

# APS106

## MIDTERM REVIEW JEOPARDY

# RULES OF THE GAME

- Form a team with your table or tables around you (sizes ~2-4)
- Choose a team name for the scoreboard
- Go to <https://www.multibuzz.app>
- Enter the Room Code TQCWMG and your Team Name
- If you get the answer right, you gain the points, and pick the next category
- If you get the answer wrong, you lose the points, and the next person who buzzed in is up
- Ask questions if you're confused! Remember, if you're confused, so are others!

PANEL

RETURN OF THE JEDI	FILE AND FURIOUS	KEY TO THE SET	CLASSABLANCA	THE HITCHHIKER'S GUIDE TO PYTHON
\$250	\$250	\$250	\$250	\$250
\$500	\$500	\$500	\$500	\$500
\$750	\$750	\$750	\$750	\$750
\$1000	\$1000	\$1000	\$1000	\$1000

RETURN OF THE JEDI · \$250

# What is the result?

```
my_list = [1,0,6,6,0,1]
my_set = set(my_list)
print(len(my_set) + len(my_list) + max(my_list))
```

◀ BACK TO PANEL

Question board

## RETURN OF THE JEDI · \$250

```
my_list = [1,0,6,6,0,1]
my_set = set(my_list)
print(len(my_set) + len(my_list) + max(my_list))
```

(i) 15

◀ BACK TO PANEL

Question board

## RETURN OF THE JEDI · \$500

- d) Assuming  $x$  is an integer, what best describes what the following function, `decide`, does?

```
def decide(x):
    if x % 1 == 0:
        return 1
    if x % 2 == 0:
        return 2
    if x % 3 == 0:
        return 3
```

- (i) returns 1
- (ii) returns 1, 2, and 3
- (iii) returns  $x$  if  $x$  is equal to 1, 2, or 3
- (iv) returns  $x$  if  $x$  is divisible by 1, 2, or 3
- (v) returns 1, 2, or 3, whichever is the largest factor of  $x$

 BACK TO PANEL

Question board

## RETURN OF THE JEDI · \$500

- d) Assuming  $x$  is an integer, what best describes what the following function, `decide`, does?

```
def decide(x):
    if x % 1 == 0:
        return 1
    if x % 2 == 0:
        return 2
    if x % 3 == 0:
        return 3
```

- (i) returns 1
- (ii) returns 1, 2, and 3
- (iii) returns  $x$  if  $x$  is equal to 1, 2, or 3
- (iv) returns  $x$  if  $x$  is divisible by 1, 2, or 3
- (v) returns 1, 2, or 3, whichever is the largest factor of  $x$

 BACK TO PANEL

Question board

RETURN OF THE JEDI · \$750

# What is the output?

```
def fun1_iterative(n):
    while n >= 10:
        print(n)
        n //= 10
    print(n)

fun1_iterative(1000)
```

◀ BACK TO PANEL

[Question board](#)

RETURN OF THE JEDI · \$750

# What is the output?

```
def fun1_iterative(n):
    while n >= 10:
        print(n)
        n //= 10
    print(n)

fun1_iterative(1000)
```

1000  
100  
10  
1

◀ BACK TO PANEL

Question board

**RETURN OF THE JEDI · \$1000**

**What is the output?**

```
number_list = [40,60,80,100,120]

def calculator(s):
    " (list) -> list"
    divisor = len(s)
    new_list = [50, 70]
    i = 0

    while i<4:
        value = s[i] / divisor
        new_list.append(s[i])
        i +=1

    return new_list

print(calculator(number_list))
```

◀ BACK TO PANEL

Question board

**RETURN OF THE JEDI · \$1000**

**What is the output?**

```
number_list = [40,60,80,100,120]

def calculator(s):
    " (list) -> list"
    divisor = len(s)
    new_list = [50, 70]
    i = 0

    while i<4:
        value = s[i] / divisor
        new_list.append(s[i])
        i +=1

    return new_list

print(calculator(number_list))
```

[ 50, 70, 40, 60, 80, 100 ]

◀ BACK TO PANEL

Question board

FILE AND FURIOUS · \$250

What is the missing line of code to write  
the names into names.txt?

```
names = ['Tina', 'Sue', 'Brad']
myfile = open("names.txt", "w")
for item in names:
    ...
myfile.close()
```



1	Tina
2	Sue
3	Brad

BACK TO PANEL

Question board

**FILE AND FURIOUS · \$250**

**What is the missing line of code to write  
the names into names.txt?**

```
names = ['Tina', 'Sue', 'Brad']
myfile = open("names.txt", "w")
for item in names:
    myfile.write(item + '\n')
myfile.close()
```



1	Tina
2	Sue
3	Brad

 BACK TO PANEL

Question board

FILE AND FURIOUS · \$500

# What is the output?



jupyterhub grades.txt 2 minutes ago Logout Control Panel

File Edit View Language Plain Text

```
John B-
Susmit A-
Julia A
Eva C+
Lulu B+
```

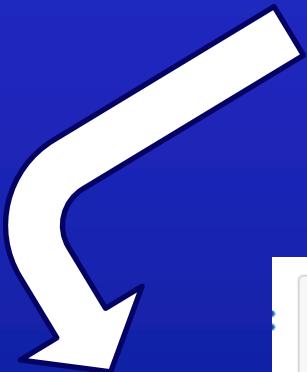
```
f = open("grades.txt", "r")
print(len(f.readlines()))
f.close()
```

BACK TO PANEL

Question board

FILE AND FURIOUS · \$500

# What is the output?



jupyterhub grades.txt 2 minutes ago Logout Control Panel

File Edit View Language Plain Text

```
John B-
Susmit A-
Julia A
Eva C+
Lulu B+
```

```
f = open("grades.txt", "r")
print(len(f.readlines()))
f.close()
```

5

BACK TO PANEL

Question board

FILE AND FURIOUS · \$750

# What is the output?

```
for i in range(5):
    if i % 2 == 0:
        file = open('my_file.txt', 'w')
        file.write('Seb')
    else:
        file = open('my_file.txt', 'a')
        file.write('Seb')
    file.close()

file = open('my_file.txt', 'r')
print(file.read())
```

◀ BACK TO PANEL

Question board

FILE AND FURIOUS · \$750

# What is the output?

```
for i in range(5):
    if i % 2 == 0:
        file = open('my_file.txt', 'w')
        file.write('Seb')
    else:
        file = open('my_file.txt', 'a')
        file.write('Seb')
    file.close()

file = open('my_file.txt', 'r')
print(file.read())
```

A. Seb

BACK TO PANEL

Question board

FILE AND FURIOUS · \$1000

# What is the output?

```
import csv

with open("people.csv", "w", newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["Name", "Age"])
    writer.writerow(["Alice", 30])
    writer.writerow(["Bob", 25])

with open("people.csv", "r") as file:
    reader = csv.reader(file)
    rows = list(reader)

print(rows[2][1])
```

⬅ BACK TO PANEL

[Question board](#)

FILE AND FURIOUS · \$1000

# What is the output?

```
import csv

with open("people.csv", "w", newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["Name", "Age"])
    writer.writerow(["Alice", 30])
    writer.writerow(["Bob", 25])

with open("people.csv", "r") as file:
    reader = csv.reader(file)
    rows = list(reader)

print(rows[2][1])
```

25

BACK TO PANEL

Question board

KEY TO THE SET · \$250

# What is the output?

```
x = {"apple", "banana", "carrot"}  
y = {"carrot", "eggplant", "fish"}  
z = {"fish", "grapes", "carrot"}  
  
result = y.intersection(z)  
  
print(result)
```

◀ BACK TO PANEL

Question board

KEY TO THE SET · \$250

# What is the output?

```
x = {"apple", "banana", "carrot"}  
y = {"carrot", "eggplant", "fish"}  
z = {"fish", "grapes", "carrot"}
```

```
result = y.intersection(z)
```

```
print(result)
```

```
{'fish', 'carrot'}
```

◀ BACK TO PANEL

[Question board](#)

## KEY TO THE SET · \$500

# What will this code output?

```
d= {'Running': 45, 'Hiking': 101, 'Biking': 29, 'Walking': 9000}
d['Swimming']=10

d['Hiking']=20
d.pop('Running')

d.values()
```

⬅ BACK TO PANEL

[Question board](#)

KEY TO THE SET · \$500

# What will this code output?

```
d= {'Running': 45, 'Hiking': 101, 'Biking': 29, 'Walking': 9000}
d['Swimming']=10

d['Hiking']=20
d.pop('Running')

d.values()
```

```
dict_values([20, 29, 9000, 10])
```

◀ BACK TO PANEL

[Question board](#)

KEY TO THE SET · \$750

# What will the code output?

```
phonebook = { 'Akako': '416-1212',
    'Ke': '647-2122',
    'Alya': '416-9923'}

print (phonebook[ 'Akako' ])
print( "Ke" in phonebook)
print (phonebook [ "Freya" ] )
```

◀ BACK TO PANEL

Question board

KEY TO THE SET · \$750

# What will the code output?

```
phonebook = { 'Akako': '416-1212',
    'Ke': '647-2122',
    'Alya': '416-9923'}

print (phonebook[ 'Akako' ])
print("Ke" in phonebook)
print (phonebook [ "Freya" ] )
```

```
416-1212
True
```

```
-----
KeyError                                     Traceback (most recent call last)
<ipython-input-10-9f149aae00fb> in <module>
      5     print (phonebook[ 'Akako' ])
      6     print("Ke" in phonebook)
----> 7     print (phonebook [ "Freya" ] )

KeyError: 'Freya'
```

⬅ BACK TO PANEL

[Question board](#)

KEY TO THE SET · \$1000

# What is the output?

```
message = "Hi!"
indices = [0, 2]

my_map = {}
for idx in indices:
    my_map[idx] = message[idx]

for idx in range(len(message)):
    print(idx, end=" ")
    if idx in my_map:
        print(my_map[idx])
    else:
        print("N/A")
```

◀ BACK TO PANEL

[Question board](#)

## KEY TO THE SET · \$1000

# What is the output?

```
message = "Hi!"
indices = [0, 2]

my_map = {}
for idx in indices:
    my_map[idx] = message[idx]

for idx in range(len(message)):
    print(idx, end=" ")
    if idx in my_map:
        print(my_map[idx])
    else:
        print("N/A")
```

0 H  
1 N/A  
2 !

⬅ BACK TO PANEL

[Question board](#)

CLASSABLANCA · \$250

# What is the output?

```
class Animal:
    def __init__(self, name):
        self.weight = 16
        self.name = name
        self.age = 10

    def __str__(self):
        return self.name + " meows"

    def speak(self):
        print(self.name + " says hi", end = ' ')
        return 'skers'

cat = Animal("Whiskers")
print(cat.speak())
```

◀ BACK TO PANEL

[Question board](#)

CLASSABLANCA · \$250

# What is the output?

```
class Animal:
    def __init__(self, name):
        self.weight = 16
        self.name = name
        self.age = 10

    def __str__(self):
        return self.name + " meows"

    def speak(self):
        print(self.name + " says hi", end = ' ')
        return 'skers'

cat = Animal("Whiskers")
print(cat.speak())
```

Whiskers says hiskers

◀ BACK TO PANEL

[Question board](#)

CLASSABLANCA · \$500

# What is the output?

```
class CityCounter:
    def __init__(self, people):
        self.people = people

    def count_city(self, city_name):
        count = 0
        for person in self.people:
            if person[2].lower() == city_name.lower():
                count += 1
        return count

# Example data and usage
people = [
    ("Alice", "Female", "London"),
    ("Bob", "Male", "Paris"),
    ("Charlie", "Male", "London")
]

counter = CityCounter(people)
print("People from London:", counter.count_city("London"))
```

◀ BACK TO PANEL

[Question board](#)

CLASSABLANCA · \$500

# What is the output?

```
: class CityCounter:
    def __init__(self, people):
        self.people = people

    def count_city(self, city_name):
        count = 0
        for person in self.people:
            if person[2].lower() == city_name.lower():
                count += 1
        return count

# Example data and usage
people = [
    ("Alice", "Female", "London"),
    ("Bob", "Male", "Paris"),
    ("Charlie", "Male", "London")
]

counter = CityCounter(people)
print("People from London:", counter.count_city("London"))

People from London: 2
```

BACK TO PANEL

Question board

CLASSABLANCA \$750

# What is the output?

```
class Journey:
    def __init__(self, origin, destination, duration):
        self.origin = origin
        self.destination = destination
        self.duration = duration

journeys = [Journey('X', 'Y', 2.5), Journey('Y', 'Z', 1.8)]

times = []
for journey in journeys:
    times.append(journey.duration)
times.sort()
times.reverse()

print(times[0], times[1], sep = '!', end = '!')|
```

◀ BACK TO PANEL

[Question board](#)

CLASSABLANCA \$750

# What is the output?

```
class Journey:
    def __init__(self, origin, destination, duration):
        self.origin = origin
        self.destination = destination
        self.duration = duration

journeys = [Journey('X', 'Y', 2.5), Journey('Y', 'Z', 1.8)]

times = []
for journey in journeys:
    times.append(journey.duration)
times.sort()
times.reverse()

print(times[0], times[1], sep = '!', end = '!!')
2.5!1.8!!
```

◀ BACK TO PANEL

[Question board](#)

CLASSABLANCA · \$1000

# What is the output?

```
class Engine:
    def __init__(self, type):
        self.type = type

class Wheels:
    def __init__(self, count):
        self.count = count

class Vehicle:
    def __init__(self, engine_type, wheel_count):
        self.engine = Engine(engine_type)
        self.wheels = Wheels(wheel_count)

    def vehicle_info(self):
        return "Engine:" + self.engine.type + "Wheels:" + self.wheels.count

vehicle = Vehicle("V8", 4)
print(vehicle.vehicle_info())
```

◀ BACK TO PANEL

[Question board](#)

# CLASSABLANCA · \$1000

## What is the output?

```
class Engine:
    def __init__(self, type):
        self.type = type

class Wheels:
    def __init__(self, count):
        self.count = count

class Vehicle:
    def __init__(self, engine_type, wheel_count):
        self.engine = Engine(engine_type)
        self.wheels = Wheels(wheel_count)

    def vehicle_info(self):
        return "Engine:" + self.engine.type + "Wheels:" + self.wheels.count

vehicle = Vehicle("V8", 4)
print(vehicle.vehicle_info())
```

```
-----  
TypeError                                     Traceback (most recent call last)
Cell In[30], line 18
      15     return "Engine:" + self.engine.type + "Wheels:" + self.wheels.count
      17 vehicle = Vehicle("V8", 4)
--> 18 print(vehicle.vehicle_info())

Cell In[30], line 15, in Vehicle.vehicle_info(self)
      14     def vehicle_info(self):
--> 15         return "Engine:" + self.engine.type + "Wheels:" + self.wheels.count

TypeError: can only concatenate str (not "int") to str
```

[Question board](#)

THE HITCHHIKER'S GUIDE TO PYTHON · \$250

# What is the output?

```
andy = 'randy'  
andy[0] = 'r'  
print(andy)
```

 BACK TO PANEL

Question board

THE HITCHHIKER'S GUIDE TO PYTHON · \$250

# What is the output?

```
andy = 'randy'  
andy[0] = 'r'  
print(andy)
```

```
TypeError  
<ipython-input-92-b75e9d9a69f8> in <module>  
  1 #everything aps106 $250  
  2 andy = 'randy'  
----> 3 andy[0] = 'r'  
  4 print(andy)  
  
TypeError: 'str' object does not support item assignment
```

 BACK TO PANEL

Question board

THE HITCHHIKER'S GUIDE TO PYTHON · \$500

# What is the output?

```
a = [1, 2, 3]
b = a
a.append(4)
b.append(5)
print(b)
```

 BACK TO PANEL

Question board

THE HITCHHIKER'S GUIDE TO PYTHON · \$500

# What is the output?

```
a = [1, 2, 3]
b = a
a.append(4)
b.append(5)
print(b)
```

[1, 2, 3, 4, 5]

◀ BACK TO PANEL

Question board

THE HITCHHIKER'S GUIDE TO PYTHON · \$750

# What is the output?

```
s = "banana"  
print(s.replace('a', 'o').upper().count('O') * s.find('n'))
```

◀ BACK TO PANEL

Question board

THE HITCHHIKER'S GUIDE TO PYTHON · \$750

# What is the output?

```
s = "banana"  
print(s.replace('a', 'o').upper().count('O') * s.find('n'))
```

6

◀ BACK TO PANEL

Question board

# THE HITCHHIKER'S GUIDE TO PYTHON · \$1000

```
x = 'a,bb,ccc'.upper().lower().split(',')
x.append('dddd')
x.sort()
x.reverse()
x.pop()
print(x)
```

 BACK TO PANEL

[Question board](#)

# THE HITCHHIKER'S GUIDE TO PYTHON · \$1000

```
x = 'a,bb,ccc'.upper().lower().split(',')
x.append('dddd')
x.sort()
x.reverse()
x.pop()
print(x)
```

- A. [ 'ddd', 'ccc', 'bb' ]

◀ BACK TO PANEL

[Question board](#)

# FINAL JEOPARDY

## QUESTION 1: FINAL JEOPARDY

You are given a dictionary where each Marvel character maps to a movie they appeared in, but there are multiple characters that appear in the same movie. Your task is to invert the dictionary such that each movie maps to a list of characters that appeared in that movie.

Write a Python function called `invert_marvel_dictionary` that takes a dictionary as input and returns a new dictionary where the keys are the unique movies from the original dictionary and the values are lists of characters from the original dictionary that appeared in those movies.

Example:

Given the dictionary:

```
marvel_dict = {  
    "Iron Man": "Iron Man",  
    "Captain America": "The First Avenger",  
    "Black Widow": "Avengers",  
    "Thor": "Thor",  
    "Hulk": "Avengers",  
    "Hawkeye": "Avengers"  
}
```

The function should return:

```
{  
    "Iron Man": ["Iron Man"],  
    "The First Avenger": ["Captain America"],  
    "Avengers": ["Black Widow", "Hulk", "Hawkeye"],  
    "Thor": ["Thor"]  
}
```

BACK TO MAIN

## QUESTION 1: FINAL JEOPARDY

You are given a dictionary where each Marvel character maps to a movie they appeared in, but there are multiple characters that appear in the same movie. Your task is to invert the dictionary such that each movie maps to a list of characters that appeared in that movie.

Write a Python function called `invert_marvel_dictionary` that takes a dictionary as input and returns a new dictionary where the keys are the unique movies from the original dictionary and the values are lists of characters from the original dictionary that appeared in those movies.

Example:

Given the dictionary:

```
marvel_dict = {
    "Iron Man": "Iron Man",
    "Captain America": "The First Avenger",
    "Black Widow": "Avengers",
    "Thor": "Thor",
    "Hulk": "Avengers",
    "Hawkeye": "Avengers"
}
```

The function should return:

```
{
    "Iron Man": ["Iron Man"],
    "The First Avenger": ["Captain America"],
    "Avengers": ["Black Widow", "Hulk", "Hawkeye"],
    "Thor": ["Thor"]
}
```

```
def invert_marvel_dictionary(marvel_dict):
    """
    dict -> dict
    """
    inverted_dict = {}
    for character, movie in marvel_dict.items():
        if movie not in inverted_dict:
            inverted_dict[movie] = []
        inverted_dict[movie].append(character)
    return inverted_dict
```

BACK TO PANEL

## QUESTION 2 · FINAL JEOPARDY

### Question 4 [10 marks] – Write the Code

For this question, you will write a program that can translate an English sentence to Spanish and a Spanish sentence to English.

#### PART A) [2 Marks]

Complete the function get\_user\_input. You can assume user input is correct.

```
def get_user_input():
    """
    () -> str, str
    Prompts the user for sentence and for its language: "eng"
    or "span".
    Returns the input sentence and the language of the sentence
    in this order.
    """
    # Write your code here.
```

#### PART B) [3 Marks]

Complete the function remove\_punctuation.

```
def remove_punctuation(sentence):
    """
    (str) -> str
    Takes a sentence as input and returns the sentence with all
    punctuations replaced by an empty string ("").
    Example:
    >>> print(remove_punctuation('Hello, how are you?'))
    Hello how are you
    """

    punctuations = "!=;,:.?!" # punctuation marks to be removed
    # Write your code here.
```

## QUESTION 2 · FINAL JEOPARDY

### Question 4 [10 marks] – Write the Code

For this question, you will write a program that can translate an English sentence to Spanish and a Spanish sentence to English.

#### PART A) [2 Marks]

Complete the function get user input. You can assume user input is correct.

```
def get_user_input():
    """
    () -> str, str
    Prompts the user for sentence and for its language: "eng"
    or "span".

    Returns the input sentence and the language of the sentence
    in this order.
    """
    # Write your code here.

    sentence = input("What sentence would you like to
translate?")
    language = input("Is this sentence Spanish or English")
    return sentence, language
```

#### PART B) [3 Marks]

Complete the function remove punctuation.

```
def remove_punctuation(sentence):
    """
    (str) -> str
    Takes a sentence as input and returns the sentence with all
    punctuations replaced by an empty string ("").

    Example:
    >>> print(remove_punctuation('Hello, how are you?'))
    Hello how are you
    """

    punctuations = "!=;,:.?!" # punctuation marks to be removed

    # Write your code here.
    formatted = str(sentence)
    for char in sentence:
        if char in punctuations:
            formatted = formatted.replace(char, '')
    return formatted
```

 BACK TO PANEL

## QUESTION 2 · FINAL JEOPARDY

### PART C) [5 Marks]

Complete the function translate.

```
def translate(sentence, language):
    """
    (str, str) -> str
    Takes as input a sentence to translate and the language of
    that sentence ("eng" or "span").
    Returns the sentence translated into the other language.
    If the sentence is English sentence, the function must
    return it in Spanish and vice versa.
    The dictionaries span_to_eng and eng_to_span contain word
    mappings between the languages for all words in the English
    and Spanish dictionaries.
    A sentence may contain punctuation and capital letters but
    the dictionaries and the translated sentence only contain
    lower-case letters.
    Example:
    >>> print(translate('I speak English, my friends!!!!', 'eng'))
    yo hablo ingles mis amigos
    """

    span_to_eng = {'yo': 'i', 'hablo': 'speak', 'espanol': 'spanish', ...}
    eng_to_span = {'i': 'yo', 'speak': 'hablo', 'spanish': 'espanol', ...}

    # Write your code here.
```

 BACK TO PANEL

## QUESTION 2 · FINAL JEOPARDY

### PART C) [5 Marks]

Complete the function translate.

```
def translate(sentence, language):
    """
    (str, str) -> str
    Takes as input a sentence to translate and the language of
    that sentence ("eng" or "span").
    Returns the sentence translated into the other language.
    If the sentence is English sentence, the function must
    return it in Spanish and vice versa.
    The dictionaries span_to_eng and eng_to_span contain word
    mappings between the languages for all words in the English
    and Spanish dictionaries.
    A sentence may contain punctuation and capital letters but
    the dictionaries and the translated sentence only contain
    lower-case letters.
    Example:
    >>> print(translate('I speak English, my friends!!!!', 'eng'))
    yo hablo ingles mis amigos
    """

    span_to_eng = {'yo': 'i', 'hablo': 'speak', 'espanol': 'spanish', ...}
    eng_to_span = {'i': 'yo', 'speak': 'hablo', 'spanish': 'espanol', ...}

    # Write your code here.
    sentence = remove_punctuation(sentence)
    words = sentence.lower().split(' ')
    translated = ''

    for word in words:
        if language == 'eng':
            translated += eng_to_span[word] + ' '
        else:
            translated += span_to_eng[word] + ' '

    return translated
```

DO PANEL