

Syed Fasih Uddin ~ 19B-029-SE ~ Chapter 2 - Book Unsolved Problems

2.1: Write Python algebraic expressions corresponding to the following statements:

In [1]:

```
# (a) The sum of the first five positive integers
sum = (5+2+3+1+4)
print("Sum of 5 integers '5,2,3,1,4 is':",sum)

# (b) The average age of Sara (age 23), Mark (age 19), and Fatima (age 31)
sara= 23
mark = 19
fatima=31
avg= (sara+mark+fatima)/3
print("Average age of Sara, Mark and Fatima is:",avg)

# (c) The number of times 73 goes into 403
# (d) The remainder when 403 is divided by 73
a =403
divisor = 73
ans = a // divisor
remainder = a%divisor
print(divisor,"goes into",a,ans,"times with remainder",remainder)

# (e) 2 to the 10th power
print(2**10)
# (f) The absolute value of the difference between Sara's height (54 inches) and Mark's height (57 inches)
sara_height= float(54)
mark_height=float(57)
print("The absolute value of the difference between Sara's height is:",mark_height - sara_height)

# (g) The lowest price among the following prices: $34.99, $29.95, and $31.50
price1=34.99
price2=29.95
price3=31.50
print(min(price1,price2,price3))
```

Sum of 5 integers '5,2,3,1,4 is': 15

Average age of Sara, Mark and Fatima is: 24.333333333333332

73 goes into 403 5 times with remainder 38

1024

The absolute value of the difference between Sara's height is: 3.0 inches

29.95

2.2 : Translate the following statements into Python Boolean expressions and evaluate them:

In [2]:

```
# (a) The sum of 2 and 2 is Less than 4
print((2+2) < 4)
# (b) The value of 7 // 3 is equal to 1 + 1
print(7 // 3 == 1 + 1)
# (c) The sum of 3 squared and 4 squared is equal to 25.
print(3**2 + 4**2 == 25)
# (d) The sum of 2, 4, and 6 is greater than 12.
print(2+4+6 > 12)
# (e) 1387 is divisible by 19.
print(1387 % 19 == 0)
# (f) 31 is even. (Hint: what does the remainder when you divide by 2 tell you?)
print(31 % 2 == 0)
# (g) The Lowest price among $34.99, $29.95, and $31.50 is Less than $30.00.
print(min(34.99,29.95,31.50,) < 30.00)
```

False

True

True

False

True

False

True

2.3 Write Python statements that correspond to the actions below and execute them:

In [3]:

```
# (a) Assign integer value 3 to variable a.
a = int(3)
# (b) Assign 4 to variable b
b = 4
# (c) Assign to variable c the value of expression a * a + b * b.
c = a * a + b * b
print("value of c is =",c)
```

value of c is = 25

In [4]:

```
#2.4
#Start by executing the assignment statements:
s1 = 'ant'
s2 = 'bat'
s3 = 'cod'
#Write Python expressions using s1, s2, and s3 and operators + and * that evaluate to:
#(a) 'ant bat cod'
print(s1+s2+s3)
#(b) 'ant ant ant ant ant ant ant ant ant '
print(s1*10)
#(c) 'ant bat bat cod cod cod'
print(s1+s2*2+s3*3)
#(d) 'ant bat ant bat ant bat ant bat ant bat ant bat '
print((s1+s2)*7)
#(e) 'batbatcod batbatcod batbatcod batbatcod batbatcod '
print(((s2*2)+(s3))*5)
```

```
antbatcod
antantantantantantantantantant
antbatbatcodcodcod
antbatantbatantbatantbatantbatantbat
batbatcodbatbatcodbatbatcodbatbatcodbatbatcod
```

In [5]:

```
#2.5
#Start by executing the assignment:
s = '0123456789'
#Now write expressions using string s and the indexing operator that evaluate to:
#(a) '0'
print(s[0])
#(b) '1'
print(s[1])
#(c) '6'
print(s[6])
#(d) '8'
print(s[8])
#(e) '9'
print(s[9])
```

```
0
1
6
8
9
```

In [6]:

```
#2.6
#First execute the assignment
words = ['bat', 'ball', 'barn', 'basket', 'badminton']
#Now write two Python expressions that evaluate to the first and last, respectively, word in
dict = {words[0]:words[-1]}
print(dict)
```

```
{'bat': 'badminton'}
```

In [7]:

```

#2.7
#Given the List of student homework grades
grades = [9, 7, 7, 10, 3, 9, 6, 6, 2]
#write:
#(a) An expression that evaluates to the number of 7 grades
print(grades.count(7))
#(b) A statement that changes the last grade to 4
grades[-1]=4
print(grades)
#(c) An expression that evaluates to the maximum grade
print(max(grades))
#(d) A statement that sorts the list grades
grades.sort()
print(grades)
#(e) An expression that evaluates to the average grade
print((grades[0]+grades[1]+grades[2]+grades[3]+grades[4]+grades[5]+grades[6]+grades[7]+grades[8])/len(grades))

```

```

2
[9, 7, 7, 10, 3, 9, 6, 6, 4]
10
[3, 4, 6, 6, 7, 7, 9, 9, 10]
52.111111111111114

```

Practice Problem 2.8

Q: In what order are the operators in the following expressions evaluated?

(a) $2 + 3 == 4$ or $a \geq 5$

First condition $2+3=4$ is checked then it is checked whether a is greater or equals to 5.

(b) $lst[1] * -3 < -10 == 0$

List's 2nd element is multiplied by negative of 3 then then it is checked whether ls is less than negative 10 and then it is checked that whether whole thing is equals to zero

(c) $(lst[1] * -3 < -10) \text{ in } [0, \text{True}]$

List 2nd element is multiplied by negative of 3 then it is checked whether it is less then negative of 10. Afterwards it is checked whether the answer is included in the list.

(d) $2 * 3 ** 2$

2 is multiplied by square of 3

(e) $4 / 2 \text{ in } [1, 2, 3]$

First 4 is divided by 2 and then it checked whether the answer (2) is in the list or not. Resulting true

Practice Problem 2.9

What is the type of the object that these expressions evaluate to?

- (a) `False + False` : Integer
- (b) `2 * 3**2.0` : Float
- (c) `4 // 2 + 4 % 2` : Integer
- (d) `2 + 3 == 4 or 5 >= 5` : Boolean

2.10 Write Python expressions corresponding to the following:

In [8]:

```
#(a) The length of the hypotenuse in a right triangle whose other two sides have lengths a
a = eval(input("Enter length of Perpendicular (in cm):"))
b = eval(input("Enter length of Base (in cm):"))
c= pow(a,2) + pow(b,2)

#(b) The value of the expression that evaluates whether the Length of the above hypotenuse
print ("Length of hypotenuse in right triangle is",c,"cm")
if c == 5:
    print("Value of Hyp is = 5 cm")

#(c) The area of a disk of radius a
```

```
Enter length of Perpendicular (in cm):2
Enter length of Base (in cm):3
Length of hypotenuse in right triangle is 13 cm
```

In [9]:

```
#(c) The area of a disk of radius a
from math import pi
a= eval(input("Enter radius (in meters):"))
area= (pi)*(a**2)
print("Area of disk is:",area,"m^2")
```

```
Enter radius (in meters):2
Area of disk is: 12.566370614359172 m^2
```

In [10]:

```
#(d) The value of the Boolean expression that checks whether a point with coordinates x and
radius= float(input("Enter Radius (cm):"))
x= float(input("Enter x-axis of point:"))
y = float(input("Enter y-axis of point:"))
#From Center to this point distance must be Less than radius
from math import sqrt
distance = sqrt(pow((x-0),2)+pow(y-0,2))
if distance > radius:
    print("Point(",x,",",y,")doesn't lie inside the circle")
elif distance < radius:
    print("Point(",x,",",y,")lies inside the circle")
else:
    print("Error: No Condition was passed")
```

```
Enter Radius (cm):6
Enter x-axis of point:3
Enter y-axis of point:2
Point( 3.0 , 2.0 )lies inside the circle
```

2.11 Write Python expressions corresponding to these statements:

In [11]:

```
#(a) The sum of negative integers -7 through -1
def sum():
    b=0
    for a in range(-7,-1):
        b=b+a
    print(b)
sum()
```

```
-27
```

In [12]:

```
#(b) The average age of a group of kids at a summer camp given than 17 are 9 years old, 24
avg=(17*9 + 24*10 + 21*11 + 27*12)/(17+24+21+27)
print("Average age is:",avg,"years")
#(c) 2 to the power -20
print(2**-20)
#(d) The number of times 61 goes into 4356
print(int(4356/61))
#(e) The remainder when 4365 is divided by 61
print( 4365 % 61)
```

```
Average age is: 10.651685393258427 years
9.5367431640625e-07
71
34
```

2.12 Start by evaluating, in the interactive shell, the assignment:

In [13]:

```

s1 = '-'
s2 = '+'
#Now write string expressions involving s1 and s2 and string operators + and * that evaluate
#(a) '-+'
print(s1+s2)
#(b) '-+-'
print(s1+s2+s1)
#(c) '+--'
print(s2 + 2*s1)
#(d) '+---'
print(2*(s2 + 2*s1))
#(e) '+-----+-----+-----+-----+-----+'
print(10*(s2 + 2*s1) + s2)
#(f) '+-----+-----+-----+-----+-----+'
print(5*(s2+s1+3*s2+2*s1))

```

```

-+
-+-
+--
+---+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+

```

2.13 Start by running, in the shell, the following assignment statement:

In [14]:

```

s = 'abcdefghijklmnopqrstuvwxyz'
#Now write expressions using string s and the indexing operator that evaluate to 'a', 'c',
print(s[0])
print(s[2])
print(s[25])
print(s[24])
print(s[12])

```

```

a
c
z
y
m

```

2.14 Start by executing

In [15]:

```

s = 'goodbye'
#Then write a Boolean expression that checks whether:
#(a) The first character of string s is 'g'
print(s[0] == 'g')
#(b) The seventh character of s is 'g'
print(s[6] == 'g')
#(c) The first two characters of s are 'g' and 'a'
print((s[0] == 'g' and s[1] == 'a'))
#(d) The next to last character of s is 'x'
print(s[-1] == 'x' )
#(e) The middle character of s is 'd'
print(s[len(s)//2] == 'd')
#(f) The first and last characters of string s are equal
print(s[0] == s[-1])
#(g) The last four characters of string s match the string 'tion'
print(s[-4:] == "tion")

```

True
 False
 False
 False
 True
 False
 False

2.15 Write Python expressions corresponding to these statements:

In [16]:

```

#(a) The number of characters in the word "anachronistically" is 1 more than the number of
print(len("anachronistically") == 1+ len("counterintuitive"))
#(b) The word "misinterpretation" appears earlier in the dictionary than the word "misrepresentation"
print("misinterpretation" < "misrepresentation")
#(c) The letter "e" does not appear in the word "floccinaucinihilipilification."
print("floccinaucinihilipilification".find("e") == -1)
#(d) The number of characters in the word "counterrevolution" is equal to the sum of the number of characters in "counter" and "resolution"
print(len("counterrevolution") == len("counter") + len("resolution"))

```

True
 True
 True
 True

2.16 Write the corresponding Python assignment statements:

In [17]:

```

#(a) Assign 6 to variable a and 7 to variable b.
a=6
b=7
print("a =",a,"b =",b)
#(b) Assign to variable c the average of variables a and b.
c = (a+b)/2
print(c)
#(c) Assign to variable inventory the list containing strings 'paper', 'staples', and 'pencils'
inventory = ["paper","staples","pencils"]
print(inventory)
#(d) Assign to variables first, middle and last the strings 'John', 'Fitzgerald', and 'Kennedy'
first, middle, last = "John","Fitzgerald","kennedy"
#(e) Assign to variable fullname the concatenation of string variables first, middle, and last
fullname = first+" "+middle+" "+last
print(fullname)

```

```

a = 6 b = 7
6.5
['paper', 'staples', 'pencils']
John Fitzgerald kennedy

```

2.17 Using variables defined in Exercise 2.16, write Boolean expressions corresponding to the following logical statements and evaluate the expressions:

In [18]:

```

#(a) The sum of 17 and -9 is Less than 10.
print(17-9 < 10)
#(b) The length of List inventory is more than five times the length of string fullname
print(len(inventory) > 5*len(fullname))
#(c) c is no more than 24.
print(c <= 24)
#(d) 6.75 is between the values of integers a and b.
print((6.75 < a and 6.75 > b) or (6.75 < b and 6.75 > a))
#(e) The length of string middle is larger than the length of string first and smaller than the length of string last
print(len(middle) > len(first) and len(middle) < len(last))
#(f) Either the List inventory is empty or it has more than 10 objects in it.
x= [1,2,3]
print(len(x)==0 or len(x) > 10)

```

```

True
False
True
True
False
False

```

2.18 Write Python statements corresponding to the following:

In [19]:

```
#(a) Assign to variable flowers a list containing strings 'rose', 'bougainvillea', 'yucca',
flowers = ["rose","bougainvillea","yucca","marigold", "daylilly", "lilly of the valley"]
#(b) Write a Boolean expression that evaluates to True if string 'potato' is in list flower
print("potato" in flowers)
#(c) Assign to list thorny the sublist consisting of the first three objects in list flower
thorny = flowers[0:3]
print(thorny)
#(d) Assign to list poisonous the sublist consisting of just the last object of list flower
poisonous = [flowers[-1]]
print(poisonous)
#(e) Assign to list dangerous the concatenation of lists thorny and poisonous.
dangerous = thorny + poisonous
print(dangerous)
```

```
False
['rose', 'bougainvillea', 'yucca']
['lilly of the valley']
['rose', 'bougainvillea', 'yucca', 'lilly of the valley']
```

2.19 Start by assigning to variable answers a list containing an arbitrary sequence of strings 'Y' and 'N'. For example:

In [20]:

```
answers = ['Y', 'N', 'N', 'Y', 'N', 'Y', 'Y', 'Y', 'N', 'N', 'N']
#Write Python statements corresponding to the following:
#(a) Assign to variable numYes the number of occurrences of 'Y' in list answers.
num = answers.count('Y')
print(num)
#(b) Assign to variable numNo the number of occurrences of 'N' in list answers.
num = answers.count('N')
print(num)
#(c) Assign to variable percentYes the percentage of strings in answers that are 'Y'.
percent = answers.count('Y')/len(answers)
print(percent)
#(d) Sort the list answers.
answers.sort()
print(answers)
#(e) Assign to variable f the index of the first occurrence of 'Y' in sorted list answers.
f = answers.index('Y')
print(f)
```

```
5
6
0.45454545454545453
['N', 'N', 'N', 'N', 'N', 'N', 'Y', 'Y', 'Y', 'Y', 'Y']
6
```

Write an expression involving a three-letter string s that evaluates to a string whose characters are the characters of s in reverse order. If s is 'top', the expression should evaluate to 'pot'.

In [21]:

```
s='top'  
t= s[::-1]  
print(t)
```

pot

2.21 Write an expression involving strings *s* and *t* containing the last name and the first name, respectively, of a person that evaluates to the person's initials. If the two strings contained the first and last name of this book's author, the expression would evaluate to 'LP'.

In [22]:

```
s='Syed'  
t='Fasih'  
print(s[0]+t[0])
```

SF

2.22 The range of a list of numbers is the largest difference between any two numbers in the list. Write a Python expression that computes the range of a list of numbers *lst*. If the list *lst* is, say, [3, 7, -2, 12], the expression should evaluate to 14 (the difference between 12 and -2).

In [23]:

```
lst= [3, 7, -2, 12]  
print(max(lst) - min(lst))
```

14

2.23 Start by assigning to variables *monthsL* and *monthsT* a list and a tuple, respectively, both containing strings 'Jan', 'Feb', 'Mar', and 'May', in that order. Then attempt the following with both containers:

In [24]:

```

monthsL = ['Jan', 'Feb', 'Mar', 'May']
monthsT = ('Jan', 'Feb', 'Mar', 'May')
#(a) Insert string 'Apr' between 'Mar' and 'May'.
monthsL.insert(3, "Apr")
print(monthsL)
#(b) Append string 'Jun'.
monthsL.append('Jun')
print(monthsL)
#(c) Pop the container.
monthsL.pop()
print(monthsL)
#(d) Remove the second item in the container
monthsL.remove(monthsL[1])
print(monthsL)
#(e) Reverse the order of items in the container.
monthsL.reverse()
print(monthsL)
#(f) Sort the container.
monthsL.sort()
print(monthsL)

```

```

['Jan', 'Feb', 'Mar', 'Apr', 'May']
['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun']
['Jan', 'Feb', 'Mar', 'Apr', 'May']
['Jan', 'Mar', 'Apr', 'May']
['May', 'Apr', 'Mar', 'Jan']
['Apr', 'Jan', 'Mar', 'May']

```

2.24 Start by assigning to variable `grades` a list containing an arbitrary sequence of grades (strings) 'A', 'B', 'C', 'D', and 'F'. For example:

In [25]:

```

grades = ['B', 'B', 'F', 'C', 'B', 'A', 'A', 'D', 'C', 'D', 'A', 'A', 'B']
#the numbers of occurrences of each grade in list grades in alphabetic order. For the given
count = [grades.count('A'), grades.count('B'), grades.count('C'), grades.count('D'), grades.cou
print(count)

```

```
[4, 4, 2, 2, 1]
```

2.25 Repeat Problem 2.24 with the following modification: variable `grades` is defined to be of type tuple rather than of type list:

In [26]:

```

grades = ('B', 'B', 'F', 'C', 'B', 'A', 'A', 'D', 'C', 'D', 'A', 'A', 'B')
#Variable count should still refer to a list.
count = [grades.count('A'), grades.count('B'), grades.count('C'), grades.count('D'), grades.cou
print(count)

```

```
[4, 4, 2, 2, 1]
```

2.26 A dartboard of radius 10 and the wall it is hanging on are represented using the twodimensional coordinate system, with the board's center at coordinate (0, 0). Variables `x` and `y` store the x- and y-coordinate of a dart hit. Write an expression using variables `x` and `y` that evaluates to True if the dart hits (is within) the dartboard, and

evaluate the expression for these dart coordinates

In [27]:

```
#(a) (0, 0)
#(b) (10, 10)
#(c) (6, -6)
#(d) (-7, 8)
r = 10
xaxes = [0,10,6,-7]
yaxes = [0,10,-6,8]
for x in xaxes:
    if x == xaxes[0]:
        y= yaxes[0]
    elif x == xaxes[1]:
        y=yaxes[1]
    elif x == xaxes[2]:
        y=yaxes[2]
    elif x == xaxes[3]:
        y=yaxes[3]
    else:
        print("List is too big")
    if x**2 + y**2 < r**2 :
        print("Dart hits the the dartboard")
    else:
        print("Dart doesn't hit the dartboard")
```

Dart hits the the dartboard
 Dart doesn't hit the dartboard
 Dart hits the the dartboard
 Dart doesn't hit the dartboard

2.27 A ladder put up right against a wall will fall over unless put up at a certain angle less than 90 degrees. Given variables length and angle storing the length of the ladder and the angle that it forms with the ground as it leans against the wall, write a Python expression involving length and angle that computes the height reached by the ladder. Evaluate the expression for these values of length and angle:

In [28]:

```
#(a) 16 feet and 75 degrees
import math
hyp = float(input("Enter Length of Ladder (feet):"))
theta = float(input("Enter angle of ladder:"))
rad= math.pi * (theta)/180
Perp = hyp * math.sin(rad)
print ("The height of ladder is ", Perp, "feet")
```

Enter Length of Ladder (feet):16
 Enter angle of ladder:75
 The height of ladder is 15.454813220625093 feet

In [29]:

```
#(b) 20 feet and 0 degrees
import math
hyp = float(input("Enter Length of Ladder (feet):"))
theta = float(input("Enter angle of ladder:"))
rad= math.pi * (theta)/180
Perp = hyp * math.sin(rad)
print ("The height of ladder is ", Perp,"feet")
```

```
Enter Length of Ladder (feet):20
Enter angle of ladder:0
The height of ladder is  0.0 feet
```

In [30]:

```
#(c) 24 feet and 45 degrees
import math
hyp = float(input("Enter Length of Ladder (feet):"))
theta = float(input("Enter angle of ladder:"))
rad= math.pi * (theta)/180
Perp = hyp * math.sin(rad)
print ("The height of ladder is ", Perp,"feet")
```

```
Enter Length of Ladder (feet):24
Enter angle of ladder:45
The height of ladder is  16.970562748477143 feet
```

In [31]:

```
#(d) 24 feet and 80 degrees
import math
hyp = float(input("Enter Length of Ladder (feet):"))
theta = float(input("Enter angle of ladder:"))
rad= math.pi * (theta)/180
Perp = hyp * math.sin(rad)
print ("The height of ladder is ", Perp,"feet")
```

```
Enter Length of Ladder (feet):24
Enter angle of ladder:80
The height of ladder is  23.63538607229299 feet
```

2.28 Write the relevant Python expression or statement, involving a list of numbers `lst` and using list operators and methods for these specifications:

In [32]:

```
#(a) An expression that evaluates to the index of the middle element of lst
lst = [4,5,6,7,8,9,10]
middle = int(len(lst) / 2)
print("Middle element Index is:" , middle)
```

```
Middle element Index is: 3
```

In [33]:

```
#(b) An expression that evaluates to the middle element of lst
lst = [4,5,6,7,8,9,10]
print("list is ", lst)
print("middle value is ",lst[int(len(lst)/2)])
```

list is [4, 5, 6, 7, 8, 9, 10]

middle value is 7

In [34]:

```
#(c) A statement that sorts the List lst in descending order
numbers = [10,20,30,40,50,60]
# Sorting List of Integers in descending order
numbers.sort(reverse = True )
print(numbers)
```

[60, 50, 40, 30, 20, 10]

In [36]:

```
#(d) A statement that removes the first number of list lst and puts it at the end
lst = [55,89,22,59,109,22]
# Program that removes first number and puts it to the end
lst.append(lst[0])
lst.remove(lst[0])
print (lst)
```

[89, 22, 59, 109, 22, 55]

2.29 Add one or more pairs of parentheses to each expression so that it evaluates to True.

In [37]:

```
#(a) 0 == 1 == 2
print(0 == (1 == 2))
#(b) 2 + 3 == 4 + 5 == 7
print((2 + (3 == 4) + 5) == 7)
#(c) 1 < -1 == 3 > 4
print((1 < -1) == (3 > 4))
```

True

True

True

2.30 Using an example of your own, explicitly convert some string to a list. Describe, in your own words, the behavior of the list constructor on a string input.

In [38]:

```
s= "StRiNGG"
s1= list(s)
print(s1)
```

['S', 't', 'R', 'i', 'N', 'G', 'G']

2.31 In this chapter we have covered some, but not all, methods of class list. Using the following interactive session as an aid, explain in your own words what the list methods `extend()`, `copy()`, and `clear()` do.

In [40]:

```
lst = [2, 3, 4]
lst.extend([5, 6])
print(lst)
print(".extend function merges another list in given list")
lst2 = lst.copy()
print(lst2)
print(".copy function copies list into another variable")
lst.clear()
print(lst)
print(".clear function deletes all the values inside list")
```

```
[2, 3, 4, 5, 6]
.extend function merges another list in given list
[2, 3, 4, 5, 6]
.copy function copies list into another variable
[]
.clear function deletes all the values inside list
```