Question 1

In [1]:
```
print (type(5))
print(type(5.0))
print(type(5 > 1))
print(type(5))
print(type(5 * 2))
print(type('5' * 2))
print(type('5' + '2'))
print(type(5 / 2))
print(type(5 % 2))
print(type({5, 2, 1}))
print(type(5 == 3))
import math
print(type(math.pi))
```

```
<class 'int'>
<class 'float'>
<class 'bool'>
<class 'int'>
<class 'int'>
<class 'str'>
<class 'str'>
<class 'float'>
<class 'int'>
<class 'set'>
<class 'bool'>
<class 'float'>
```

Question 2

In [ ]:
```
How many letters are there in 'Supercalifragilisticexpialidocious'?
```

In [3]:
```
print(len("Supercalifragilisticexpialidocious"))
```

```
34
```

In [ ]:
```
Does 'Supercalifragilisticexpialidocious' contain 'ice' as a substring?
```

In [4]:
```
print("ice"in("Supercalifragilisticexpialidocious") )
```

```
True
```

In [8]:
```
strings = ["Supercalifragilisticexpialidocious","Honorificabilitudinitatibus",
longest_word = ""
for x in strings:
    if(len(x)>len(longest_word)):
        longest_word = x
print("The longest word from the bunchHow many letters are there in 'Supercali
```

```
The longest word from the bunch is or Babababadalgharaghtakamminarronnkonn
```

In [ ]:
```
Which composer comes first in the dictionary: 'Berlioz', 'Borodin', 'Brian', '
Which one comes last?
```

In [ ]:
```
strings = ['Berlioz', 'Borodin', 'Brian','Bartok', 'Bellini', 'Buxtehude', 'Be
strings = sorted(strings)
print(strings)
```

Question 3

In [ ]:
```
Implement function triangleArea(a,b,c) that takes as input the lengths of the
and returns the area of the triangle. By Heron's formula, the area of a triang
  s(s-a)(s-b)(s-c)
  s=(a+b+c)/2
>>> triangleArea(2,2,2)
1.7320508075688772
```

In [20]:
```
import math
def triangleArea(a, b, c):
    s = (a + b + c) / 2
    area = math.sqrt(s * (s - a) * (s - b) * (s - c))
    return area
    result = triangleArea('2', '2', '2',)
print(result)
```

```
1.7320508075688772
```

Question 4

In [ ]:
```
Write a program in python to separate odd and even integers in separate arrays
Test Data :
Input the number of elements to be stored in the array :5
Input 5 elements in the array :
element - 0 : 25
element - 1 : 47
element - 2 : 42
element - 3 : 56
element - 4 : 32
Expected Output:
The Even elements are:
42 56 32
The Odd elements are :
25 47
```

In [24]:
```
Test_Data = int(input("Please enter the count of element: "))
even = []
odd  = []
for i in range(Test_Data):
    number = int(input(f" Enter element - {i}: "))
    if number % 2 == 0:
        even.append(number)
    else:
        odd.append(number)
print("\n The Even Elements are :\n", even)
print("\n The Odd Elements are :\n", odd)
```

```
Please enter the count of element: 5
 Enter element - 0: 25
 Enter element - 1: 47
 Enter element - 2: 42
 Enter element - 3: 56
 Enter element - 4: 32

 The Even Elements are :
 [42, 56, 32]

 The Odd Elements are :
 [25, 47]
```

Question 5

In [ ]:
```
Write a function inside(x,y,x1,y1,x2,y2) that returns True or False
depending on whether the point (x,y) lies in the rectangle with lower left cor
>>> inside(1,1,0,0,2,3)
True
>>> inside(-1,-1,0,0,2,3)
False
Use function inside() from part a. to write an expression that tests whether t
lies in both of the following rectangles: one with lower left corner (0.3, 0.5
and the other with lower left corner (0.5, 0.2) and upper right corner (1.1, 2
```

In [26]:
```
def inside(x, y, x1, y1, x2, y2):
    if x >= x1 and x <= x2 and y >= y1 and y <= y2:
        return True
    else:
        return False
x =  float(input("Please enter the value for x: "))
y =  float(input("Please enter the value for y: "))
x1 = float(input("Please enter the value for x1: "))
y1 = float(input("Please enter the value for y1: "))
x2 = float(input("Please enter the value for x2: "))
y2 = float(input("Please enter the value for y2: "))
print(inside(x, y, x1, y1, x2, y2))
if inside(x, y, x1, y1, x2, y2):
        print("The point(",x,",",y,") lies between the lower left point (",x1,
else:
        print("The point(",x,",",y,") do not lie between the lower left point
```

```
Please enter the value for x: -1
Please enter the value for y: -1
Please enter the value for x1: 0
Please enter the value for y1: 0
Please enter the value for x2: 2
Please enter the value for y2: 3
False
The point( -1.0 , -1.0 ) do not lie between the lower left point ( 0.0 , 0.0
) and the upper right point ( 2.0 , 3.0 ) in the rectangle
```

Question 6

In [ ]:
```
You can turn a word into pig-Latin using the following two rules (simplified):
If the word starts with a consonant, move that letter to the end and append 'a
For example, 'happy' becomes 'appyhay' and 'pencil' becomes 'encilpay'.
If the word starts with a vowel, simply append 'way' to the end of the word.
For example, 'enter' becomes 'enterway' and 'other' becomes 'otherway' .
For our purposes, there are 5 vowels: a, e, i, o, u (so we count y as a conson
Write a function pig() that takes a word (i.e., a string) as input and returns
Your function should still work if the input word contains upper case characte
Your output should always be lower case however.
>>> pig('happy')
'appyhay'
>>> pig('Enter')
'enterway'
```

In [27]:
```
def pig(word):
    vowels = ['a', 'e', 'i', 'o', 'u']
    word = word.lower()

    if word[0] in vowels:
        pig_latin = word + 'way'
    else:
        pig_latin = word[1:] + word[0] + 'ay'

    return pig_latin

Consonant_output = str(input("Enter the word starts from a consonant: "))
Vowels_output = str(input("Enter the word starts from a vowel: "))

print("\nThe pig_latin word for ",Consonant_output," is ",pig(Consonant_output
print("The pig_latin word for ",Vowels_output," is ",pig(Vowels_output))
```

```
Enter the word starts from a consonant: happy
Enter the word starts from a vowel: Enter

The pig_latin word for  happy  is  appyhay
The pig_latin word for  Enter  is  enterway
```

Question 7

In [ ]:
```
File bloodtype1.txt records blood-types of patients (A, B, AB, O or OO) at a c
Write a function bldcount() that reads the file with name name and reports (i.
how many patients there are in each bloodtype.
>>> bldcount('bloodtype.txt')
There are 10 patients of blood type A.
There is one patient of blood type B.
There are 10 patients of blood type AB.
There are 12 patients of blood type O.
There are no patients of blood type OO.
```

In [28]:
```
#f = open("bloodtype1.txt", "r")
#get_data = f.read()
#just to show the program works
get_data = "AB AB B O A A AB O AB A O O A A A O O O AB O A A A A A AB AB A AB
get_count = 0
covert_array = get_data.split()
base_data = ['A','B','AB','O','OO']
for x in base_data:
    get_count = covert_array.count(x)
    print("There are {0} patients with blood group {1}".format(get_count,x))
```

```
There are 15 patients with blood group A
There are 1 patients with blood group B
There are 13 patients with blood group AB
There are 15 patients with blood group O
There are 0 patients with blood group OO
```

Question 8

In [ ]:
```
Write a function curconv() that takes as input:
a currency represented using a string (e.g., 'JPY' for the Japanese Yen or 'EU
an amount
and then converts and returns the amount in US dollars.
>>> curconv('EUR', 100)
122.96544
>>> curconv('JPY', 100)
1.241401
The currency rates you will need are stored in file currencies.txt:
AUD 1.0345157 Australian Dollar
CHF 1.0237414 Swiss Franc
CNY 0.1550176 Chinese Yuan
DKK 0.1651442 Danish Krone
EUR 1.2296544 Euro
GBP 1.5550989 British Pound
HKD 0.1270207 Hong Kong Dollar
INR 0.0177643 Indian Rupee
JPY 0.01241401 Japanese Yen
MXN 0.0751848 Mexican Peso
MYR 0.3145411 Malaysian Ringgit
NOK 0.1677063 Norwegian Krone
NZD 0.8003591 New Zealand Dollar
PHP 0.0233234 Philippine Peso
SEK 0.148269 Swedish Krona
SGD 0.788871 Singapore Dollar
THB 0.0313789 Thai Baht
```

In [31]:
```
def curcov(curr_input,val_input):
    #f = open("currencies.txt", "r")
    #get_data = f.read()
    split_data = ['AUD', '1.0345157', 'Australian', 'Dollar', 'CHF', '1.023741
                  '0.1550176', 'Chinese', 'Yuan', 'DKK', '0.1651442', 'Danish'
                  'Euro', 'GBP', '1.5550989', 'British', 'Pound', 'HKD', '0.12
                  'INR', '0.0177643', 'Indian', 'Rupee', 'JPY', '0.01241401',
                  'Mexican', 'Peso', 'MYR', '0.3145411', 'Malaysian', 'Ringgit
                  'NZD', '0.8003591', 'New', 'Zealand', 'Dollar', 'PHP', '0.02
                  'Swedish', 'Krona', 'SGD', '0.788871', 'Singapore', 'Dollar'
    #split_data = get_data.split()
    get_index = split_data.index(curr_input)
    conv_value = split_data[get_index+1]
    calculate = float(conv_value) * float(val_input)
    print(calculate)

print("Enter Currency")
curr_input = input()
print("Enter Value")
val_input = input()
curcov(curr_input,val_input)
```

```
Enter Currency
SGD
Enter Value
500
394.4355
```

Question 9

In [ ]:
```
Each of the following will cause an exception (an error). Identify what type o
Trying to add incompatible variables, as inadding6 + ‘a’
Referring to the 12th item of a list that has only 10items
Using a value that is out of range for a function’sinput, such as callingmath.
Using an undeclared variable, such asprint(x) when x has not been defined
Trying to open a file that does not exist, such asmistyping the file name or l
```

In [32]:
```
XYZ = 6+'a'
print=XYZ
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[32], line 1
----> 1 XYZ = 6+'a'
      2 print=XYZ

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

In [33]:
```
XYZ = ["AAA", "BBB", "CCC", "DDD", "EEE", "FFF", "GGG", "HHH", "III", "JJJ"]
print(XYZ[12])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[33], line 2
      1 XYZ = ["AAA", "BBB", "CCC", "DDD", "EEE", "FFF", "GGG", "HHH", "III",
"JJJ"]
----> 2 print(XYZ[12])

IndexError: list index out of range
```

In [34]:
```
import math
result = math.sqrt(-1.0)
print()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[34], line 2
      1 import math
----> 2 result = math.sqrt(-1.0)
      3 print()

ValueError: math domain error
```

In [37]:
```
Cell In[55], line 7
      1 # 4. Using an undeclared variable
----> 2 print(x)
      3 print()
```

```
  Cell In[37], line 1
    Cell In[55], line 7
         ^
SyntaxError: invalid syntax
```

In [38]:
```
Cell In[90], line 20
      1 # 5. Trying to open a file that does not exist
----> 2 file = open('nonexistent_file.txt', 'r')
      3 print()
```

```
  Cell In[38], line 1
    Cell In[90], line 20
         ^
SyntaxError: invalid syntax
```

Question 10

In [ ]:
```
Encryption is the process of hiding the meaning of a text by substituting lett
according to some system. If the process is successful, no one but the intende
Cryptanalysis refers to attempts to undo the encryption, even if some details
(for example, if an encrypted message has been intercepted).
The first step of cryptanalysis is often to build up a table of letter frequen
Assume that the string letters is already defined as 'abcdefghijklmnopqrstuvwx
Write a function called frequencies() that takes a string as its only paramete
showing the number of times each character appears in the text.
Your function may ignore any characters that are not in letters.

>>> frequencies('The quick red fox got bored and went home.')

[1, 1, 1, 3, 5, 1, 1, 2, 1, 0, 1, 0, 1, 2, 4, 0, 1, 2, 0, 2, 1, 0, 1, 1, 0, 0]
```

In [46]:
```
def frequencies(text):
    characters = 'abcdefghijklmnopqrstuvwxyz'
    characters_count = [0] * 26
    lower_text = text.lower()
    for char in lower_text:
        if char in characters:
            index = characters.index(char)
            characters_count[index] += 1
    return characters_count
encrypted_text = str(input("Enter any String : "))
letters = [chr(i) for i in range(ord('a'), ord('z') + 1)]
output = frequencies(encrypted_text)
print(output)
```

```
Enter any String : 'The quick red fox got bored and went home.'
[1, 1, 1, 3, 5, 1, 1, 2, 1, 0, 1, 0, 1, 2, 4, 0, 1, 2, 0, 3, 1, 0, 1, 1, 0,
0]
```