PYTHON WORK SHEET SET 02

Q1 to Q7 have only one correct answer. Choose the correct option to answer your question.

1. Which of the following is not a core datatype in python?

A) list B) struct C) tuple D) set

Answer: B) struct

2. Which of the following is an invalid variable name in python?

A) _init_ B) no_1 C) 1_no D) _1

Answer: C) 1_no

3. Which one of the following is a keyword in python?

A) in B) _init_ C) on D) foo

Answer: A) in

4. In which of the following manner are the operators of the same precedence executed in python?

A) Left to Right B) BODMAS C) Right to Left D) None of these

Answer: A) Left to Right

5. Arrange the following in decreasing order of the precedence when they appear in an expression in python? i) Multiplication ii) Division iii) Exponential iv) Parentheses

A) iii – iv – ii – I B) iii – iv – i – ii C) iv – iii – ii – I D) iii – ii – i – iv

Answer: C) iv – iii – ii – I

6. (28//6)**3/3%3 = ?

A) 7.1111… B) 0 C) 0.3333… D) 1

Answer: C) 0.3333…

7. a = input("Enter an integer"). What will be the data type of a?

A) int B) str C) float D) double

Answer: B) str

Q8 and Q10 have multiple correct answers. Choose all the correct options to answer your question.

8. Which of the following statements are correct?

A) Division and multiplication have same precedence in python

B) Python's operators' precedence is based on PEDMAS

C) Python's operators' precedence is based on VBODMAS

D) In case of operators' having the same precedence, the one on the left side is executed first.

Answer: A) Division and multiplication have same precedence in python, D) In case of operators' having the same precedence, the one on the left side is executed first.

9. Which of the following is(are) valid statement(s) in python?

A) abc = 1,000,000   B) a b c = 1000 2000 3000

C) a,b,c = 1000, 2000, 3000   D) a_b_c = 1,000,000

Answer: A) abc = 1,000,000, C) a,b,c = 1000, 2000, 3000, D) a_b_c = 1,000,000

10. Which of the following is not equal to x16 in python?

A) x**4**4   B) x**16   C) x^16   D) (x**4)**4

Answer: C) x^16, A) x**4**4, B) x**16, D) (x**4)**4

All are indicating exponential operation, not multiplication, ^ is not exponentiation

<u>Q11 to Q13 are subjective questions, answer them briefly</u>

11. Differentiate between a list, tuple, set and dictionary.

Answer:

In Python, there are four built-in data types for storing collections of data: lists, tuples, sets, and dictionaries. Here's how they differ:

1. Lists: A list is a mutable, ordered collection of elements, which means that it can be changed after creation, and the order of the elements in the list is preserved. Lists are created using square brackets [] and elements are separated by commas.

Example: my_list = [1, 2, 3, "hello", True]

2. Tuples: A tuple is an immutable, ordered collection of elements, which means that it cannot be changed after creation, and the order of the elements in the tuple is preserved. Tuples are created using parentheses () and elements are separated by commas.

Example: my_tuple = (1, 2, 3, "hello", True)

3. Sets: A set is a mutable, unordered collection of unique elements, which means that it can be changed after creation, but duplicates are not allowed. Sets are created using curly braces {} or the set() function.

Example: my_set = {1, 2, 3, "hello", True}

4. Dictionaries: A dictionary is a mutable, unordered collection of key-value pairs, which means that it can be changed after creation, and the elements are accessed by their keys rather than their position. Dictionaries are created using curly braces {} and the key-value pairs are separated by colons :.

Example: my_dict = {"name": "John", "age": 30, "gender": "male"}

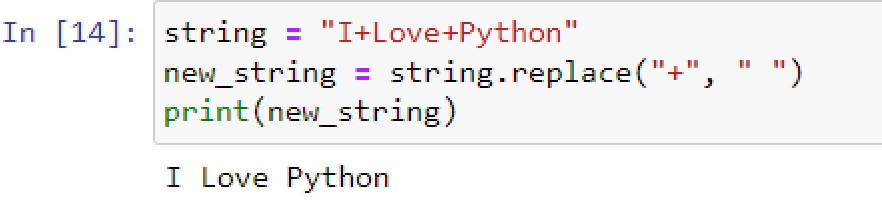
In summary, lists and tuples are ordered collections of elements, while sets and dictionaries are unordered collections. Lists and dictionaries are mutable, while tuples and sets are immutable.

12. Are strings mutable in python? Suppose you have a string “I+Love+Python”, write a small code to replace ‘+’ with space in python.

Answer:

No, strings are immutable in Python. Once a string is created, its contents cannot be modified. If you want to modify a string, you have to create a new string with the modified contents.

To replace ‘+’ with space in the string “I+Love+Python”, we can use the **replace()** method in Python. Here is the code:



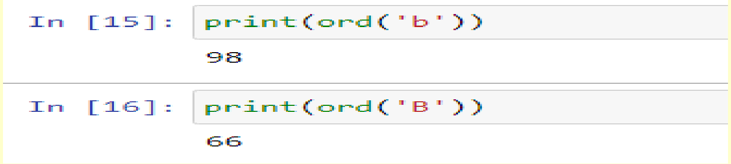
```
In [14]: string = "I+Love+Python"
         new_string = string.replace("+", " ")
         print(new_string)

         I Love Python
```

13. What does the function ord() do in python? Explain with an example. Also, write down the function for getting the data type of a variable in python.

Answer:

The ord() function in Python returns an integer representing the Unicode character. It takes a string of length 1 as an argument and returns the Unicode code point (integer representation) of the passed character.

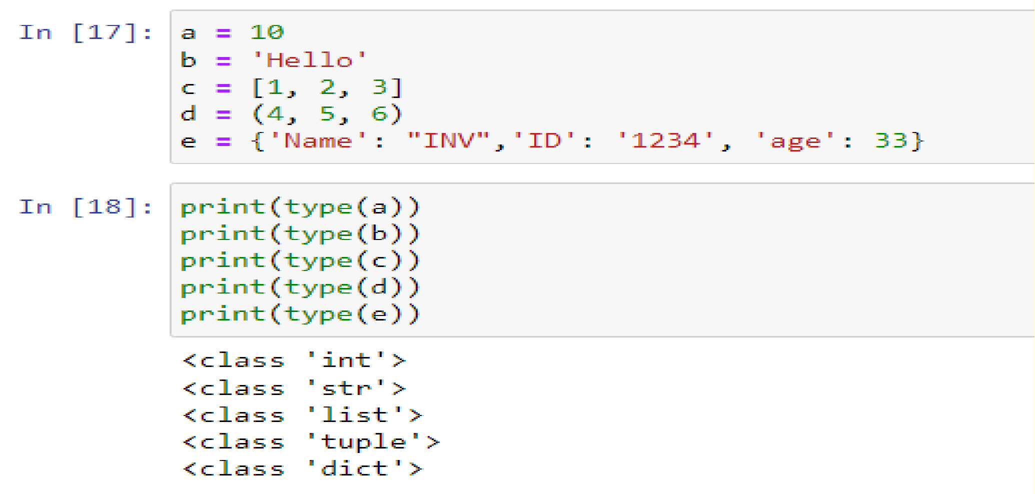
For example:



```
In [15]: print(ord('b'))

         98

In [16]: print(ord('B'))

         66
```

The **type()** function in Python is used to get the data type of a variable or an object. It takes a single argument and returns the data type of that argument.

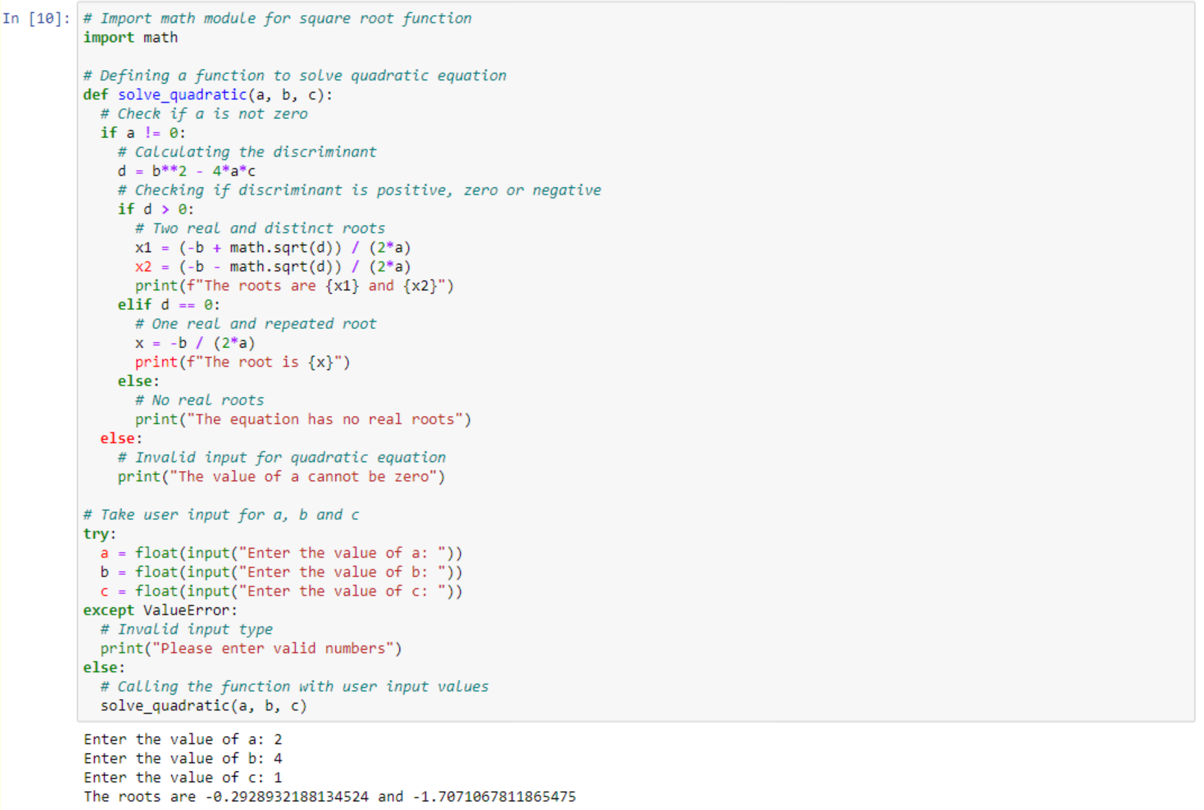
For example:



```
In [17]: a = 10
         b = 'Hello'
         c = [1, 2, 3]
         d = (4, 5, 6)
         e = {'Name': "INV",'ID': '1234', 'age': 33}

In [18]: print(type(a))
         print(type(b))
         print(type(c))
         print(type(d))
         print(type(e))

         <class 'int'>
         <class 'str'>
         <class 'list'>
         <class 'tuple'>
         <class 'dict'>
```

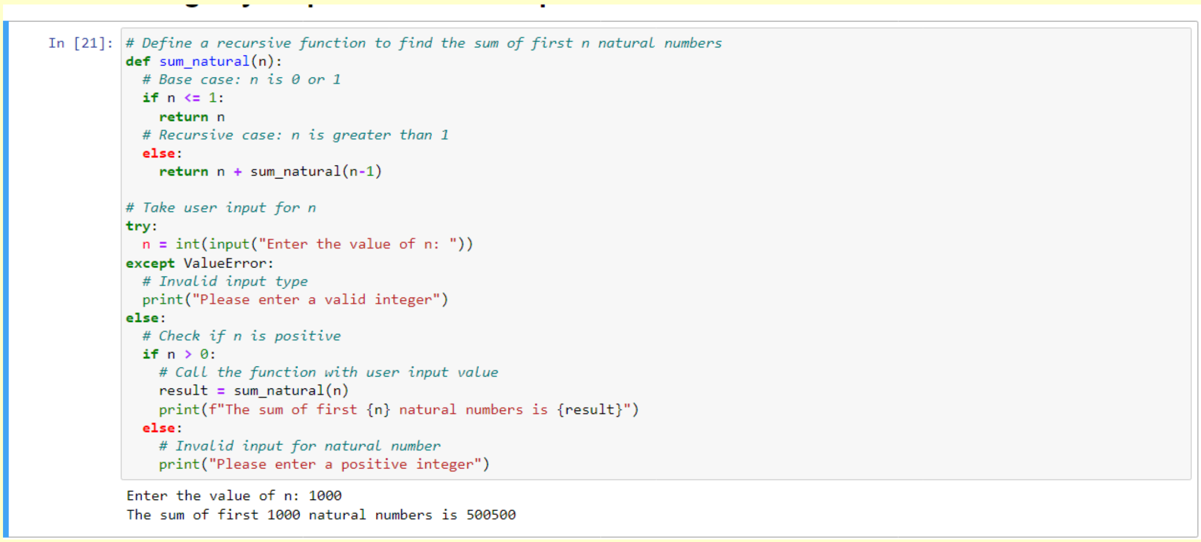
Q14 and Q15 are programming questions. Answer them in Jupyter Notebook.

14. Write a python program to solve a quadratic equation of the form ax^2+bx+c=0. Where a, b and c are to be taken by user input. Handle the erroneous input, such as ‘a’ should not be equal to 0.



```
In [10]: # Import math module for square root function
         import math

         # Defining a function to solve quadratic equation
         def solve_quadratic(a, b, c):
           # Check if a is not zero
           if a != 0:
             # Calculating the discriminant
             d = b**2 - 4*a*c
             # Checking if discriminant is positive, zero or negative
             if d > 0:
               # Two real and distinct roots
               x1 = (-b + math.sqrt(d)) / (2*a)
               x2 = (-b - math.sqrt(d)) / (2*a)
               print(f"The roots are {x1} and {x2}")
             elif d == 0:
               # One real and repeated root
               x = -b / (2*a)
               print(f"The root is {x}")
             else:
               # No real roots
               print("The equation has no real roots")
           else:
             # Invalid input for quadratic equation
             print("The value of a cannot be zero")

         # Take user input for a, b and c
         try:
           a = float(input("Enter the value of a: "))
           b = float(input("Enter the value of b: "))
           c = float(input("Enter the value of c: "))
         except ValueError:
           # Invalid input type
           print("Please enter valid numbers")
         else:
           # Calling the function with user input values
           solve_quadratic(a, b, c)

         Enter the value of a: 2
         Enter the value of b: 4
         Enter the value of c: 1
         The roots are -0.2928932188134524 and -1.7071067811865475
```

15. Write a python program to find the sum of first ‘n’ natural numbers without using any loop. Ask users to input the value of ‘n’.



```
In [21]: # Define a recursive function to find the sum of first n natural numbers
         def sum_natural(n):
           # Base case: n is 0 or 1
           if n <= 1:
             return n
           # Recursive case: n is greater than 1
           else:
             return n + sum_natural(n-1)

         # Take user input for n
         try:
           n = int(input("Enter the value of n: "))
         except ValueError:
           # Invalid input type
           print("Please enter a valid integer")
         else:
           # Check if n is positive
           if n > 0:
             # Call the function with user input value
             result = sum_natural(n)
             print(f"The sum of first {n} natural numbers is {result}")
           else:
             # Invalid input for natural number
             print("Please enter a positive integer")

         Enter the value of n: 1000
         The sum of first 1000 natural numbers is 500500
```