

PYTHON – WORKSHEET 1

Q1 to Q8 have only one correct answer. Choose the correct option to answer your question.

1. Which of the following operators is used to calculate remainder in a division?

A) # B) & C) % D) \$

Ans : C) %

% ,Modulus operation : returns the remainder when 1st operand is divides by the second.

```
In [1]: # Example:
a = 56%4
b = 754%21
print(a,b)

0 19
```

2. In python 2//3 is equal to?

A) 0.6666 B) 0 C) 1 D) 0.67

Ans : B) 0

Floor Division("/"): The division of operands where the result is the quotient in which the digits after the decimal point are removed.

But if one of the operands is negative, the result is floored , i.e., rounded away from zero (towards negative infinity).

Mathematically it is represented :

x/y = math.floor(x/y) x = 2, y =3 : 2/3 = 0.666

Since , x/y is positive thus, x/y = 0

```
In [2]: 2//3

0
```

3. In python, 6<<2 is equal to ?

36 B), 10 C) 24 D) 45

Ans : C) 24

x<<y Returns x with the bits shifted to the left by y places (and new bits on the right-hand-side are zeros).

This is the same as multiplying x by 2**y.

i.e. x = 6 , y= 2 ; 6(2**2) = 6(4) = 24

```
In [3]: 6<<2

24
```

4. In python, 6&2 will give which of the following as output ?

2 B) True C) False D) 0

Ans: A) 2

'&' performs bitwise AND. Results 1 if both operand bits are 1 , else 0. [0&0 = 0; 0&1=1; 1&1=1; 1&0=0]

6 binary value = 110 2 binary value = 010 6&2 = 010 Integer value of (010) = 2

```
In [4]: 6&2

2
```

5. In python 6|2 will give which of the following as output ?

A) 2 B) 4 C) 0 D) 6

Ans : D) 6

'|' Bitwise OR operator. Results 1 if any one operand is 1 , else 0. [0|0 = 0; 0|1=1; 1|1=1; 1|0=1]

6 binary value = 110 2 binary value = 010 6|2 = 110 Integer value of 110 = 6

```
In [5]: 6|2

6
```

6. What does the finally keywords denotes in python ?

A)It is used to mark the end of the code
B)It encloses the lines of code which will be executed if any error occurs while executing the lines of code in the try block.
C)The finally block will be executed no matter if the try block raises an error or not
D)None of the above

Ans : C) The finally block will be executed no matter if the try block raises an error or not.

The finally keyword is used in try...except blocks. It defines a block of code to run when the try...except...else block is final.

The finally block will be executed no matter if the try block raises an error or not.

This can be useful to close objects and clean up resources.

7. What does the raise keyword is used for in python ?

A)It is used to raise exception
B)It is used to define lambda function
C)It's not a keyword in python
D)None of the above

Ans : A) It is used to raise exception.

raise Keyword is used to raise exceptions or errors.

The raise keyword raises an error and stops the control flow of the program.

It is used to bring up the current exception in an exception handler so that it can be handled further up the call stack.

8. Which of the following is common use case of yield keyword in python ?

A)In defining an iterator
B)While defining a lambda function
C)In defining a generator
D)In for loop

Ans : C) In defining a generator

yield keyword is used to create a generator function.

A type of function that is memory efficient and can be used like an iterator object so, instead of storing each number in an array or list and then returning the list, we have to use yield method to store it in an object.
Example- Fibonacci Series.

Q9 and Q10 have multiple correct answers. Choose all the correct options to answer your question.

9. Which of the following are the valid variable names ?

A) _abc B) labc C) abc2 D) None of the above

Ans: A) _abc and C) abc2

Rules for Python variables:

A variable name must start with a letter or the underscore character

A variable name cannot start with a number

A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)

Variable names are case-sensitive (age, Age and AGE are three different variables)

10. Which of the following are the keywords in the python ?

A)yield B)raise C)Look-in D)All of the above

Ans : A) yield and B) raise

Yield : yield keyword is used to create a generator function Raise : The raise keyword raises an error and stops the control flow of the program

```
In [6]: help("keywords")

Here is a list of the Python keywords. Enter any keyword to get more help.

False      break       for         not
None       class      from        or
True        continue  global      pass
__peg_parser__ def        if          raise
and         del        import      return
as          elif       in          try
assert      else       is          while
async       except    lambda      with
await       finally   nonlocal    yield
```

11. Write a python program to find the factorial of a number.

Factorial is a non-negative integer. It is the product of all positive integers less than or equal to that number you ask for factorial. It is denoted by !

Expected output example : 5! = 1x2x3x4x5 = 120

Although, there are multiple approaches available to code for a factorial no :

1. Iterative method : uses for loop and if-else.
2. Recursive method : method that calls itself.
3. Built in function : Python provides a built in factorial() mthod. under math module.

Making use of Python built in function : to write a code to find the factorial of a number.

We have imported the math module that has factorial() function.

It takes an integer number to calculate the factorial.

Input : 5

Output : 120

```
In [7]: import math #import math library
global n # creating global variable n

n = int(input("Enter a number to fetch its factorial :")) #defining n integer

def fact(n): # defining a function fact to calculate factorial

    return(print("\n Factorial of ", n , "is :", math.factorial(n))) # using math.factorial() to calculate factorial

fact(n)

## To continue the code for different input
global c # creating global variable c

c = str(input("\nType n to End else enter a number :")) # defining c string

def new(c):
    if c.isdigit(): # checking if user input is number
        n =int(c) # if yes, then assigning the number to n
        fact(n) # applying fact function
        return new(str(input("\n Enter any key to End else enter a number :"))) #recursive new
    else:
        return print("END") # if string c is not number

new(c)

Enter a number to fetch its factorial :5

Factorial of 5 is : 120

Type n to End else enter a number :15

Factorial of 15 is : 1307674368000

Enter any key to End else enter a number :n
END
```

12. Write a python program to find whether a number is prime or composite.

Input: n=11 Output: 11 is prime

Input: n=10 Output: 10 is composite

A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. The first few prime numbers are {2, 3, 5, 7, 11,}.

Again, we have iterative, recursive methods to solve this problem. But, I am using Math module to solve the problem.

METHOD-USING MATH

APPROACH:

The code implements a basic approach to check if a number is prime or not, by traversing all the numbers from 2 to sqrt(n)+1 and checking if n is divisible by any of those numbers.

ALGORITHM:

1.Check if the given number n is less than or equal to 1, if yes, return False. 2.Traverse all numbers in the range of 2 to sqrt(n)+1. 3.Check if n is divisible by any of the numbers from 2 to sqrt(n)+1, if yes, return False. 4.If n is not divisible by any of the numbers from 2 to sqrt(n)+1, return True.

```
In [8]: import math #import math library
global n # creating global variable n
n = int(input("Enter the number:")) #defining n integer

def is_prime(n):
    if n <= 1: # if n is 1 or less than 1 then , composite
        return (print('The entered number : ', n , 'is Composite'))
    for i in range(2, int(math.sqrt(n)) + 1): # Traverse all numbers in the range of 2 to sqrt(n)+1.
        if n % i == 0: # Checking if n is divisible by 2 through sqrt(n)+1
            return (print('The entered number : ', n , 'is Composite')) #if divisible then return composite)
    return (print('The entered number : ', n , 'is Prime')) # not divisible , return Prime.

is_prime(n)

## To continue the code for different input
global c # creating global variable c
c = str(input("\nType n to End else enter a number :")) # defining c string

def new(c):
    if c.isdigit():
        n =int(c)
        is_prime(n)
        return new(str(input("\n Enter any key to End else enter a number :")))
    else:
        return print("END")

new(c)

Enter the number:342547389
The entered number : 342547389 is Composite

Type n to End else enter a number :8364321
The entered number : 8364321 is Composite

Enter any key to End else enter a number :171
The entered number : 171 is Composite

Enter any key to End else enter a number :161
The entered number : 161 is Composite

Enter any key to End else enter a number :11
The entered number : 11 is Prime

Enter any key to End else enter a number :n
END
```

13. Write a python program to check whether a given string is palindrome or not.

Given a string, write a python function to check if it is palindrome or not. A string is said to be a palindrome if the reverse of the string is the same as the string. For example, "radar" is a palindrome, but "radix" is not a palindrome.
Again , many methods available the best method I relate to is using reverse()and extend() function.

Algorithm:

Step 1: read input s Step 2: Convert the string s into list x Step 3: Create an empty list y and copy the items of list x using extend() Step 4: Reverse the list x using reverse() Step 5: Check if x == y for a palindrome.

```
In [9]: global s
s = str(input("\nEnter the string:"))
def isPalindrome(s):
    x=list(s)
    y=[]
    y.extend(x)
    x.reverse()
    if(x==y):
        return (print('\nEntered string :',s,'is Palindrome'))
    return (print('\nEntered string :',s,'is not Palindrome'))

isPalindrome(s)

global c

c = str(input("\nType n to END else Enter the string :"))
def new(c):
    if c == 'n':
        return (print("END"))

    else:
        isPalindrome(c)

        return new(str(input("\nType n to END else Enter the string:")))

new(c)

Enter the string:MADAM

Entered string : MADAM is Palindrome

Type n to END else Enter the string :POLITE

Entered string : POLITE is not Palindrome

Type n to END else Enter the string:n
END
```

14. Write a Python program to get the third side of right-angled triangle from two given sides.

Pythagoras theorem states that the square of hypotenuse of a right angled triangle is equal to the sum of squares of the other two sides.

Formula : h = (((side1 side1) + (side2 side2))**(1/2))

```
In [11]: global side1 , side2

print("Note : The 2 values to be entered should be separated by a space")

side1,side2 = [int(x) for x in input("Enter a value for base and perpendicular of the right angled triangle ").split()]

def thirdSide(side1,side2):
    h = (((side1 * side1) + (side2 * side2))**(1/2));
    return (print("\nFor entered base :",side1, "and perpendicular :",
    side2, "\n\nThe Hypotenuse of the right angled triangle is :",h));

thirdSide(side1,side2)

global c

c = str(input("\nType y to enter input / n to end:"))

def new(c):
    if c == 'y':
        side1,side2 = [int(x) for x in input("Enter a value for base and perpendicular of the right angled triangle ").split()]
        thirdSide(side1,side2)
        return new(str(input("\n\nType y to enter input / any key to end:")))
    elif c=='n':
        return(print("END"))
    else:
        return new(str(input("Enter a valid choice y/n :")))

new(c)

Note : The 2 values to be entered should be separated by a space
Enter a value for base and perpendicular of the right angled triangle 2 3

For entered base : 2 and perpendicular : 3

The Hypotenuse of the right angled triangle is : 3.605551275463989

Type y to enter input / n to end:y
Enter a value for base and perpendicular of the right angled triangle 12 5

For entered base : 12 and perpendicular : 5

The Hypotenuse of the right angled triangle is : 13.0

Type y to enter input / any key to end:n
END
```

15 Write a python program to print the frequency of each of the characters present in a given string.

Given a string, the task is to find the frequencies of all the characters in that string and return a dictionary with key as the character and its value as its frequency in the given string.

There are n number of ways available on internet to try out, the most basic is : to iterate through string and form a key in dictionary of newly occurred element or if element is already occurred, increase its value by 1.
But, again leverage of using Python : it provides a method Counter under collections library which makes the entire process a lot easier.

This method could be used to find all occurrences is this function, which actually gets all element frequencies and could also be used to print single element frequencies if required.

```
In [12]: # collections.Counter()
from collections import Counter

# initializing string
global string
string = str(input("Enter the string: "))

# using collections.Counter() to get
# count of each element in string and printing the result

print("Count of all characters in", string, "is :\n\n" + str(Counter(string)))

global c

c = str(input("\nType n to END else Enter the string :"))
def new(c):
    if c == 'n':
        return (print("END"))

    else:
        string = c

        print("Count of all characters in", string, "is :\n\n" + str(Counter(string)))

        return new(str(input("\nType n to END else Enter the string:")))

new(c)

Enter the string: FLAMBOYANT
Count of all characters in FLAMBOYANT is :

Counter({'A': 2, 'F': 1, 'L': 1, 'M': 1, 'B': 1, 'O': 1, 'Y': 1, 'N': 1, 'T': 1})

Type n to END else Enter the string :LULLABY
Count of all characters in LULLABY is :

Counter({'L': 3, 'U': 1, 'A': 1, 'B': 1, 'Y': 1})

Type n to END else Enter the string:n
END
```