**Task-1**

# Task 1 start

print("Task 1 Display Menu")

def display\_menu():

print("Menu:")

print("a. Print help menu")

print("b. Exit the program")

display\_menu()

# Task 1 End

**Task-2**

# Task 2 Start

print("Task 2 User Input")

def display\_menu():

print("Menu:")

print("a. Print help menu")

print("b. Exit the program")

def main():

display\_menu()

while True:

user\_input = input("wildlife> ")

if user\_input == 'help':

display\_menu()

elif user\_input == 'exit':

print("Exiting the program.")

return

else:

print("Invalid command. Please enter 'help' or 'exit'.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Task 2 End #

**Task-3**

# Task 3 Start #

def display\_menu():

print("Menu:")

print("a. Print help menu")

print("b. Exit the program")

print("c. Display animal species in a city")

def search\_species(city):

return [

{"Species": {"TaxonID":"2336","AcceptedCommonName": "dolphin", "PestStatus": "Nil"}},

{"Species": {"TaxonID":"655","AcceptedCommonName": "snake", "PestStatus": "Venomous"}}

]

def display\_species(species\_list):

print("Species found:")

for species in species\_list:

name = species["Species"]["AcceptedCommonName"]

status = species["Species"]["PestStatus"]

Tax = species["Species"]["TaxonID"]

print(f"TaxonID: {Tax} , Name: {name}, Pest Status: {status}")

def main():

display\_menu()

while True:

user\_input = input("wildlife> ")

if user\_input == 'help':

display\_menu()

elif user\_input == 'exit':

print("Exiting the program.")

return

elif user\_input.startswith('species'):

city = user\_input.split(maxsplit=1)[1].strip()

species\_list = search\_species(city)

display\_species(species\_list)

else:

print("Invalid command. Please enter 'help', 'exit', or 'species <city>'.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Task 3 End #

# To run your code you want to type this in command

# species melbourne #

**Task-4**

# Task 4 Start #

def display\_menu():

print("Menu:")

print("a. Print help menu")

print("b. Exit the program")

print("c. Display animal species in a city")

print("d. Display animal sightings in a city")

def search\_species(city):

return [

{"Species": {"TaxonID":"1036","AcceptedCommonName": "dolphin", "PestStatus": "Nil"}},

{"Species": {"TaxonID":"236","AcceptedCommonName": "snake", "PestStatus": "Venomous"}}

]

def display\_species(species\_list):

print("Species found:")

for species in species\_list:

name = species["Species"]["AcceptedCommonName"]

status = species["Species"]["PestStatus"]

Tax = species["Species"]["TaxonID"]

print(f"TaxonID: {Tax} , Name: {name}, Pest Status: {status}")

def search\_sightings(taxonid, city):

return [

{"properties": {"Taxonid":"1020","StartDate": "2003-06-22", "LocalityDetails": "Kew","SiteCode":"Incidental"}},

{"properties": {"Taxonid":"2120","StartDate": "1996-09-21", "LocalityDetails": "Murrumbeena","SiteCode":"Incidental"}},

{"properties": {"Taxonid":"630","StartDate": "1999-11-15", "LocalityDetails": "Tinaroo","SiteCode":"Incidental"}}

]

def display\_sightings(sightings):

print("Animal sightings:")

for sighting in sightings:

taxonid = sighting["properties"]["Taxonid"]

start\_date = sighting["properties"]["StartDate"]

locality = sighting["properties"]["LocalityDetails"]

sitecode = sighting["properties"]["SiteCode"]

print(f"taxonid: {taxonid},Start Date: {start\_date}, Locality: {locality},sitecode: {sitecode}")

def main():

display\_menu() # Display the initial help menu

while True:

user\_input = input("wildlife> ")

if user\_input == 'help':

display\_menu()

elif user\_input == 'exit':

print("Exiting the program.")

return

elif user\_input.startswith('species'):

city = user\_input.split(maxsplit=1)[1].strip()

species\_list = search\_species(city)

display\_species(species\_list)

elif user\_input.startswith('sightings'):

\_, species, city = user\_input.split(maxsplit=2)

sightings = search\_sightings(species, city)

display\_sightings(sightings)

else:

print("Invalid command. Please enter 'help', 'exit', 'species <city>', or 'sightings <species>, <city>'.")

# Debugging and testing the main() function

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Task 4 End #

# # To run your code you want to type this in command

# sighthins species melbourne #

**Task-5**

# Task 5 Start #

def display\_menu():

"""

Display the menu options for the wildlife sighting program.

"""

print("Menu:")

print("a. Print help menu")

print("b. Exit the program")

print("c. Display animal species in a city")

print("d. Display animal sightings in a city")

print("e. Display venomous species")

def search\_species(city):

# Stub implementation, will be replaced with actual functionality

return [

{"Species": {"TaxonID":"1023","AcceptedCommonName": "dolphin", "PestStatus": "Nil"}},

{"Species": {"TaxonID":"562","AcceptedCommonName": "snake", "PestStatus": "Venomous"}}

]

def display\_species(species\_list):

print("Species found:")

for species in species\_list:

name = species["Species"]["AcceptedCommonName"]

status = species["Species"]["PestStatus"]

Tax = species["Species"]["TaxonID"]

print(f"TaxonID: {Tax} , Name: {name}, Pest Status: {status}")

def search\_sightings(taxonid, city):

# Stub implementation, will be replaced with actual functionality

return [

{"properties": {"StartDate": "1998-11-15", "LocalityDetails": "Tinaroo"}}

]

def display\_sightings(sightings):

print("Animal sightings:")

for sighting in sightings:

start\_date = sighting["properties"]["StartDate"]

locality = sighting["properties"]["LocalityDetails"]

print(f"Start Date: {start\_date}, Locality: {locality}")

def filter\_venomous(species\_list):

return [species for species in species\_list if species["Species"]["PestStatus"] == "Venomous"]

def main():

display\_menu() # Display the initial help menu

while True:

user\_input = input("wildlife> ").strip().lower()

if user\_input == 'help':

display\_menu()

elif user\_input == 'exit':

print("Exiting the program.")

return

elif user\_input.startswith('species'):

if 'venomous' in user\_input:

city = user\_input.split(maxsplit=1)[1].split()[0]

species\_list = search\_species(city)

venomous\_species = filter\_venomous(species\_list)

display\_species(venomous\_species)

else:

city = user\_input.split(maxsplit=1)[1].strip()

species\_list = search\_species(city)

display\_species(species\_list)

elif user\_input.startswith('sightings'):

\_, species, city = user\_input.split(maxsplit=2)

sightings = search\_sightings(species, city)

display\_sightings(sightings)

elif user\_input == 'venomous':

city = input("Enter city: ").strip()

species\_list = search\_species(city)

venomous\_species = filter\_venomous(species\_list)

display\_species(venomous\_species)

else:

print("Invalid command. Please enter 'help', 'exit', 'species <city>', 'sightings <species>, <city>', or 'venomous'.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Task 5 End #

# To run this code you right this command

#wildlife> venomous

#Enter city: melbourne

#Species found:

#TaxonID: 236 , Name: snake, Pest Status: Venomous

**Task-6**

def gps(city):

# Stub implementation, always returning Brisbane's coordinates

brisbane\_coordinates = { "latitude": -27.4689682, "longitude": 153.0234991 }

return brisbane\_coordinates

def search\_species(city):

# Stub im

plementation, returning a list of species dictionary

species\_list = [

{"name": "Koala", "population": 5000},

{"name": "Kangaroo", "population": 3000},

{"name": "Platypus", "population": 100},

{"name": "Shark", "population": 500},

{"name": "Dolphine", "population": 3500},

{"name": "Cat", "population": 1200},

]

print (" Here the Brisbane all species:---------- \n",species\_list)

# Assert statements to check if Brisbane's coordinates are returned correctly

brisbane\_coordinates = gps("Brisbane")

assert brisbane\_coordinates == { "latitude": -27.4689682, "longitude": 153.0234991 }

# Test the search\_species function

city\_species = search\_species("Brisbane")

print(city\_species) # Output will be a list of species dictionaries