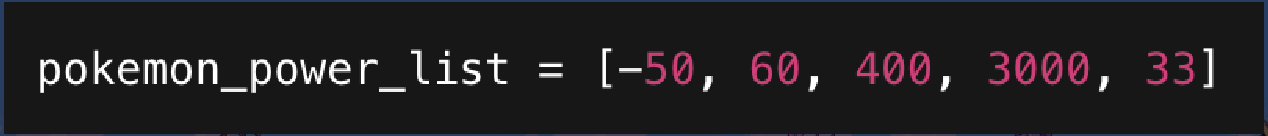
C) `['Pikachu', 'Charizard', 'Charizard']`

D) `['Pikachu', 'Charizard', 'Gengar']`

Question board

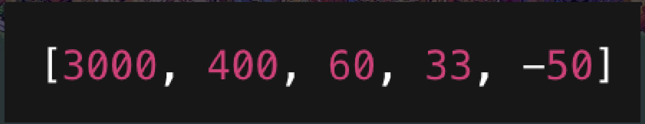
➢ POKÉMON POWER RANKINGS · $500

How can you transform the following list:



```
pokemon_power_list = [-50, 60, 400, 3000, 33]
```

into this output in the fewest steps??

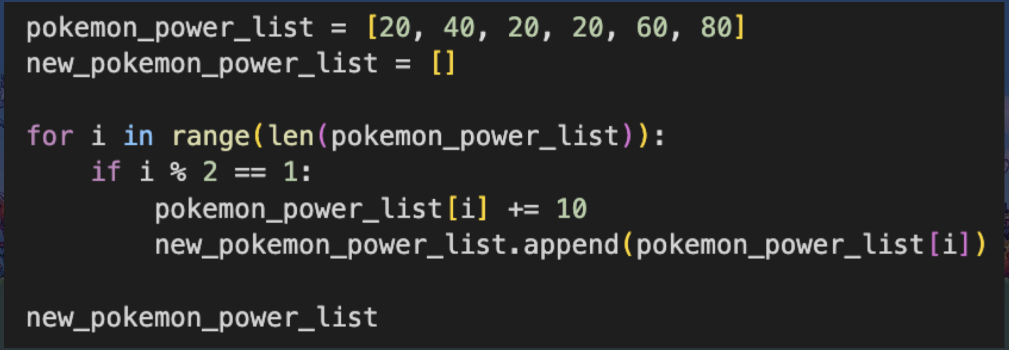


```
[3000, 400, 60, 33, -50]
```

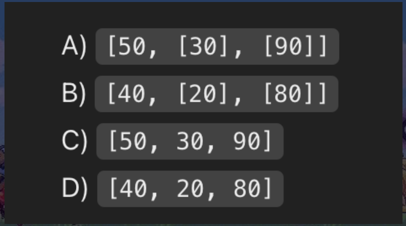
Question board

➢ POKÉMON TRAINING BOOST · $1000

Which is the correct output?



```
pokemon_power_list = [20, 40, 20, 20, 60, 80]
new_pokemon_power_list = []

for i in range(len(pokemon_power_list)):
    if i % 2 == 1:
        pokemon_power_list[i] += 10
        new_pokemon_power_list.append(pokemon_power_list[i])

new_pokemon_power_list
```



A) `[50, [30], [90]]`

B) `[40, [20], [80]]`

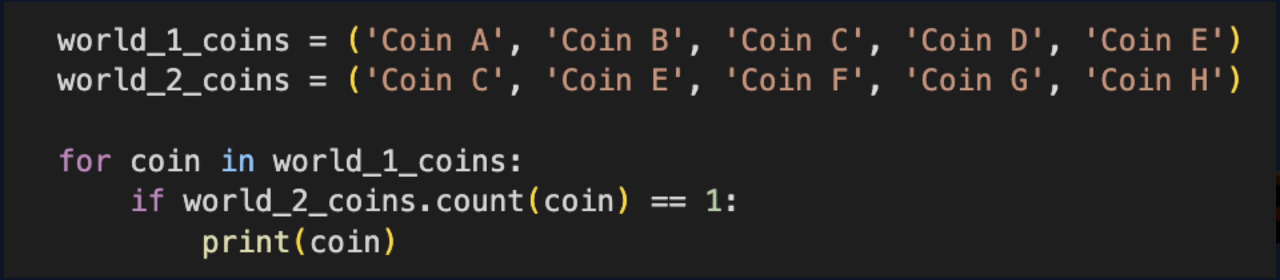
C) `[50, 30, 90]`

D) `[40, 20, 80]`

Question board

➢COLLECTING COINS · $250

What is the output?

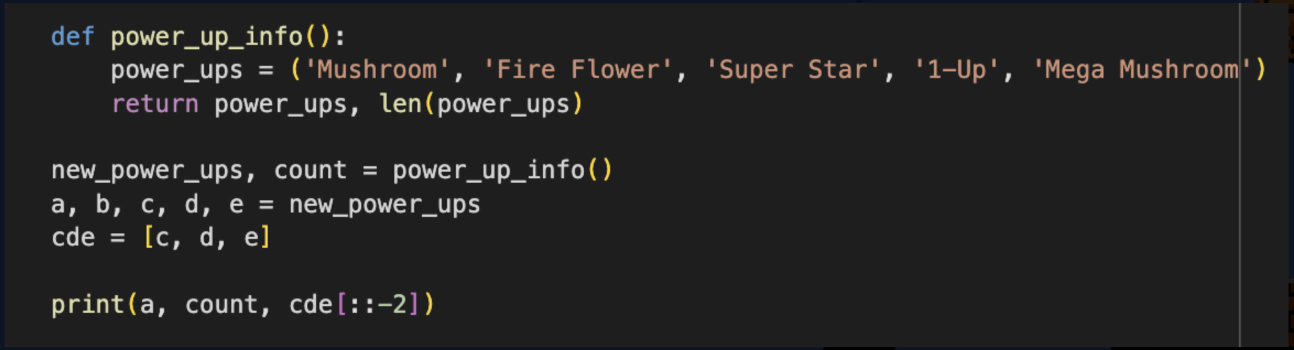


```
world_1_coins = ('Coin A', 'Coin B', 'Coin C', 'Coin D', 'Coin E')
world_2_coins = ('Coin C', 'Coin E', 'Coin F', 'Coin G', 'Coin H')

for coin in world_1_coins:
    if world_2_coins.count(coin) == 1:
        print(coin)
```

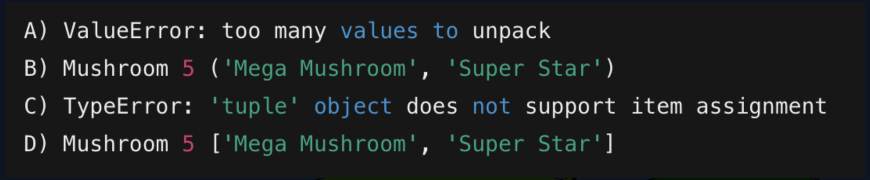
Question board

➢ POWER-UPS IN MARIO'S INVENTORY · $500

Which is the correct output?



```
def power_up_info():
    power_ups = ('Mushroom', 'Fire Flower', 'Super Star', '1-Up', 'Mega Mushroom')
    return power_ups, len(power_ups)

new_power_ups, count = power_up_info()
a, b, c, d, e = new_power_ups
cde = [c, d, e]

print(a, count, cde[::-2])
```

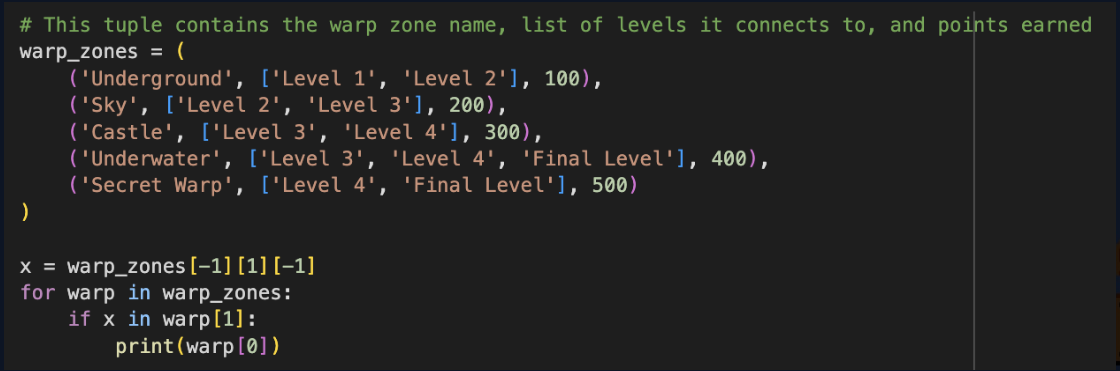


```
A) ValueError: too many values to unpack
B) Mushroom 5 ('Mega Mushroom', 'Super Star')
C) TypeError: 'tuple' object does not support item assignment
D) Mushroom 5 ['Mega Mushroom', 'Super Star']
```

Question board

➢ NAVIGATING THE WARP ZONES · $1000

Which is the correct output?



```
# This tuple contains the warp zone name, list of levels it connects to, and points earned
warp_zones = (
    ('Underground', ['Level 1', 'Level 2'], 100),
    ('Sky', ['Level 2', 'Level 3'], 200),
    ('Castle', ['Level 3', 'Level 4'], 300),
    ('Underwater', ['Level 3', 'Level 4', 'Final Level'], 400),
    ('Secret Warp', ['Level 4', 'Final Level'], 500)
)

x = warp_zones[-1][1][-1]
for warp in warp_zones:
    if x in warp[1]:
        print(warp[0])
```

```
A) Underwater
   Secret Warp
```

```
B) Level 3
   Level 4
   Final Level
```

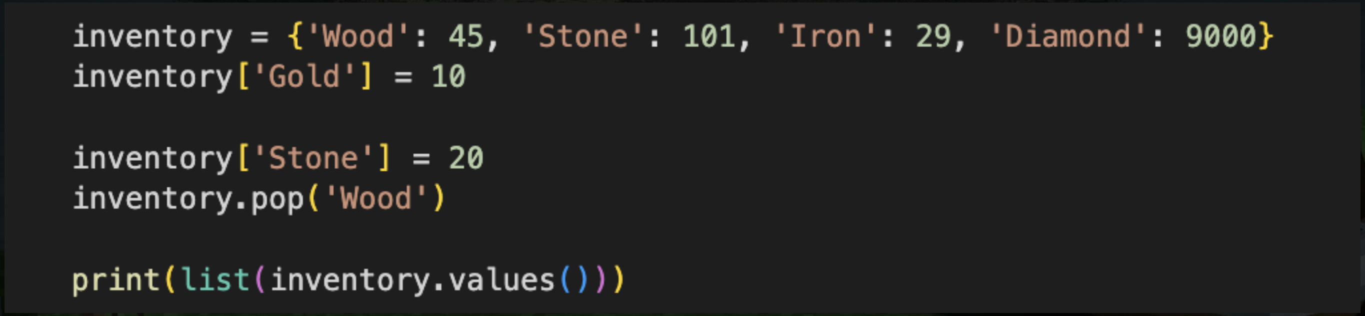
```
C) 100
   200
   300
```

```
D) ['Level 3', 'Level 4', 'Final Level']
   ['Level 4', 'Final Level']
```

Question board

➢ MANAGING YOUR MINECRAFT INVENTOR · $250

What is the Output?

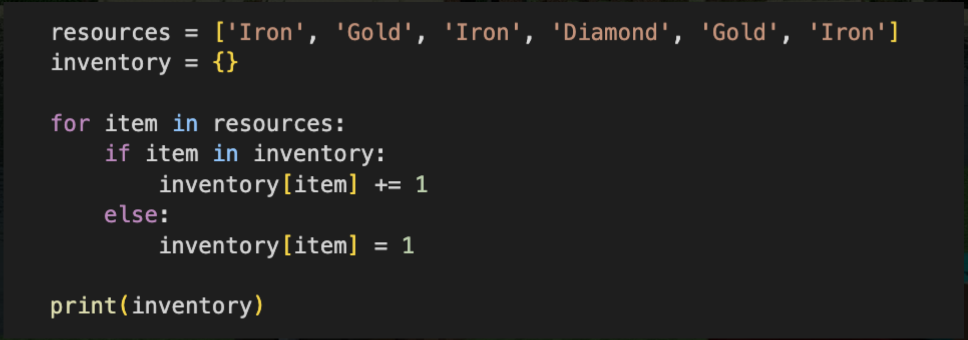


```
inventory = {'Wood': 45, 'Stone': 101, 'Iron': 29, 'Diamond': 9000}
inventory['Gold'] = 10

inventory['Stone'] = 20
inventory.pop('Wood')

print(list(inventory.values()))
```

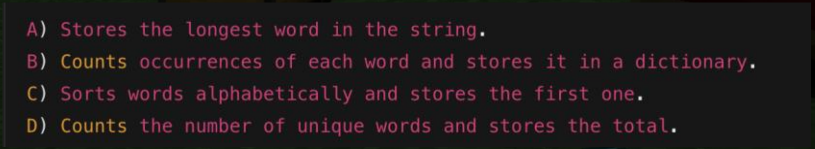
Question board

➢ RESOURCES IN YOUR CHEST · $500

What does the following code do?



```python
resources = ['Iron', 'Gold', 'Iron', 'Diamond', 'Gold', 'Iron']
inventory = {}

for item in resources:
    if item in inventory:
        inventory[item] += 1
    else:
        inventory[item] = 1

print(inventory)
```



A) Stores the longest word in the string.

B) Counts occurrences of each word and stores it in a dictionary.

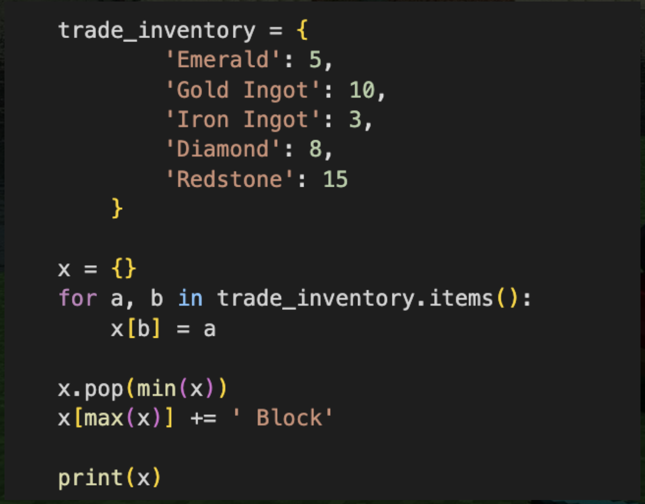
C) Sorts words alphabetically and stores the first one.

D) Counts the number of unique words and stores the total.

Question board

➢ TRADING WITH VILLAGERS · $1000

What is the output?

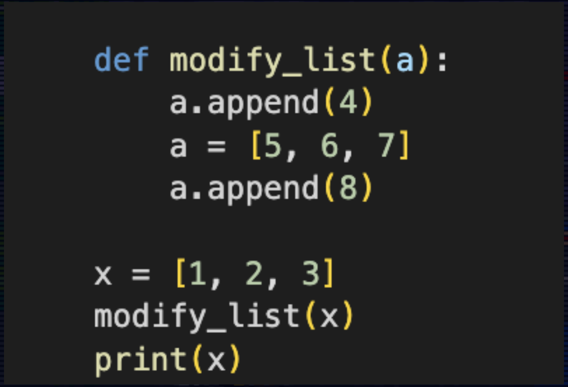


```
trade_inventory = {
        'Emerald': 5,
        'Gold Ingot': 10,
        'Iron Ingot': 3,
        'Diamond': 8,
        'Redstone': 15
    }

x = {}
for a, b in trade_inventory.items():
    x[b] = a

x.pop(min(x))
x[max(x)] += ' Block'

print(x)
```

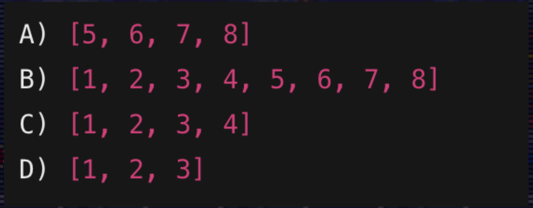
Question board

➢Upgrading Weapons· $250

Which one is the correct output?



```python
def modify_list(a):
    a.append(4)
    a = [5, 6, 7]
    a.append(8)

x = [1, 2, 3]
modify_list(x)
print(x)
```

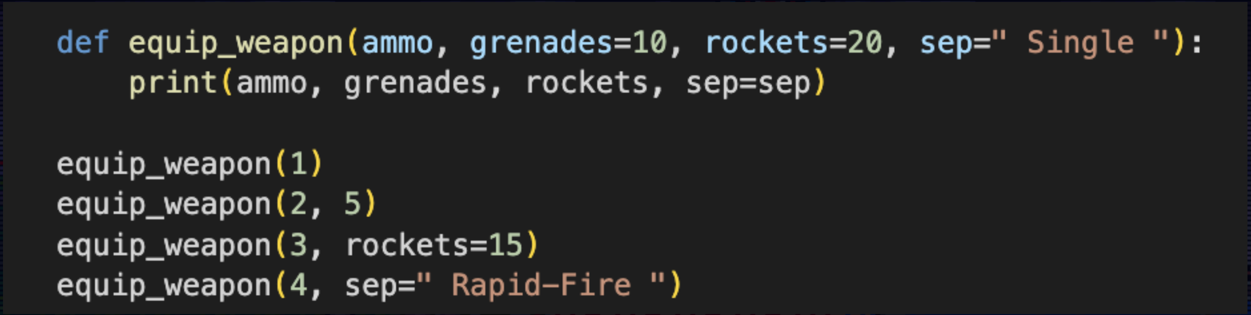


```
A) [5, 6, 7, 8]
B) [1, 2, 3, 4, 5, 6, 7, 8]
C) [1, 2, 3, 4]
D) [1, 2, 3]
```

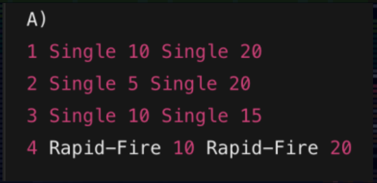
Question board

➢ WEAPON LOADOUT CONFIGURATION · $500

What is the Output?

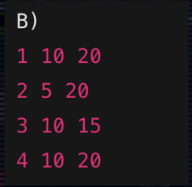


```python
def equip_weapon(ammo, grenades=10, rockets=20, sep=" Single "):
    print(ammo, grenades, rockets, sep=sep)

equip_weapon(1)
equip_weapon(2, 5)
equip_weapon(3, rockets=15)
equip_weapon(4, sep=" Rapid-Fire ")
```



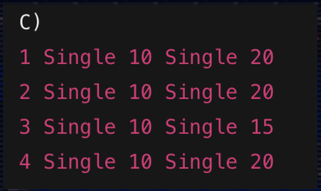
A)

```
1 Single 10 Single 20
2 Single 5 Single 20
3 Single 10 Single 15
4 Rapid-Fire 10 Rapid-Fire 20
```



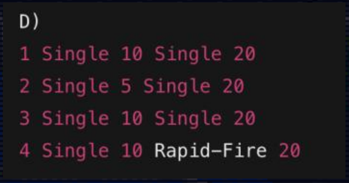
B)

```
1 10 20
2 5 20
3 10 15
4 10 20
```



C)

```
1 Single 10 Single 20
2 Single 10 Single 20
3 Single 10 Single 15
4 Single 10 Single 20
```



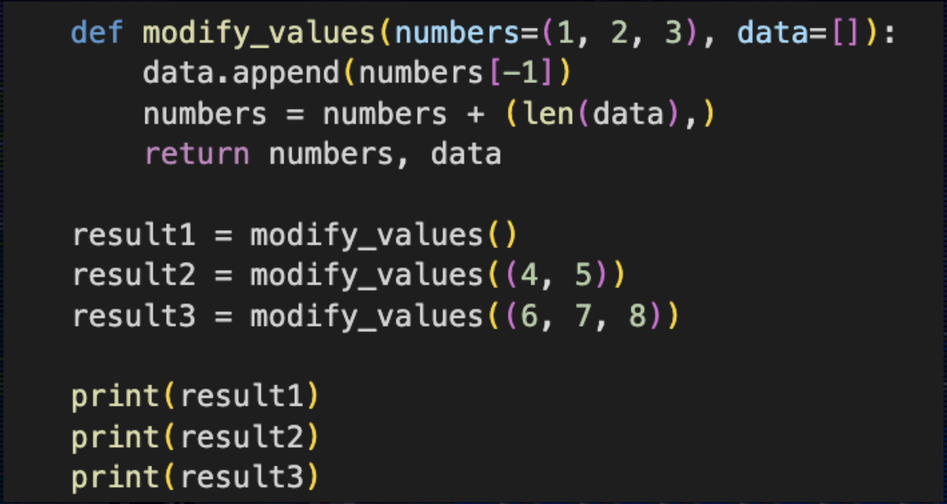
D)

```
1 Single 10 Single 20
2 Single 5 Single 20
3 Single 10 Single 20
4 Single 10 Rapid-Fire 20
```

Question board

➢ EXTRA LIVES & SCORE CALCULATION · $1000

What is the output?



```
def modify_values(numbers=(1, 2, 3), data=[]):
    data.append(numbers[-1])
    numbers = numbers + (len(data),)
    return numbers, data

result1 = modify_values()
result2 = modify_values((4, 5))
result3 = modify_values((6, 7, 8))

print(result1)
print(result2)
print(result3)
```

Question board

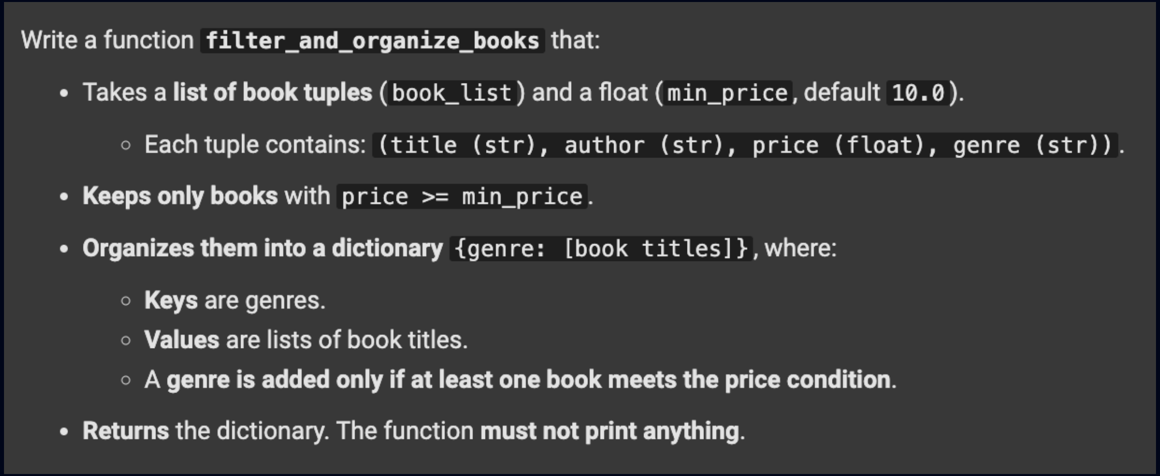
THE LIBRARY OF SURVIVAL

You are given access to a massive library, you must pass this Test to advance. Fail, and you’re eliminated!

Go to the Task

➢ TASK: FILTER AND ORGANIZE BOOKS BY GENRE

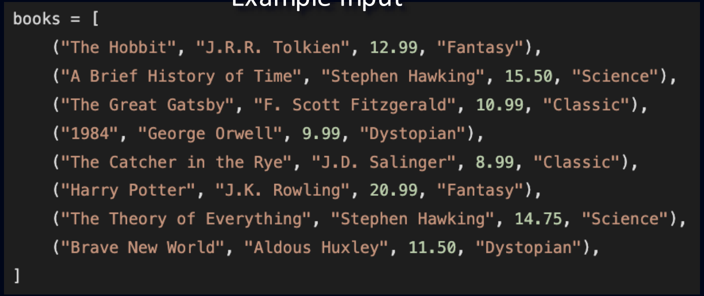
Function Description



Write a function `filter_and_organize_books` that:

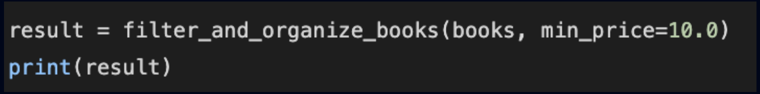
- Takes a **list of book tuples** (`book_list`) and a float (`min_price`, default `10.0`).
  - Each tuple contains: `(title (str), author (str), price (float), genre (str))`.
- **Keeps only books** with `price >= min_price`.
- **Organizes them into a dictionary** `{genre: [book titles]}`, where:
  - **Keys** are genres.
  - **Values** are lists of book titles.
  - A **genre is added only if at least one book meets the price condition.**
- **Returns** the dictionary. The function **must not print anything**.

Example Input



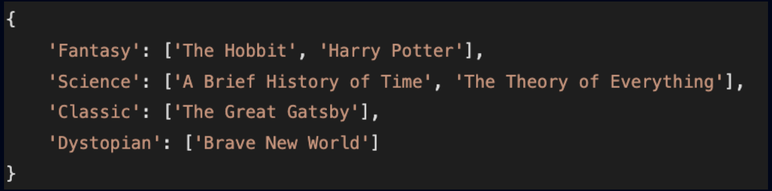
```
books = [
    ("The Hobbit", "J.R.R. Tolkien", 12.99, "Fantasy"),
    ("A Brief History of Time", "Stephen Hawking", 15.50, "Science"),
    ("The Great Gatsby", "F. Scott Fitzgerald", 10.99, "Classic"),
    ("1984", "George Orwell", 9.99, "Dystopian"),
    ("The Catcher in the Rye", "J.D. Salinger", 8.99, "Classic"),
    ("Harry Potter", "J.K. Rowling", 20.99, "Fantasy"),
    ("The Theory of Everything", "Stephen Hawking", 14.75, "Science"),
    ("Brave New World", "Aldous Huxley", 11.50, "Dystopian"),
]
```

Function Call



```
result = filter_and_organize_books(books, min_price=10.0)
print(result)
```

Expected output



```
{
    'Fantasy': ['The Hobbit', 'Harry Potter'],
    'Science': ['A Brief History of Time', 'The Theory of Everything'],
    'Classic': ['The Great Gatsby'],
    'Dystopian': ['Brave New World']
}
```

Question board