Question 1

In [1]:
```
a = 0
def b():
 global a
 a = c(a)
def c(a):
 return a + 2
```

In [2]:
```
b()
b()
b()
a
```

Out[2]:
```
6
```

Here, we start with a number called a, which is 0.

Then, we have a function called b() that adds 2 to a.

We use b() three times in a row.

Now, if we look at a, it's become 6 because each time we used b(), it added 2 to a.

Question 2

In [3]:
```
def fileLength(filename):
    try:
        infile = open(filename)
        content = infile.read()
        infile.close()
        return len(content)
    except FileNotFoundError:
        print(f"Oops! The file '{filename}' seems to be missing. Please che
```

In [4]:
```
fileLength('Filelength.txt')
```

Out[4]:
```
123
```

In [5]:
```
fileLength('idterm.py')
```

```
Oops! The file 'idterm.py' seems to be missing. Please check the file nam
e and try again.
```

Question 3

SUB QUESTION 1

In [6]:
```
class Marsupial:
    def __init__(self):
        self.pouch = []

    def carry(self, item):
        self.pouch.append(item)

    def contents(self):
        return self.pouch
```

In [7]:
```
k = Marsupial()
k.carry('doll')
k.carry('firetruck')
k.carry('kitten')
```

In [8]:
```
print(k.contents())
```

```
['doll', 'firetruck', 'kitten']
```

SUB QUESTION 2

In [9]:
```
class Marsupial:
    def __init__(self):
        self.pouch = []

    def carry(self, item):
        self.pouch.append(item)

    def contents(self):
        return self.pouch

class Kangaroo(Marsupial):
    def __init__(self, x=0, y=0):
        super().__init__()
        self.x = x
        self.y = y

    def jump(self, dx, dy):
        self.x += dx
        self.y += dy

    def __str__(self):
        return f"I am a Kangaroo located at coordinates ({self.x},{self.y}
```

In [10]:
```
k = Kangaroo()
k.carry('doll')
k.carry('firetruck')
k.carry('kitten')
```

In [11]:
```
print(k.contents())
```

```
['doll', 'firetruck', 'kitten']
```

In [12]:
```
k.jump(1, 0)
k.jump(1, 0)
k.jump(1, 0)
```

In [13]:
```
print(k)
```

```
I am a Kangaroo located at coordinates (3,0)
```

Question 4

In [14]:
```
def collatz(x):
    print(x, end=' ')
    if x == 1:
        return
    elif x % 2 == 0:
        collatz(x // 2)
    else:
        collatz(3 * x + 1)
```

In [15]:
```
collatz(1)
```

```
1
```

In [16]:
```
collatz(10)
```

```
10 5 16 8 4 2 1
```

Question 5

In [17]:
```python
def binary(n):
    if n == 0:
        print(0, end='')
    elif n == 1:
        print(1, end='')
    else:
        binary(n // 2)
        print(n % 2, end='')
```

In [18]:
```python
binary(0)
print()
binary(1)
print()
binary(3)
print()
binary(9)
```

```
0
1
11
1001
```

Question 6

In [19]:
```python
from html.parser import HTMLParser

class HeadingParser(HTMLParser):
    def __init__(self):
        super().__init__()
        self.indentation = 0
        self.in_heading = False

    def handle_starttag(self, tag, attrs):
        if tag.startswith('h') and tag[1:].isdigit():
            self.indentation = int(tag[1:]) - 1
            self.in_heading = True

    def handle_endtag(self, tag):
        if tag.startswith('h'):
            self.indentation = 0
            self.in_heading = False

    def handle_data(self, data):
        if self.in_heading:
            print(' ' * self.indentation + data.strip())
```

In [20]:
```python
infile = open('w3c.html')
content = infile.read()
infile.close()
```

In [21]:
```
hp = HeadingParser()
hp.feed(content)
```

```
W3C Mission
 Principles
```

Question 7

In [22]:
```
import requests
from bs4 import BeautifulSoup

def webdir(url, depth, indent=0):
    if depth < 0:
        return

    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    print(' ' * indent + url)

    if depth == 0:
        return

    links = soup.find_all('a', href=True)
    for link in links:
        next_url = link['href']
        if next_url.startswith('http'):  # Ensure it's an absolute URL
            webdir(next_url, depth - 1, indent + 1)
```

In [23]:
```
webdir('http://reed.cs.depaul.edu/lperkovic/csc242/test1.html', 2, 0)
```

```
http://reed.cs.depaul.edu/lperkovic/csc242/test1.html (http://reed.cs.dep
aul.edu/lperkovic/csc242/test1.html)
```

Question 8

In [24]:
```
import pandas as pd
import sqlite3
```

```
In [25]: !pip install ipython-sql
```

```
Requirement already satisfied: ipython-sql in c:\users\kashi\appdata\loca
l\programs\python\python311\lib\site-packages (0.5.0)
Requirement already satisfied: prettytable in c:\users\kashi\appdata\loca
l\programs\python\python311\lib\site-packages (from ipython-sql) (3.10.0)
Requirement already satisfied: ipython in c:\users\kashi\appdata\local\pr
ograms\python\python311\lib\site-packages (from ipython-sql) (8.8.0)
Requirement already satisfied: sqlalchemy>=2.0 in c:\users\kashi\appdata
\local\programs\python\python311\lib\site-packages (from ipython-sql) (2.
0.27)
Requirement already satisfied: sqlparse in c:\users\kashi\appdata\local\p
rograms\python\python311\lib\site-packages (from ipython-sql) (0.4.4)
Requirement already satisfied: six in c:\users\kashi\appdata\local\progra
ms\python\python311\lib\site-packages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils in c:\users\kashi\appdata
\local\programs\python\python311\lib\site-packages (from ipython-sql) (0.
2.0)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\kashi
\appdata\local\programs\python\python311\lib\site-packages (from sqlalche
my>=2.0->ipython-sql) (4.9.0)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\kashi\appdata
\local\programs\python\python311\lib\site-packages (from sqlalchemy>=2.0-
>ipython-sql) (3.0.3)
Requirement already satisfied: backcall in c:\users\kashi\appdata\local\p
rograms\python\python311\lib\site-packages (from ipython->ipython-sql)
(0.2.0)
Requirement already satisfied: decorator in c:\users\kashi\appdata\local
\programs\python\python311\lib\site-packages (from ipython->ipython-sql)
(5.1.1)
Requirement already satisfied: jedi>=0.16 in c:\users\kashi\appdata\local
\programs\python\python311\lib\site-packages (from ipython->ipython-sql)
(0.18.2)
Requirement already satisfied: matplotlib-inline in c:\users\kashi\appdat
a\local\programs\python\python311\lib\site-packages (from ipython->ipytho
n-sql) (0.1.6)
Requirement already satisfied: pickleshare in c:\users\kashi\appdata\loca
l\programs\python\python311\lib\site-packages (from ipython->ipython-sql)
(0.7.5)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.11 in c:\users
\kashi\appdata\local\programs\python\python311\lib\site-packages (from ip
ython->ipython-sql) (3.0.36)
Requirement already satisfied: pygments>=2.4.0 in c:\users\kashi\appdata
\local\programs\python\python311\lib\site-packages (from ipython->ipython
-sql) (2.14.0)
Requirement already satisfied: stack-data in c:\users\kashi\appdata\local
\programs\python\python311\lib\site-packages (from ipython->ipython-sql)
(0.6.2)
Requirement already satisfied: traitlets>=5 in c:\users\kashi\appdata\loc
al\programs\python\python311\lib\site-packages (from ipython->ipython-sq
l) (5.8.0)
Requirement already satisfied: colorama in c:\users\kashi\appdata\local\p
rograms\python\python311\lib\site-packages (from ipython->ipython-sql)
(0.4.6)
Requirement already satisfied: wcwidth in c:\users\kashi\appdata\local\pr
ograms\python\python311\lib\site-packages (from prettytable->ipython-sql)
(0.2.5)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in c:\users\kashi\appd
ata\local\programs\python\python311\lib\site-packages (from jedi>=0.16->i
python->ipython-sql) (0.8.3)
Requirement already satisfied: executing>=1.2.0 in c:\users\kashi\appdata
\local\programs\python\python311\lib\site-packages (from stack-data->ipyt
hon->ipython-sql) (1.2.0)
Requirement already satisfied: asttokens>=2.1.0 in c:\users\kashi\appdata
\local\programs\python\python311\lib\site-packages (from stack-data->ipyt
hon->ipython-sql) (2.2.1)
Requirement already satisfied: pure-eval in c:\users\kashi\appdata\local
\programs\python\python311\lib\site-packages (from stack-data->ipython->i
python-sql) (0.2.2)


[notice] A new release of pip is available: 23.2.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [26]:
```
df= pd.DataFrame({'City': ['Mumbai', 'Mumbai','Mumbai','Mumbai','London','
                  'Country': ['India','India','India','India', 'United King
                  'Season': ['Winter','Sprng','Summer','Fall','Winter','Spr
                  'Temperature(C)': [24.8,28.4,27.9,27.6,4.2,8.3,15.7,10.4
                 'Rainfall(mm)': [5.9,16.2,1549.4,346.0,207.7,169.6,157.0,
```

In [27]:
```
df
```

Out[27]:

	City	Country	Season	Temperature(C)	Rainfall(mm)
0	Mumbai	India	Winter	24.8	5.9
1	Mumbai	India	Sprng	28.4	16.2
2	Mumbai	India	Summer	27.9	1549.4
3	Mumbai	India	Fall	27.6	346.0
4	London	United Kingdom	Winter	4.2	207.7
5	London	United Kingdom	Spring	8.3	169.6
6	London	United Kingdom	Summer	15.7	157.0
7	London	United Kingdom	Fall	10.4	218.5
8	Cairo	Egypt	Winter	13.6	16.5
9	Cairo	Egypt	Spring	20.7	6.5
10	Cairo	Egypt	Summer	27.7	0.1
11	Cairo	Egypt	Fall	22.2	4.5

In [47]:
```
cnn = sqlite3.connect('jupyter_sql_tutorial.db')
```

In [48]:
```
df.to_sql('people', cnn)
%load_ext sql
```

In [49]:

```
%sql sqlite:///jupyter_sql_tutorial.db
```

In [50]:

```
%%sql
SELECT *
FROM people
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[50]:

index	City	Country	Season	Temperature(C)	Rainfall(mm)
0	Mumbai	India	Winter	24.8	5.9
1	Mumbai	India	Sprng	28.4	16.2
2	Mumbai	India	Summer	27.9	1549.4
3	Mumbai	India	Fall	27.6	346.0
4	London	United Kingdom	Winter	4.2	207.7
5	London	United Kingdom	Spring	8.3	169.6
6	London	United Kingdom	Summer	15.7	157.0
7	London	United Kingdom	Fall	10.4	218.5
8	Cairo	Egypt	Winter	13.6	16.5
9	Cairo	Egypt	Spring	20.7	6.5
10	Cairo	Egypt	Summer	27.7	0.1
11	Cairo	Egypt	Fall	22.2	4.5

a) All the temperature data:

```
In [67]: %%sql

         SELECT "Temperature(C)"
         FROM people;
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[67]:

Temperature(C)
24.8
28.4
27.9
27.6
4.2
8.3
15.7
10.4
13.6
20.7
27.7
22.2

b) All the cities, but without repetition:

```
In [60]: %%sql

         SELECT DISTINCT City
         FROM people;
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[60]:

City
Mumbai
London
Cairo

c) All the records for India:

```
In [61]:
%%sql
SELECT *
FROM people
WHERE Country = 'India';
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[61]:

index	City	Country	Season	Temperature(C)	Rainfall(mm)
0	Mumbai	India	Winter	24.8	5.9
1	Mumbai	India	Sprng	28.4	16.2
2	Mumbai	India	Summer	27.9	1549.4
3	Mumbai	India	Fall	27.6	346.0

d) All the Fall records:

```
In [62]:
%%sql
SELECT *
FROM people
WHERE Season = 'Fall';
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[62]:

index	City	Country	Season	Temperature(C)	Rainfall(mm)
3	Mumbai	India	Fall	27.6	346.0
7	London	United Kingdom	Fall	10.4	218.5
11	Cairo	Egypt	Fall	22.2	4.5

e) The city, country, and season for which the average rainfall is between 200 and 400 millimeters:

In [63]:
```
%%sql
SELECT City, Country, Season
FROM people
GROUP BY City, Country, Season
HAVING AVG("Rainfall(mm)") BETWEEN 200 AND 400;
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[63]:

City	Country	Season
London	United Kingdom	Fall
London	United Kingdom	Winter
Mumbai	India	Fall

f) The city and country for which the average Fall temperature is above 20 degrees, in increasing temperature order:

In [64]:
```
%%sql
SELECT City, Country
FROM people
WHERE Season = 'Fall'
GROUP BY City, Country
HAVING AVG("Temperature(C)") > 20
ORDER BY AVG("Temperature(C)") ASC;
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[64]:

City	Country
Cairo	Egypt
Mumbai	India

g) The total annual rainfall for Cairo:

In [65]:
```
%%sql
SELECT SUM("Rainfall(mm)")
FROM people
WHERE City = 'Cairo';
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[65]:

SUM("Rainfall(mm)")
27.6

h) The total rainfall for each season:

In [66]:
```
%%sql
SELECT Season, SUM("Rainfall(mm)") AS Total_Rainfall
FROM people
GROUP BY Season;
```

```
 * sqlite:///jupyter_sql_tutorial.db
Done.
```

Out[66]:

Season	Total_Rainfall
Fall	569.0
Spring	176.1
Sprng	16.2
Summer	1706.5
Winter	230.1

Question 9

In [77]:
```
 words = ['The', 'quick', 'brown', 'fox', 'jumps', 'over',
'the', 'lazy', 'dog']
```

a) Convert all words to uppercase:

In [78]:
```
upper_words = [word.upper() for word in words]
upper_words
```

Out[78]:
```
['THE', 'QUICK', 'BROWN', 'FOX', 'JUMPS', 'OVER', 'THE', 'LAZY', 'DOG']
```

b) Leave all words as lowercase:

In [79]:
```
lower_words = [word.lower() for word in words]
lower_words
```

Out[79]:
```
['the', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']
```

c) Get the lengths of all words:

In [80]:
```
word_lengths = [len(word) for word in words]
word_lengths
```

Out[80]:
```
[3, 5, 5, 3, 5, 4, 3, 4, 3]
```

d) Generate a list of lists containing each word in uppercase, lowercase, and its length:

```
In [81]:  word_info = [[word.upper(), word.lower(), len(word)] for word in words]
          word_info
```

```
Out[81]: [['THE', 'the', 3],
          ['QUICK', 'quick', 5],
          ['BROWN', 'brown', 5],
          ['FOX', 'fox', 3],
          ['JUMPS', 'jumps', 5],
          ['OVER', 'over', 4],
          ['THE', 'the', 3],
          ['LAZY', 'lazy', 4],
          ['DOG', 'dog', 3]]
```

e) Filter words with 4 or more characters:

```
In [82]:  long_words = [word for word in words if len(word) >= 4]
          long_words
```

```
Out[82]: ['quick', 'brown', 'jumps', 'over', 'lazy']
```