**Python MCQ 🡺 Python MCQ - geeksforgeeks.org**

<https://www.geeksforgeeks.org/functions-python-gq/>

Python Functions =>

**1. What will be the output of the following code :**

Print(type(type(int)))

A. type 'int'

B. type 'type'

C. Error

D. 0

**Answer:** B. type 'type' =>> <class ‘type’>

2. What is the output of the following code :

L = ['a','b','c','d']

print "".join(L)

A. Error

B. None

C. abcd

D. [‘a’,’b’,’c’,’d’]

**Answer:**

3. What is the output of the following segment :

chr(ord('A'))

A. A

B. B

C. a

D. Error

**Answer:** A. A

4. What is the output of the following program :

y = 8

z = lambda x : x \* y

print(z(6))

A. 48

B. 14

C. 64

D. None of the above

**Answer:** A. 48

5. What is called when a function is defined inside a class?

A. Module

B. Class

C. Another Function

D. Method

**Answer:** D. Method

6. Which of the following is the use of id() function in python?

A. Id returns the identity of the object

B. Every object doesn’t have a unique id

C. All of the mentioned

D. None of the mentioned

**Answer:**

7. What is the output of the following program :

import re

sentence = 'horses are fast'

regex = re.compile('(?P<animal>w+) (?P<verb>w+) (?P<adjective>w+)')

matched = re.search(regex, sentence)

print(matched.groupdict())

A. {‘animal’: ‘horses’, ‘verb’: ‘are’, ‘adjective’: ‘fast’}

B. (‘horses’, ‘are’, ‘fast’)

C. ‘horses are fast’

D. ‘are’

**Answer:**

8. Suppose list1 is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after list1.pop(1)?

A. [3, 4, 5, 20, 5, 25, 1, 3]

B. [1, 3, 3, 4, 5, 5, 20, 25]

C. [3, 5, 20, 5, 25, 1, 3]

D. [1, 3, 4, 5, 20, 5, 25]

**Answer:**

9. time.time() returns \_\_\_\_\_\_\_\_

A. the current time

B. the current time in milliseconds

C. the current time in milliseconds since midnight

D. the current time in milliseconds since midnight, January 1, 1970

E. the current time in milliseconds since midnight, January 1, 1970 GMT (the Unix time)

**Answer:**

10. Consider the results of a medical experiment that aims to predict whether someone is going to develop myopia based on some physical measurements and heredity. In this case, the input dataset consists of the person’s medical characteristics and the target variable is binary: 1 for those who are likely to develop myopia and 0 for those who aren’t. This can be best classified as

A. Regression

B. Decision Tree

C. Clustering

D. Association Rules

**Answer:**

Python Operators =>

1. What is the output of the following code :

print 9//2

A. 4.5

B. 4.0

C. 4

D. Error

**Answer:**

2. Which function overloads the >> operator?

A. more()

B. gt()

C. ge()

D. None of the above

**Answer:**

3. Which operator is overloaded by the or() function?

A. ||

B. |

C. //

D. /

**Answer:**

4. What is the output of the following program :

i = 0

while i < 3:

print i

i++

print i+1

A. 0 2 1 3 2 4

B. 0 1 2 3 4 5

C. Error

D. 1 0 2 4 3 5

**Answer:**

Python Miscellaneous =>

1. What is the output of the following program :

print "Hello World"[::-1]

A. dlroW olleH

B. Hello Worl

C. d

D. Error

**Answer:**

2. Given a function that does not return any value, what value is shown when executed at the shell?

A. int

B. bool

C. void

D. None

**Answer:**

Question 3. Which module in Python supports regular expressions?

A. re

B. regex

C. pyregex

D. None of the above

**Answer:**

4. What is the output of the following program :

print 0.1 + 0.2 == 0.3

A. True

B. False

C. Machine dependent

D. Error

**Answer:**

Question 5. Which of the following is not a complex number?

A. k = 2 + 3j

B. k = complex(2, 3)

C. k = 2 + 3l

D. k = 2 + 3J

**Answer:**

6. What does ~~~~~~5 evaluate to?

A. +5

B. -11

C. +11

D. -5

**Answer:**

Question 7. Given a string s = “Welcome”, which of the following code is incorrect?

A. print s[0]

B. print s.lower()

C. s[1] = ‘r’

D. print s.strip()

**Answer:**

8. \_\_\_\_\_\_\_\_ is a simple but incomplete version of a function.

A. Stub

B. Function

C. A function developed using bottom-up approach

D. A function developed using top-down approach

**Answer:**

Question 9. To start Python from the command prompt, use the command \_\_\_\_\_\_

A. execute python

B. go python

C. python

D. run python

**Answer:**

10. Which of the following is correct about Python?

A. It supports automatic garbage collection.

B. It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java

C. Both of the above

D. None of the above

**Answer:**

Python Data Type =>

1. Which of these is not a core data type?

A. Lists

B. Dictionary

C. Tuples

D. Class

**Answer:**

Question 2. What data type is the object below ? L = [1, 23, ‘hello’, 1]

A. List

B. Dictionary

C. Tuple

D. Array

**Answer:**

3. Which of the following function convert a string to a float in python?

A. int(x [,base])

B. long(x [,base] )

C. float(x)

D. str(x)

**Answer:**

4. Which of the following statement(s) is TRUE?

1. A hash function takes a message of arbitrary length and generates a fixed length code.

2. A hash function takes a message of fixed length and generates a code of variable length.

3. A hash function may give the same hash value for distinct messages.

A. I only

B. II and III only

C. I and III only

D. II only

**Answer:**

Python Output Type =>

1. What is the output of the following program :

def myfunc(a):

a = a + 2

a = a \* 2

return a

print myfunc(2)

A. 8 B. 16 C. Indentation Error D. Runtime Error

**Answer:**

2. What is the output of the expression : 3\*1\*\*3

A. 27 B. 9 C. 3 D. 1

**Answer:**

Question 3. What is the output of the following program :

print '{0:.2}'.format(1.0 / 3)

A. 0.333333

B. 0.33

C. 0.333333:-2

D. Error

**Answer:**

4. What is the output of the following program :

print '{0:-2%}'.format(1.0 / 3)

A. 0.33 B. 0.33% C. 33.33% D. 33%

**Answer:**

5. What is the output of the following program :

i = 0

while i < 3:

print i

i += 1

else:

print 0

A. 0 1 2 3 0

B. 0 1 2 0

C. 0 1 2

D. Error

**Answer:**

6. What is the output of the following program :

i = 0

while i < 5:

print(i)

i += 1

if i == 3:

break

else:

print(0)

A. 0 1 2 0

B. 0 1 2

C. Error

D. None of the above

**Answer:**

7. What is the output of the following program :

print 'cd'.partition('cd')

A. (‘cd’)

B. (”)

C. (‘cd’, ”, ”)

D. (”, ‘cd’, ”)

**Answer:**

8. What is the output of the following program :

print 'abef'.partition('cd')

A. (‘abef’)

B. (‘abef’, ‘cd’, ”)

C. (‘abef’, ”, ”)

D. Error

**Answer:**

9. What is the output of the following program :

print 'abcefd'.replace('cd', '12')

A. ab1ef2

B. abcefd

C. ab1efd

D. ab12ed2

**Answer:**

10. What will be displayed by the following code?

def f(value, values):

v = 1

values[0] = 44

t = 3

v = [1, 2, 3]

f(t, v)

print(t, v[0])

A. 1 1

B. 1 44

C. 3 1

D. 3 44

**Answer:**

[Output of Python Program | Set 1](https://www.geeksforgeeks.org/output-python-program-set-1/) =>>

1:

r = lambda q: q \* 2

s = lambda q: q \* 3

x = 2

x = r(x)

x = s(x)

x = r(x)

print x

Output: 24

Explanation : In the above program r and s are lambda functions or anonymous functions and q is the argument to both of the functions. In first step we have initialized x to 2. In second step we have passed x as argument to the lambda function r, this will return x\*2 which is stored in x. That is, x = 4 now. Similarly in third step we have passed x to lambda function s, So x = 4\*3. i.e, x = 12 now. Again in the last step, x is multiplied by 2 by passing it to function r. Therefore, x = 24.

2:

a = 4.5

b = 2

print a//b

Output: 2.0

Explanation : This type of division is called truncating division where the remainder is truncated or dropped.

3:

a = True

b = False

c = False

if a or b and c:

print "GEEKSFORGEEKS"

else:

print "geeksforgeeks"

Output: GEEKSFORGEEKS

Explanation : In Python, AND operator has higher precedence than OR operator. So, it is evaluated first. i.e, (b and c) evaluates to false.Now OR operator is evaluated. Here, (True or False) evaluates to True. So the if condition becomes True and GEEKSFORGEEKS is printed as output.

4:

a = True

b = False

c = False

if not a or b:

print 1

elif not a or not b and c:

print 2

elif not a or b or not b and a:

print 3

else:

print 4

Output: 3

Explanation: In Python the precedence order is first NOT then AND and in last OR. So the if condition and second elif condition evaluates to False while third elif condition is evaluated to be True resulting in 3 as output.

5:

count = 1

def doThis():

global count

for i in (1, 2, 3):

count += 1

doThis()

print count

Output: 4

Explanation: The variable count declared outside the function is global variable and also the count variable being referenced in the function is the same global variable defined outside of the function. So, the changes made to variable in the function is reflected to the original variable. So, the output of the program is 4.

[Output of python program | Set 2](https://www.geeksforgeeks.org/output-python-program-set-2/) =>>

1:

class Acc:

def \_\_init\_\_(self, id):

self.id = id

id = 555

acc = Acc(111)

print acc.id

Output: 111

Explanation: Instantiation of the class “Acc” automatically calls the method \_\_init\_\_ and passes the object as the self parameter. 111 is assigned to data attribute of the object called id.

The value “555” is not retained in the object as it is not assigned to a data attribute of the class/object. So, the output of the program is “111”

2:

for i in range(2):

print i

for i in range(4,6):

print i

Output:

0

1

4

5

Explanation: If only single argument is passed to the range method, Python considers this argument as the end of the range and the default start value of range is 0. So, it will print all the numbers starting from 0 and before the supplied argument.

For the second for loop the starting value is explicitly supplied as 4 and ending is 5.

3:

values = [1, 2, 3, 4]

numbers = set(values)

def checknums(num):

if num in numbers:

return True

else:

return False

for i in filter(checknums, values):

print i

Output:

1

2

3

4

Explanation: The function “filter” will return all items from list values which return True when passed to the function “checkit”. “checkit” will check if the value is in the set. Since all the numbers in the set come from the values list, all of the original values in the list will return True.

4:

counter = {}

def addToCounter(country):

if country in counter:

counter[country] += 1

else:

counter[country] = 1

addToCounter('China')

addToCounter('Japan')

addToCounter('china')

print len(counter)

Output: 3

Explanation: The task of “len” function is to return number of keys in a dictionary. Here 3 keys are added to the dictionary “country” using the “addToCounter” function.

Please note carefully – The keys to a dictionary are case sensitive.

[Output of python program | Set 3](https://www.geeksforgeeks.org/output-python-program-set-3/) =>>

1.

class Geeks:

def \_\_init\_\_(self, id):

self.id = id

manager = Geeks(100)

manager.\_\_dict\_\_['life'] = 49

print manager.life + len(manager.\_\_dict\_\_)

Output: 51

Explanation : In the above program we are creating a member variable having name ‘life’ by adding it directly to the dictionary of the object ‘manager’ of class ‘Geeks’. Total numbers of items in the dictionary is 2, the variables ‘life’ and ‘id’. Therefore the size or the length of the dictionary is 2 and the variable ‘life’ is assigned a value ’49’. So the sum of the variable ‘life’ and the size of the dictionary is 49 + 2 = 51.

2.

a = "GeeksforGeeks "

b = 13

print a + b

Output: An error is shown.

Explanation : As you can see the variable ‘b’ is of type integer and the variable ‘a’ is of type string. Also as Python is a strongly typed language we cannot simply concatenate an integer with a string. We have to first convert the integer variable to the type string to concatenate it with a string variable. So, trying to concatenate an integer variable to a string variable, an exception of type “TypeError” is occurred.

3.

dictionary = {}

dictionary[1] = 1

dictionary['1'] = 2

dictionary[1] += 1

sum = 0

for k in dictionary:

sum += dictionary[k]

print sum

Output: 4

Explanation : In the above dictionary, the key 1 enclosed between single quotes and only 1 represents two different keys as one of them is integer and other is string. So, the output of the program is 4.

4.

dictionary = {1:'1', 2:'2', 3:'3'}

del dictionary[1]

dictionary[1] = '10'

del dictionary[2]

print len(dictionary)

Output: 2

Explanation : The task of the ‘del’ function is to remove key-value pairs from a dictionary. Initially the size of the given dictionary was 3. Then the key value pair for key 1 is first removed and then added back with a new value. Then the key value pair for key 2 is removed. So, finally the size of the dictionary is 2.

[Output of python program | Set 4](https://www.geeksforgeeks.org/output-python-program-set-4/) =>>

1.

nameList = ['Harsh', 'Pratik', 'Bob', 'Dhruv']

print nameList[1][-1]

Output: k

Explanation:

The index position -1 represents either the last element in a list or the last character in a String. In the above given list of names “nameList”, the index 1 represents the second element i.e, the second string “Pratik” and the index -1 represents the last character in the string “Pratik”. So, the output is “k”.

2.

nameList = ['Harsh', 'Pratik', 'Bob', 'Dhruv']

pos = nameList.index("GeeksforGeeks")

print pos \* 5

Output: An Exception is thrown, ValueError: 'GeeksforGeeks' is not in list

Explanation:

The task of the index is to find the position of a supplied value in a given list. In the above program the supplied value is “GeeksforGeeks” and the list is nameList. As GeeksforGeeks is not present in the list, an exception is thrown.

3.

geekCodes = [1, 2, 3, 4]

# List will look like as [1,2,3,4,[5,6,7,8]]

geekCodes.append([5,6,7,8])

print len(geekCodes)

print(geekCodes)

#new list will be appended at the index 4 of geekCodes.

Output:

5

[1,2,3,4,[5,6,7,8]]

Explanation: The task of append() method is to append a passed obj into an existing list. But instead of passing a list to the append method will not merge the two lists, the entire list which is passed is added as an element of the list. So the output is 5.

4.

def addToList(listcontainer):

listcontainer += [10]

mylistContainer = [10, 20, 30, 40]

addToList(mylistContainer)

print len(mylistContainer)

Output: 5

Explanation:

In Python, everything is a reference, and references are passed by value. Parameter passing in Python is the same as reference passing in Java. As a consequence, the function can modify the value referred by passed argument, i.e. the value of the variable in the caller’s scope can be changed. Here the task of the function “addToList” is to add an element 10 in the list, So this will increase the length of the list by 1. So the output of the program is 5.

[Output of python program | Set 5](https://www.geeksforgeeks.org/output-python-program-set-5/) =>>

1.

def gfgFunction():

"Geeksforgeeks is cool website for boosting up technical skills"

return 1

print gfgFunction.\_\_doc\_\_[17:21]

Output: cool

Explanation:

There is a docstring defined for this method, by putting a string on the first line after the start of the function definition. The docstring can be referenced using the \_\_doc\_\_ attribute of the function.

And hence it prints the indexed string.

2.

class A(object):

val = 1

class B(A):

pass

class C(A):

pass

print A.val, B.val, C.val

B.val = 2

print A.val, B.val, C.val

A.val = 3

print A.val, B.val, C.val

Output:

1 1 1

1 2 1

3 2 3

Explanation:

In Python, class variables are internally handled as dictionaries. If a variable name is not found in the dictionary of the current class, the class hierarchy (i.e., its parent classes) are searched until the referenced variable name is found, if the variable is not found error is being thrown.

So, in the above program the first call to print() prints the initialized value i.e, 1.

In the second call since B. val is set to 2, the output is 1 2 1.

The last output 3 2 3 may be surprising. Instead of 3 3 3, here B.val reflects 2 instead of 3 since it is overridden earlier.

3.

check1 = ['Learn', 'Quiz', 'Practice', 'Contribute']

check2 = check1

check3 = check1[:]

check2[0] = 'Code'

check3[1] = 'Mcq'

count = 0

for c in (check1, check2, check3):

if c[0] == 'Code':

count += 1

if c[1] == 'Mcq':

count += 10

print count

Output: 12

Explanation:

When assigning check1 to check2, we create a second reference to the same list. Changes to check2 affects check1. When assigning the slice of all elements in check1 to check3, we are creating a full copy of check1 which can be modified independently (i.e, any change in check3 will not affect check1).

So, while checking check1 ‘Code’ gets matched and count increases to 1, but Mcq doest gets matched since its available only in check3.

Now checking check2 here also ‘Code’ gets matched resulting in count value to 2.

Finally while checking check3 which is separate than both check1 and check2 here only Mcq gets matched and count becomes 12.

4.

def gfg(x,l=[]):

for i in range(x):

l.append(i\*i)

print(l)

gfg(2)

gfg(3,[3,2,1])

gfg(3)

Output:

[0, 1]

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

[0, 1, 0, 1, 4]

Explanation:

The first function call should be fairly obvious, the loop appends 0 and then 1 to the empty list, l. l is a name for a variable that points to a list stored in memory. The second call starts off by creating a new list in a new block of memory. l then refers to this new list. It then appends 0, 1 and 4 to this new list. So that’s great. The third function call is the weird one. It uses the original list stored in the original memory block. That is why it starts off with 0 and 1.

[Output of python program | Set 6(Lists)](https://www.geeksforgeeks.org/output-python-program-set-6/) =>>

1.

list1 = ['physics', 'chemistry', 1997, 2000]

list2 = [1, 2, 3, 4, 5, 6, 7 ]

print "list1[0]: ", list1[0] #statement 1

print "list1[0]: ", list1[-2] #statement 2

print "list1[-2]: ", list1[1:] #statement 3

print "list2[1:5]: ", list2[1:5] #statement 4

Output:

list1[0]: physics

list1[0]: 1997

list1[-2]: ['chemistry', 1997, 2000]

list2[1:5]: [2, 3, 4, 5]

Explanation:

To access values in lists, we use the square brackets for slicing along with the index or indices to obtain required value available at that index.For N items in a List MAX value of index will be N-1.

Statement 1 : This will print item located at index 0 in Output.

Statement 2 : This will print item located at index -2 i.e.second last element in Output.

Statement 3 : This will print items located from index 1 to end of the list.

Statement 4 : This will print items located from index 1 to 4 of the list.

2.

list1 = ['physics', 'chemistry', 1997, 2000]

print "list1[1][1]: ", list1[1][1] #statement 1

print "list1[1][-1]: ", list1[1][-1] #statement 2

Output:

list1[1][1]: h

list1[1][-1]: y

Explanation:

In python we can slice a list but we can also slice a element within list if it is a string. The declaration list[x][y] will mean that ‘x’ is the index of element within a list and ‘y’ is the index of entity within that string.

3.

list1 = [1998, 2002, 1997, 2000]

list2 = [2014, 2016, 1996, 2009]

print "list1 + list 2 = : ", list1 + list2 #statement 1

print "list1 \* 2 = : ", list1 \* 2 #statement 2

Output:

list1 + list 2 = : [1998, 2002, 1997, 2000, 2014, 2016, 1996, 2009]

list1 \* 2 = : [1998, 2002, 1997, 2000, 1998, 2002, 1997, 2000]

Explanation:

When addition(+) operator uses list as its operands then the two lists will get concatenated. And when a list id multiplied with a constant k>=0 then the same list is appended k times in the original list.

4.

list1 = range(100, 110) #statement 1

print "index of element 105 is : ", list1.index(105) #statement 2

Output:

index of element 105 is : 5

Explanation:

Statement 1 : will genetrate numbers from 100 to 110 and appent all these numbers in the list.

Statement 2 : will give the index value of 105 in the list list1.

5.

list1 = [1, 2, 3, 4, 5]

list2 = list1

list2[0] = 0;

print "list1= : ", list1 #statement 2

Output:

list1= : [0, 2, 3, 4, 5]

Explanation:

In this problem, we have provided a reference to the list1 with another name list2 but these two lists are same which have two references(list1 and list2). So any alteration with list2 will affect the original list.

[Output of python program | Set 7](https://www.geeksforgeeks.org/output-python-programs-set-7/) =>>

1.

var1 = 'Hello Geeks!'

var2 = "GeeksforGeeks"

print "var1[0]: ", var1[0] # statement 1

print "var2[1:5]: ", var2[1:5] # statement 2

Output:

var1[0]: H

var2[1:5]: eeks

Explanation:

Strings are among the most popular types in Python. We can create a string by enclosing characters within quotes. Python treats single quotes same as double quotes. It is notable that unlike C or C++ python does not support a character type; in fact single characters are treated as strings of length one, thus also considered a substring. To access substrings, use the square brackets for slicing along with the index or indices to obtain your substring.

Statement 1: will simply put character at 0 index on the output screen.

Statement 2: will put the character starting from 0 index to index 4.

2.

var1 = 'Geeks'

print "Original String :-", var1

print "Updated String :- ", var1[:5] + 'for' + 'Geeks' # statement 1

Output:

Original String :- Geeks

Updated String :- GeeksforGeeks

Explanation:  
Python provides a flexible way to update string in your code. Use square brackets and specify the index from where string has to be updated and use + operator to append the string. [x:y] operator is called Range Slice and gives the characters from the given range.

Statement 1: In the given code tells the interpreter that from 5th index of string present in var1 append ‘for’ and ‘Geeks’ to it.

3.

para\_str = """this is a long string that is made up of

several lines and non-printable characters such as

TAB ( \t ) and they will show up that way when displayed.

NEWLINEs within the string, whether explicitly given like

this within the brackets [ \n ], or just a NEWLINE within

the variable assignment will also show up.

"""

print para\_str

Output:

this is a long string that is made up of

several lines and non-printable characters such as

TAB ( ) and they will show up that way when displayed.

NEWLINEs within the string, whether explicitly given like

this within the brackets [

], or just a NEWLINE within

the variable assignment will also show up.

Explanation:

Python’s triple quotes comes to the rescue by allowing strings to span multiple lines, including NEWLINEs, TABs, and any other special characters.The syntax for triple quotes consists of three consecutive single or double quotes.

4.

print 'C:\\inside C directory' # statement1

print r'C:\\inside C directory' # statement2

Output:

C:\inside C directory

C:\\inside C directory

Explanation:

Raw strings do not treat the backslash as special characters at all.

Statement 1 : will print the message while considering backslash as a special character.

Statement 2 : is a raw string that will treat backslash as a normal character.

5.

print '\x25\x26'

Output:

%&

Explanation:

In the above code \x is an escape sequence that means the following 2 digits are a hexadecimal number encoding a character. Hence the corresponding symbols will be on the output screen.

[Output of python program | Set 8](https://www.geeksforgeeks.org/output-python-programs-set-8/) =>>

1.

list = [1, 2, 3, None, (1, 2, 3, 4, 5), ['Geeks', 'for', 'Geeks']]

print len(list)

Output:

6

Explanation:

The beauty of python list datatype is that within a list, a programmer can nest another list, a dictionary or a tuple. Since in the code there are 6 items present in the list the length of the list is 6.

2.

list = ['python', 'learning', '@', 'Geeks', 'for', 'Geeks']

print list[::]

print list[0:6:2]

print list[ :6: ]

print list[ :6:2]

print list[ ::3]

print list[ ::-2]

Output:

['python', 'learning', '@', 'Geeks', 'for', 'Geeks']

['python', '@', 'for']

['python', 'learning', '@', 'Geeks', 'for', 'Geeks']

['python', '@', 'for']

['python', 'Geeks']

['Geeks', 'Geeks', 'learning']

Explanation:

In python list slicing can also be done by using the syntax listName[x:y:z] where x means the initial index, y-1 defines the final index value and z specifies the step size. If anyone of the values among x, y and z is missing the interpreter takes default value.

Note:

1. For x default value is 0 i.e. start of the list.

2. For y default value is length of the list.

3. For z default value is 1 i.e. every element of the list.

3.

d1 = [10, 20, 30, 40, 50]

d2 = [1, 2, 3, 4, 5]

print d1 - d1

Output:

No Output

Explanation:

Unlike additon or relational operators not all the artihmatic operaters can use lists as their operands. Since – minus operator can’t take lists as its operand no output will be produced. Program will produce following error.

TypeError: unsupported operand type(s) for -: 'list' and 'list'

4.

list = ['a', 'b', 'c', 'd', 'e']

print list[10:]

Output:

[]

Explanation:

As one would expect, attempting to access a member of a list using an index that exceeds the number of members (e.g., attempting to access list[10] in the list above) results in an IndexError. However, attempting to access a slice of a list at a starting index that exceeds the number of members in the list will not result in an IndexError and will simply return an empty list.

5.

list = ['a', 'b', 'c']\*-3

print list

Output:

[]

Explanation:

A expression list[listelements]\*N where N is a integer appends N copies of list elements in the original list. If N is a negetive integer or 0 output will be a empty list else if N is positive list elements will be added N times to the original list.

[Output of python program | Set 9(Dictionary)](https://www.geeksforgeeks.org/output-python-programs-set-9-dictionary/) =>>

**1) What is the output of the following program?**

dictionary = {'GFG' : 'geeksforgeeks.org',

'google' : 'google.com',

'facebook' : 'facebook.com'

}

del dictionary['google'];

for key, values in dictionary.items():

print(key)

dictionary.clear();

for key, values in dictionary.items():

print(key)

del dictionary;

for key, values in dictionary.items():

print(key)

a) Both b and d  
b) Runtime error  
c) GFG

facebook  
d) facebook

GFG

**Ans. (a)**  
Output:

facebook

GFG

Explanation: The statement: del dictionary; removes the entire dictionary, so iterating over a deleted dictionary throws a runtime error as follows:

Traceback (most recent call last):

File "cbeac2f0e35485f19ae7c07f6b416e84.py", line 12, in

for key, values in dictionary.items():

NameError: name 'dictionary' is not defined

2) What is the output of the following program?

dictionary1 = {'Google' : 1,

'Facebook' : 2,

'Microsoft' : 3

}

dictionary2 = {'GFG' : 1,

'Microsoft' : 2,

'Youtube' : 3

}

dictionary1.update(dictionary2);

for key, values in dictionary1.items():

print(key, values)

a) Compilation error

b) Runtime error

c) (‘Google’, 1)

(‘Facebook’, 2)

(‘Youtube’, 3)

(‘Microsoft’, 2)

(‘GFG’, 1)

d) None of these

Ans. (c)

Explanation: dictionary1.update(dictionary2) is used to update the entries of dictionary1 with entries of dictionary2. If there are same keys in two dictionaries, then the value in second dictionary is used.

3) What is the output of the following program?

dictionary1 = {'GFG' : 1,

'Google' : 2,

'GFG' : 3

}

print(dictionary1['GFG']);

a) Compilation error due to duplicate keys

b) Runtime time error due to duplicate keys

c) 3

d) 1

Ans. (c)

Explanation: Here, GFG is the duplicate key. Duplicate keys are not allowed in python. If there are same keys in a dictionay, then the value assigned mostly recently is assigned to the that key.

4) What is the output of the following program?

temp = dict()

temp['key1'] = {'key1' : 44, 'key2' : 566}

temp['key2'] = [1, 2, 3, 4]

for (key, values) in temp.items():

print(values, end = "")

a) Compilation error

b) {‘key1’: 44, ‘key2’: 566}[1, 2, 3, 4]

c) Runtime error

d) None of the above

Ans. (b)

Explanation: A dictionary can hold any value such as an integer, string, list or even another dictionary holding key value pairs.

Note: This code can be executed only in python versions above 3

5) What is the output of the following program?

temp = {'GFG' : 1,

'Facebook' : 2,

'Google' : 3

}

for (key, values) in temp.items():

print(key, values, end = " ")

a) Google 3 GFG 1 Facebook 2

b) Facebook 2 GFG 1 Google 3

c) Facebook 2 Google 3 GFG 1

d) Any of the above

Ans. (d)

Explanation: Dictionaries are unordered. So any key value pairs can be added at any location within a dictionary. Any of the output may come.

Note: This code can be executed only in python versions above 3.

[Output of python program | Set 10(Exception Handling)](https://www.geeksforgeeks.org/output-of-python-programs-set-10-exception-handling/) =>>

1) What is the output of the following program?

data = 50

try:

data = data/0

except ZeroDivisionError:

print('Cannot divide by 0 ', end = '')

else:

print('Division successful ', end = '')

try:

data = data/5

except:

print('Inside except block ', end = '')

else:

print('GFG', end = '')

a) Cannot divide by 0 GFG

b) Cannot divide by 0

c) Cannot divide by 0 Inside except block GFG

d) Cannot divide by 0 Inside except block

Ans. (a)

Explanation: The else block of code is executed only when there occurs no exception in try block.

2) What is the output of the following program?

data = 50

try:

data = data/10

except ZeroDivisionError:

print('Cannot divide by 0 ', end = '')

finally:

print('GeeksforGeeks ', end = '')

else:

print('Division successful ', end = '')

a) Runtime error

b) Cannot divide by 0 GeeksforGeeks

c) GeeksforGeeks Division successful

d) GeeksforGeeks

Ans. (a)

Explanation: else block following a finally block is not allowed in python. Python throws syntax error when such format is used.

3) What is the output of the following program?

value = [1, 2, 3, 4]

data = 0

try:

data = value[4]

except IndexError:

print('GFG', end = '')

except:

print('GeeksforGeeks ', end = '')

a) GeeksforGeeks

b) GFG

c) GFG GeeksforGeeks

d) Compilation error

Ans. (b)

Explanation: At a time only one exception is caught, even though the throw exception in the try block is likely to belong to multiple exception type.

4) What is the output of the following program?

value = [1, 2, 3, 4]

data = 0

try:

data = value[3]

except IndexError:

print('GFG IndexError ', end = '')

except:

print('GeeksforGeeks IndexError ', end = '')

finally:

print('Geeks IndexError ', end = '')

data = 10

try:

data = data/0

except ZeroDivisionError:

print('GFG ZeroDivisionError ', end = '')

finally:

print('Geeks ZeroDivisionError ')

a) GFG ZeroDivisionError GFG ZeroDivisionError

b) GFG ZeroDivisionError Geeks ZeroDivisionError

c) Geeks IndexError GFG ZeroDivisionError Geeks ZeroDivisionError

d) Geeks IndexError GFG ZeroDivisionError

Ans. (c)

Explanation: finally block of code is always executed whether the exception occurs or not. If exception occurs, the except block is executed first followed by finally block.

5) What is the output of the following program?

value = [1, 2, 3, 4, 5]

try:

value = value[5]/0

except (IndexError, ZeroDivisionError):

print('GeeksforGeeks ', end = '')

else:

print('GFG ', end = '')

finally:

print('Geeks ', end = '')

a) Compilation error

b) Runtime error

c) GeeksforGeeks GFG Geeks

d) GeeksforGeeks Geeks

Ans. (d)

Explanation: An else block between finally block between try is defined in python. If there is no exception in try block then else is executed and then the finally block. An except block can be defined to catch multiple exception.

[Output of python program | Set 11(Lists)](https://www.geeksforgeeks.org/output-python-program-set-11lists/) =>>

1) What is the output of the following program?

data = [2, 3, 9]

temp = [[x for x in[data]] for x in range(3)]

print (temp)

a) [[[2, 3, 9]], [[2, 3, 9]], [[2, 3, 9]]]

b) [[2, 3, 9], [2, 3, 9], [2, 3, 9]]

c) [[[2, 3, 9]], [[2, 3, 9]]]

d) None of these

Ans. (a)

Explanation: [x for x in[data] returns a new list copying the values in the list data and the outer for statement prints the newly created list 3 times.

2) What is the output of the following program?

data = [x for x in range(5)]

temp = [x for x in range(7) if x in data and x%2==0]

print(temp)

a) [0, 2, 4, 6]

b) [0, 2, 4]

c) [0, 1, 2, 3, 4, 5]

d) Runtime error

Ans. (b)

Explanation: The is statement checks whether the value lies in list data and if it does whether it’s divisible by 2. It does so for x in (0, 7).

3) What is the output of the following program?

temp = ['Geeks', 'for', 'Geeks']

arr = [i[0].upper() for i in temp]

print(arr)

a) [‘G’, ‘F’, ‘G’]

b) [‘GEEKS’]

c) [‘GEEKS’, ‘FOR’, ‘GEEKS’]

d) Compilation error

Ans. (a)

Explanation: The variable i is used to iterate over each element in list temp. i[0] represent the character at 0th index of i and .upper() function is used to capitalize the character present at i[0].

4) Whats is the output of the following program?

temp = 'Geeks 22536 for 445 Geeks'

data = [x for x in (int(x) for x in temp if x.isdigit()) if x%2 == 0]

print(data)

a) [2, 2, 6, 4, 4]

b) Compilation error

c) Runtime error

d) [‘2’, ‘2’, ‘5’, ‘3’, ‘6’, ‘4’, ‘4’, ‘5’]

Ans. (a)

Explanation: This is an example of nested list comprehension. The inner list created contains a list of integer in temp. The outer list only procures those x which are a multiple of 2.

5) What is the output of the following program?

data = [x for x in (x for x in 'Geeks 22966 for Geeks' if x.isdigit()) if

(x in ([x for x in range(20)]))]

print(data)

a) [2, 2, 9, 6, 6]

b) []

c) Compilation error

d) Runtime error

Ans. (b)

Explanation: Since here x have not been converted to int, the condition in the if statement fails and therefore, the list remains empty.

[Output of python program | Set 12(Lists and Tuples)](https://www.geeksforgeeks.org/output-python-program-set-12lists-tuples/) =>>

1) What is the output of the following program?

L1 = list()

L1.append([1, [2, 3], 4])

L1.extend([7, 8, 9])

print(L1[0][1][1] + L1[2])

a) Type Error: can only concatenate list (not “int”) to list

b) 12

c) 11

d) 38

Ans. (c)

Explanation: In the print(), indexing is used. L1[0] denotes [1, [2, 3], 4], L1[0][1] denotes [2, 3],

L1[0][1][1] = 3 and L1[2] = 8. Thus, the two integers are added, 3 + 8 = 11 and output comes as 11.

2) What is the output of the following program?

L1 = [1, 1.33, 'GFG', 0, 'NO', None, 'G', True]

val1, val2 = 0, ''

for x in L1:

if(type(x) == int or type(x) == float):

val1 += x

elif(type(x) == str):

val2 += x

else:

break

print(val1, val2)

a) 2 GFGNO

b) 2.33 GFGNOG

c) 2.33 GFGNONoneGTrue

d) 2.33 GFGNO

Ans. (d)

Explanation: val1 will only have integer and floating values val1 = 1 + 1.33 + 0 = 2.33 and val2 will have string values val2 =’GFG’ + ‘NO’ = ‘GFGNO’. String ‘G’ will not be part of val2 as the for loop will break at None, thus ‘G’ will not be added to val2.

3) What is the output of the following program?

L1 = [1, 2, 3, 4]

L2 = L1

L3 = L1.copy()

L4 = list(L1)

L1[0] = [5]

print(L1, L2, L3, L4)

a) [5, 2, 3, 4] [5, 2, 3, 4] [1, 2, 3, 4] [1, 2, 3, 4]

b) [[5], 2, 3, 4] [[5], 2, 3, 4] [[5], 2, 3, 4] [1, 2, 3, 4]

c) [5, 2, 3, 4] [5, 2, 3, 4] [5, 2, 3, 4] [1, 2, 3, 4]

d) [[5], 2, 3, 4] [[5], 2, 3, 4] [1, 2, 3, 4] [1, 2, 3, 4]

Ans. (d)

Explanation: List L2 is the Shallow copy of L1, while L3 and L4 are Deep Copy(True Copy) of List L1. L1[0] = [5], implies that at index 0, list [5] will be present and not integer value 5.

4) What is the output of the following program?

import sys

L1 = tuple()

print(sys.getsizeof(L1), end = " ")

L1 = (1, 2)

print(sys.getsizeof(L1), end = " ")

L1 = (1, 3, (4, 5))

print(sys.getsizeof(L1), end = " ")

L1 = (1, 2, 3, 4, 5, [3, 4], 'p', '8', 9.777, (1, 3))

print(sys.getsizeof(L1))

a) 0 2 3 10

b) 32 34 35 42

c) 48 64 72 128

d) 48 144 192 480

Ans. (c)

Explanation: An Empty Tuple has 48 Bytes as Overhead size and each additional element requires 8 Bytes.

(1, 2) Size: 48 + 2 \* 8 = 64

(1, 3, (4, 5)) Size: 48 + 3 \* 8 = 72

(1, 2, 3, 4, 5, [3, 4], ‘p’, ‘8’, 9.777, (1, 3)) Size: 48 + 10 \* 8 = 128

2) What is the output of the following program?

T1 = (1)

T2 = (3, 4)

T1 += 5

print(T1)

print(T1 + T2)

a) TypeError

b) (1, 5, 3, 4)

c) 1 TypeError

d) 6 TypeError

Ans. (d)

Explanation: T1 is an integer while T2 is tuple. Thus T1 will become 1 + 5 = 6. But an integer and tuple cannot be added, it will throw TypeError.

[Output of python program | Set 13(Lists and Tuples)](https://www.geeksforgeeks.org/output-python-program-set-13lists-tuples/) =>>

1) What is the output of the following program?

List = [True, 50, 10]

List.insert(2, 5)

print(List, "Sum is: ", sum(List))

a) [True, 50, 10, 5] Sum is: 66

b) [True, 50, 5, 10] Sum is: 65

c) TypeError: unsupported operand type(s) for +: ‘int’ and ‘str’

d) [True, 50, 5, 10] Sum is: 66

Ans. (d)

Explanation: The List is initially has 3 elements. The insert() adds element 5 at index 2, moving element 10 at index 3 and the List becomes [True, 50, 5, 10]. Boolean has an integer value of 1, thus sum becomes 1 + 50 + 5 + 10 = 66.

2) What is the output of the following program?

T = (1, 2, 3, 4, 5, 6, 7, 8)

print(T[T.index(5)], end = " ")

print(T[T[T[6]-3]-6])

a) 4 0

b) 5 8

c) 5 IndexError

d) 4 1

Ans. (b)

Explanation: The inbuilt function index() returns the index of the element. T.index(5) = 4 and T[4] = 5. The other print statement has indexing of tuples, similar to that of Lists. T[6] = 7, T[6]-3 = 4, T[T[6]-3] = 5, T[T[6]-3]-6 = -1 and T[T[T[6]-3]-6], i.e. T[-1] = 8.

3) What is the output of the following program?

L = [1, 3, 5, 7, 9]

print(L.pop(-3), end = ' ')

print(L.remove(L[0]), end = ' ')

print(L)

a) 5 None [3, 7, 9]

b) 5 1 [3, 7, 9]

c) 5 1 [3, 7, 9]

d) 5 None [1, 3, 7, 9]

Ans. (a)

Explanation: pop() will delete and return the element whose index was passed as parameter. L.pop(-3) will delete 5 and return 5, which is printed by print(). remove() does return any value, it’s a void function. L[0] = 1, L.remove(1) will delete 1 from the list and the list remains to be [3, 7, 9].

4) What is the output of the following program?

def REVERSE(L):

L.reverse()

return(L)

def YKNJS(L):

List = list()

List.extend(REVERSE(L))

print(List)

L = [1, 3.1, 5.31, 7.531]

YKNJS(L)

a) [1, 3.1, 5.31, 7.531]

b) [7.531, 5.31, 3.1, 1]

c) IndexError

d) AttributeError: ‘NoneType’ object has no attribute ‘REVERSE’

Ans. (b)

Explanation: REVERSE() reverses the list and returns it. YKNJS() adds reverse of a list L to the empty list List. L = [1, 3.1, 5.31, 7.531], gets reversed and becomes [7.531, 5.31, 3.1, 1].

5) What is the output of the following program?

from math import sqrt

L1 = [x\*\*2 for x in range(10)].pop()

L1 + = 19

print(sqrt(L1), end = " ")

L1 = [x\*\*2 for x in reversed(range(10))].pop()

L1 + = 16

print(int(sqrt(L1)))

a) 10.0 4.0

b) 4.3588 4

c) 10 .0 4

d) 10.0 0

Ans. (c)

Explanation: The first list comphrension will create list as [0, 1, 4, 9, 16, 25, 36, 49, 64, 81] and pop() will return 81. 81 + 19 = 100, whose square root is 10.0 and similarly in 2nd case pop() will return 0 due to the reversed range and integer value of square root of 16 is 4.

[Output of python program | Set 14 (Dictionary)](https://www.geeksforgeeks.org/output-python-program-set-14-dictionary/) =>>

1) What is the output of the following program?

D = dict()

for x in enumerate(range(2)):

D[x[0]] = x[1]

D[x[1]+7] = x[0]

print(D)

a) KeyError

b) {0: 1, 7: 0, 1: 1, 8: 0}

c) {0: 0, 7: 0, 1: 1, 8: 1}

d) {1: 1, 7: 2, 0: 1, 8: 1}

Ans. (c)

Explanation: enumerate() will return a tuple, the loop will have x = (0, 0), (1, 1). Thus D[0] = 0, D[1] = 1, D[0 + 7] = D[7] = 0 and D[1 + 7] = D[8] = 1.

Note: Dictionary is unordered, so the sequence of the key-value pair may differ in each output.

2) What is the output of the following program?

D = {1 : 1, 2 : '2', '1' : 1, '2' : 3}

D['1'] = 2

print(D[D[D[str(D[1])]]])

a) 2

b) 3

c) ‘2’

d) KeyError

Ans. (b)

Explanation: Simple key-value pair is used recursively, D[1] = 1, str(1) = ‘1’. So, D[str(D[1])] = D[‘1’] = 2, D[2] = ‘2’ and D[‘2’] = 3.

3) What is the output of the following program?

D = {1 : {'A' : {1 : "A"}, 2 : "B"}, 3 :"C", 'B' : "D", "D": 'E'}

print(D[D[D[1][2]]], end = " ")

print(D[D[1]["A"][2]])

a) D C

b) E B

c) D B

d) E KeyError

Ans. (d)

Explanation: Key-Value Indexing is used in the example above. D[1] = {‘A’ : {1 : “A”}, 2 : “B”}, D[1][2] = “B”, D[D[1][2]] = D[“B”] = “D” and D[“D”] = “E”. D[1] = {‘A’ : {1 : “A”}, 2 : “B”}, D[1][“A”] = {1 : “A”} and D[1][“A”][2] doesn’t exists, thus KeyError.

4) What is the output of the following program?

D = dict()

for i in range (3):

for j in range(2):

D[i] = j

print(D)

a) {0: 0, 1: 0, 2: 0}

b) {0: 1, 1: 1, 2: 1}

c) {0: 0, 1: 0, 2: 0, 0: 1, 1: 1, 2: 1}

d) TypeError: Immutable object

Ans. (b)

Explanation: 1st loop will give 3 values to i 0, 1 and 2. In the empty dictionary, valued are added and overwrited in j loop, for eg. D[0] = [0] becomes D[0] = 1, due to overwriting.

5) Which of the options below could possibly be the output of the following program?

D = {1 : [1, 2, 3], 2: (4, 6, 8)}

D[1].append(4)

print(D[1], end = " ")

L = list(D[2])

L.append(10)

D[2] = tuple(L)

print(D[2])

a) [1, 2, 3, 4] [4, 6, 8, 10]

b) [1, 2, 3] (4, 6, 8)

c) ‘[1, 2, 3, 4] TypeError: tuples are immutable

d) [1, 2, 3, 4] (4, 6, 8, 10)

Ans. (d)

Explanation: In the first part key-value indexing is used and 4 is appended into the list. As tuples are immutable, in the second part the tuple is converted into a list, valued 10 is added and then converted back to list.

[Output of python program | Set 15 (Modules)](https://www.geeksforgeeks.org/output-python-program-set-15-modules/) =>>

1) Which of the options below could possibly be the output of the following program?

from random import randrange

L = list()

for x in range(5):

L.append(randrange(0, 100, 2)-10)

# Choose which of outputs below are valid for this code.

print(L)

a) [-8, 88, 8, 58, 0]

b) [-8, 81, 18, 46, 0]

c) [-7, 88, 8, 58, 0]

d) [-8, 88, 94, 58, 0]

Ans. (a)

Explanation: The for loop will result in appending 5 elements to list L. Range of the elements lies from [0, 98] – 10 = [-10, 88], which rules out option (d). The upper range is 98 because the step size is 2, thus option (c) and (b) are invalid. Also note that each time you may not get the same output or the one in the options as the function is random.

2) What is the output of the following program?

from math import \*

a = 2.13

b = 3.7777

c = -3.12

print(int(a), floor(b), ceil(c), fabs(c))

a) 2 3 -4 3

b) 2 3 -3 3.12

c) 2 4 -3 3

d) 2 3 -4 3.12

Ans. (b)

Explanation: int() returns the integer value of a number, int(2.13) = 2. floor() returns the largest integer lesser or equal to the number, floor(3.777) = 3. ceil() returns smallest integer greater or equal to the number, ceil(-3.12) = -3. fabs() return the modulus of the number, thus fabs(-3.12) = 3.12.

3) What is the output of the following program?

import re

p = re.compile('\d+')

print(p.findall("I met him once at 11 A.M. on 4th July 1886"), end = " ")

p = re.compile('\d')

print(p.findall("I went to him at 11 A.M."))

a) [’11’, ‘4’, ‘1886’, ’11’]

b) [‘1141886’] [‘1’, ‘1’]

c) [’11’, ‘4’, ‘1886’] [’11’]

d) [’11’, ‘4’, ‘1886’] [‘1’, ‘1’]

Ans. (d)

Explanation: \d is equivalent to [0-9] and \d+ will match a group on [0-9], group of one or greater size. In first statement, group of digits are 11, 4, 1886. In the second statement, \d will treat each each digit as different entity, thus 1, 1.

4) What is the output of the following program?

import re

print(re.sub('ge', '\*\*', 'Geeksforgeeks', flags = re.IGNORECASE), end = " ")

print(re.sub('ge', '\*\*', 'Geeksforgeeks'))

a) \*\*eksfor\*\*eks \*\*eksfor\*\*eks

b) \*\*eksfor\*\*eks Geeksfor\*\*eks

c) \*\*Geeksfor\*\*geeks Geeksfor\*\*geeks

d) TypeError: ‘str’ object does not support item assignment

Ans. (b)

Explanation: In the first print statement, all ‘ge’ will be replaced ‘\*\*’, and case is igonred. Case is not igonred in 2nd statement, thus ‘ge’ will be replaced but not ‘Ge’.

5) Which of the options below could possibly be the output of the following program?

import math

import random

L = [1, 2, 30000000000000]

for x in range(3):

L[x] = math.sqrt(L[x])

# random.choices() is available on Python 3.6.1 only.

string = random.choices(["apple", "carrot", "pineapple"], L, k = 1)

print(string)

a) [‘pineapple’]

b) [‘apple’]

c) ‘pineapple’

d) both a and b

Ans. (d)

Explanation: Two modules math and random are used, L after the for loop will be [1.0, 1.4142135623730951, 5477225.575051662]. choices() has choice as 1st parameter and their weights as second parameter, k is the number valued needed from choice. The answer will come out to ‘pineapple’ almost always due to it’s weight but ‘apple’ and ‘carrot’ may turn out to be the output at times.

[Output of Python program | Set 15 (Loops)](https://www.geeksforgeeks.org/output-python-program-set-15-loops/) =>>

1) What is the output of the following program?

x = ['ab', 'cd']

for i in x:

i.upper()

print(x)

Output:

['ab', 'cd']

Explanation:

The function upper() does not modify a string in place, but it returns a new string which here isn’t being stored anywhere. So we will get our original list as output.

2) What is the output of the following program?

x = ['ab', 'cd']

for i in x:

x.append(i.upper())

print(x)

Output:

No Output

Explanation:

The loop does not terminate as new elements are being added to the list in each iteration. So our program will stuck in infinite loop

3) What is the output of the following program?

i = 1

while True:

if i%3 == 0:

break

print(i)

i + = 1

Output:

No Output

Explanation:

The program will give no output as there is an error in the code. In python while using expression there shouldn’t be a space between + and = in +=.

4) What is the output of the following program?

x = 123

for i in x:

print(i)

Output:

Error!

Explanation:

Objects of type int are not iterable instead a list, dictionary or a tuple should be used.

5) What is the output of the following program?

for i in [1, 2, 3, 4][::-1]:

print (i)

Output:

4

3

2

1

Explanation:

Adding [::-1] beside your list reverses the list. So output will be the elements of original list but in reverse order.

[Output of Python program | Set 16 (Threads)](https://www.geeksforgeeks.org/output-python-program-set-16-threads/) =>>

1) What is the output of the following program?

import threading

barrier = threading.Barrier(4)

class thread(threading.Thread):

def \_\_init\_\_(self, thread\_ID, thread\_name):

threading.Thread.\_\_init\_\_(self)

self.thread\_ID = thread\_ID

self.thread\_name = thread\_name

def run(self):

print("ThreadID = " + str(self.thread\_ID) + ", ThreadName = " +

self.thread\_name + "\n")

try:

barrier = threading.Barrier(4)

barrier.wait()

except:

print("barrier broken")

thread1 = thread(100, "GFG")

thread2 = thread(101, "Geeks")

thread3 = thread(102, "GeeksforGeeks")

thread1.start()

thread2.start()

thread3.start()

barrier.wait()

print("Exit")

a) ThreadID = 100, ThreadName = GFG

ThreadID = 101, ThreadName = Geeks

ThreadID = 102, ThreadName = GeeksforGeeks

b) ThreadID = 100, ThreadName = GFG

ThreadID = 101, ThreadName = Geeks

ThreadID = 102, ThreadName = GeeksforGeeks

Exit

c) Compilation error

d) Runtime error

Ans. (a)

Explanation: This is an example of deadlock. Each thread creates it’s own barrier and calls .wait() function on that barrier.

2) Which among the following is NOT the output of the following program?

import threading

class thread(threading.Thread):

def \_\_init\_\_(self, thread\_ID, thread\_name):

threading.Thread.\_\_init\_\_(self)

self.thread\_ID = thread\_ID

self.thread\_name = thread\_name

def run(self):

print(self.thread\_name)

thread1 = thread(100, "GFG ")

thread2 = thread(101, "Geeks ")

thread3 = thread(102, "GeeksforGeeks ")

thread1.start()

thread2.start()

thread3.start()

print("Exit")

a) GFG Geeks GeeksforGeeks Exit

b) Exit Geeks GeeksforGeeks GFG

c) GFG Exit GeeksforGeeks Geeks

d) None of the above

Ans. (d)

Explanation: Calling start() method on a thread moves the thread to ready state. It’s the responsibility of the thread scheduler to schedule the thread. So, a particular thread can be scheduled at any instant.

3) What is the output of the following program?

import threading

class thread(threading.Thread):

def \_\_init\_\_(self, thread\_ID, thread\_name):

threading.Thread.\_\_init\_\_(self)

self.thread\_ID = thread\_ID

self.thread\_name = thread\_name

def run(self):

print(self.thread\_name)

thread1 = thread(100, "GFG ")

thread2 = thread(101, "Geeks ")

thread3 = thread(102, "GeeksforGeeks ")

thread = []

thread.append(thread1)

thread.append(thread2)

thread.append(thread3)

thread1.start()

thread2.start()

for thread in thread:

thread.join()

thread3.start()

print("Exit")

a) GFG Geeks GeeksforGeeks Exit

b) Compilation error

c) Program will halt in between due to Runtime error

d) None of these

Ans. (c)

Explanation: join() method cannot be called on a thread that hasn’t yet started it’s execution.

4) What is the output of the following program?

import threading

i = 5

class thread(threading.Thread):

def \_\_init\_\_(self, thread\_ID, thread\_name):

threading.Thread.\_\_init\_\_(self)

self.thread\_ID = thread\_ID

self.thread\_name = thread\_name

def run(self):

i = i + 1

print(i)

thread1 = thread(100, "GFG ")

thread2 = thread(101, "Geeks")

thread1.start()

thread2.start()

a) 66

b) 67

c) Compilation error

d) Runtime error

Ans. (d)

Explanation: Each thread has it’s own space reserved in memory. So, for each threads, thread1 and thread2, the variable temp is not declared as temp is not defined within the thread’s run method.

5) What is the output of the following program?

import threading

class thread(threading.Thread):

def \_\_init\_\_(self, thread\_ID):

self.thread\_ID = thread\_ID

def run(self):

print(self.thread\_ID)

thread1 = thread(100)

thread1.start()

a) 100

b) Compilation error

c) Runtime error

d) None of these

Ans. (c)

Explanation: thread.\_\_init\_\_() has to be called explicitly by each of the threads being created within the \_\_init\_\_ function.

[Output of Python program | Set 17](https://www.geeksforgeeks.org/output-python-program-set-17/) =>>

1. What is the output of the following program?

numberGames = {}

numberGames[(1,2,4)] = 8

numberGames[(4,2,1)] = 10

numberGames[(1,2)] = 12

sum = 0

for k in numberGames:

sum += numberGames[k]

print len(numberGames) + sum

Output:

33

Explanation:

Tuples can be used for keys into dictionary. The tuples can have mixed length and the order of the items in the tuple is considered when comparing the equality of the keys.

2. What is the output of the following program?

my\_tuple = (1, 2, 3, 4)

my\_tuple.append( (5, 6, 7) )

print len(my\_tuple)

Output:

Error !

Explanation:

Tuples are immutable and don’t have an append method as in case of Lists.Hence an error is thrown in this case.

3. What is the output of the following program?

t = (1, 2)

print 2 \* t

Output:

(1, 2, 1, 2)

Explanation:

Asterick Operatr (\*) operator concatenates tuple.

4. What is the output of the following program?

d1 = {"john":40, "peter":45}

d2 = {"john":466, "peter":45}

print d1 > d2

Output:

False

Explanation:

Arithmetic operator less than ( < ) or greater than ( > ) can be used with dictionaries and each corresponding key with its values are compared

5. What is the output of the following program?

my\_tuple = (6, 9, 0, 0)

my\_tuple1 = (5, 2, 3, 4)

print my\_tuple > my\_tuple1

Output:

True

Explanation:

Each elements of the tuples are compared one by one and if maximum number of elements are there in tuple1 which are greater of equal to corresponding element of tuple2 then tuple1 is said to be grater than tuple2.

[Output of Python Programs | Set 18 (List and Tuples)](https://www.geeksforgeeks.org/output-python-programs-set-18-list-tuples/) =>>

1) What is the output of the following program?

L = list('123456')

L[0] = L[5] = 0

L[3] = L[-2]

print(L)

a) [0, ‘2’, ‘3’, ‘4’, ‘5’, 0]

b) [‘6’, ‘2’, ‘3’, ‘5’, ‘5’, ‘6’]

c) [‘0’, ‘2’, ‘3’, ‘5’, ‘5’, ‘0’]

d) [0, ‘2’, ‘3’, ‘5’, ‘5’, 0]

Ans. (d)

Explanation: L[0] is ‘1’ and L[5] is ‘6’, both of these elements will be replaced by 0 in the List. L[3], which is 4 will be replaced L[-2] i.e. 5.

2) What is the output of the following program?

T = tuple('geeks')

a, b, c, d, e = T

b = c = '\*'

T = (a, b, c, d, e)

print(T)

a) (‘g’, ‘\*’, ‘\*’, ‘k’, ‘s’)

b) (‘g’, ‘e’, ‘e’, ‘k’, ‘s’)

c) (‘geeks’, ‘\*’, ‘\*’)

d) KeyError

Ans. (a)

Explanation: A tuple is created as T = (‘g’, ‘e’, ‘e’, ‘k’, ‘s’), then it is unpacked into a, b, c, d and e, mapping from ‘g’ to a and ‘s’ to e. b and c which are both ‘e’ are equal to ‘\*’ and then the existing tuple is replaced by packing a, b, c, d and e into a tupple T.

3) What is the value of L at the end of execution of the following program?

L = [2e-04, 'a', False, 87]

T = (6.22, 'boy', True, 554)

for i in range(len(L)):

if L[i]:

L[i] = L[i] + T[i]

else:

T[i] = L[i] + T[i]

break

a) [6.222e-04, ‘aboy’, True, 641]

b) [6.2202, ‘aboy’, 1, 641]

c) [6.2202, ‘aboy’, True, 87]

d) [6.2202, ‘aboy’, False, 87]

Ans. (d)

Explanation: The for loop will run for i = 0 to i = 3, i.e. 4 times(len(L) = 4). 2e-04 is same as 0.0002, thus L[i] = 6.22 + 0.0002 = 6.2202. String addition will result in concatenation, ‘a’ + ‘boy’ = ‘aboy’. False + True is True, it’ll return the integer value of 1. As tuples are immutable, the code will end with TypeError, but elements of L will be updated.

4) What is the output of the following program?

T = (2e-04, True, False, 8, 1.001, True)

val = 0

for x in T:

val += int(x)

print(val)

a) 12

b) 11

c) 11.001199999999999

d) TypeError

Ans. (b)

Explanation: Integer value of 2e-04(0.0002) is 0, True holds a value 1 and False a 0, integer value of 1.001 is 1. Thus total 0 + 1 + 0 + 8 + 1 + 1 = 11.

5) Which of the options below could possibly be the output of the following program?

L = [3, 1, 2, 4]

T = ('A', 'b', 'c', 'd')

L.sort()

counter = 0

for x in T:

L[counter] += int(x)

counter += 1

break

print(L)

a) [66, 97, 99, 101]

b) [66, 68, 70, 72]

c) [66, 67, 68, 69]

d) ValueError

Ans. (d)

Explanation: After sort(L), L will be = [1, 2, 3, 4]. Counter = 0, L[0] i.e. 1, x = ‘A’, but Type Conversion of char ‘A’ to integer will throw error and the value cannot be stored in L[0], thus a ValueError.

[Output of Python Programs | Set 19 (Strings)](https://www.geeksforgeeks.org/output-python-programs-set-19-strings/) =>>

1) What is the output of the following program?

str1 = '{2}, {1} and {0}'.format('a', 'b', 'c')

str2 = '{0}{1}{0}'.format('abra', 'cad')

print(str1, str2)

a) c, b and a abracad0

b) a, b and c abracadabra

c) a, b and c abracadcad

d) c, b and a abracadabra

Ans. (d)

Explanation: String function format takes a format string and an arbitrary set of positional and keyword arguments. For str1 ‘a’ has index 2, ‘b’ index 1 and ‘c’ index 0. str2 has only two indices 0 and 1. Index 0 is used twice at 1st and 3rd time.

2) What is the output of the following program?

a = 2

b = '3.77'

c = -8

str1 = '{0:.4f} {0:3d} {2} {1}'.format(a, b, c)

print(str1)

a) 2.0000 2 -8 3.77

b) 2 3.77 -8 3.77

c) 2.000 3 -8 3.77

d) 2.000 2 8 3.77

Ans. (a)

Explanation: At Index 0, integer a is formatted into a float with 4 decimal points, thus 2.0000. At Index 0, a = 2 is formatted into a integer, thus it remains to 2. Index 2 and 1 values are picked next, which are -8 and 3.77 respectively.

3) What is the output of the following program?

import string

import string

Line1 = "And Then There Were None"

Line2 = "Famous In Love"

Line3 = "Famous Were The Kol And Klaus"

Line4 = Line1 + Line2 + Line3

print(string.find(Line1, 'Were'), string.count((Line4), 'And'))

a) True 1

b) 15 2

c) (15, 2)

d) True 2

Ans. (c)

Explanation: ‘Were’ is at Index 15 in Line1, find() returns the index of substring if found in the string Line1. count() returns the total number of occurrences of the substring. Line4 is concatenated string from Line1, Line2 and Line3. This code works well with Python v2.x, as some string functions are deprecated in Python v3.x.

4) What is the output of the following program?

line = "I'll come by then."

eline = ""

for i in line:

eline += chr(ord(i)+3)

print(eline)

a) L\*oo frph e| wkhq1

b) L\*oo#frph#e|#wkhq1

c) l\*oo@frph@e|$wkhq1

d) O\*oo#Frph#E|#wKhq1

Ans. (b)

Explanation: This piece of code ciphers the plain text. Each character is moved to its 3rd next character by increasing the ascii value. ‘I’ becomes ‘L’, thus option (c) and (d) are ruled out. ‘ ‘ has ascii value of 32, thus it’ll become 35(‘#’), thus option (a) is ruled out as, ‘ ‘ can not remain to be ‘ ‘ in the ciphered text.

5) What is the output of the following program?

line = "What will have so will"

L = line.split('a')

for i in L:

print(i, end=' ')

a) [‘What’, ‘will’, ‘have’, ‘so’, ‘will’]

b) Wh t will h ve so will

c) What will have so will

d) [‘Wh’, ‘t will h’, ‘ve so will’]

Ans. (b)

Explanation: split() will use ‘a’ as the delimiter. It’ll create parition at ‘a’, thus split() return an array L, which is in [‘Wh’, ‘t will h’, ‘ve so will’]. For loop will print the elements of the list.

[Output of Python Programs | Set 20 (Tuples)](https://www.geeksforgeeks.org/output-python-programs-set-20-tuples/) =>>

1. What will be the output of the following program ?

tuple = (1, 2, 3, 4)

tuple.append( (5, 6, 7) )

print(len(my\_tuple))

Options:

1. 1

2. 2

3. 5

4. Error

Output:

4. Error

Explanation: In this case an exception will be thrown as tuples are immutable and don’t have an append method.

2. What will be the output of the following program ?

tuple = {}

tuple[(1,2,4)] = 8

tuple[(4,2,1)] = 10

tuple[(1,2)] = 12

\_sum = 0

for k in tuple:

\_sum += tuple[k]

print(len(tuple) + \_sum)

Options:

1. 34

2. 12

3. 31

4. 33

Output:

4. 33

Explanation: Tuples can be used for keys into dictionary. The tuples can have mixed length and the order of the items in the tuple is considered when comparing the equality of the keys.

3. What will be the output of the following program ?

tuple1 = (1, 2, 4, 3)

tuple2 = (1, 2, 3, 4)

print(tuple1 < tuple2)

Options:

1.Error

2.True

3.False

4.Unexpected

Output:

3. False

Explanation: In this case elements will be compared one by one. So, when it compare 4 with 3 it will return False.

4. What will be the output of the following program ?

tuple = (1, 2, 3)

print(2 \* tuple)

Options:

1.(1, 2, 3, 1, 2, 3)

2.(1, 2, 3, 4, 5, 6)

3.(3, 6, 9)

4.Error

Output:

1. (1, 2, 3, 1, 2, 3)

Explanation: ‘\*’ operator is used to concatenate tuples.

5. What will be the output of the following program ?

tuple=("Check")\*3

print(tuple)

Options:

1.Unexpected

2.(3Check)

3.(CheckCheckCheck)

4.Syntax Error

Output:

3. (CheckCheckCheck)

Explanation: Here (“Check”) will be treated as is a string not a tuple as there is no comma after the element.

[Output of Python Programs | Set 21 (Bool)](https://www.geeksforgeeks.org/output-python-programs-set-21-bool/) =>>

1. What is the output of the code:

print(bool('False'))

print(bool())

1. False, True

2. None, None

3. True, True

4. True, False

Output:

4. True, False

Explanation: If the argument passed to the bool function does not amount to zero then the Boolean function returns true else it always returns false. In the above code, in first line ‘False’ is passed to the function which is not amount to 0. Therefore output is true. In the second line, an empty list is passed to the function bool. Hence the output is false.

2. What is the output of the code:

print(not(4>3))

print(not(5&5))

1. False, False

2. None, None

3. True, True

4. Ture, False

Output:

1. False, False

Explanation: The not function returns true if the argument is false, and false if the argument is true. Hence the first line of above code returns false, and the second line will also returns false.

3. What is the output of the code:

print(['love', 'python'][bool('gfg')])

1. love

2. python

3. gfg

4. None

Output:

2. python

Explanation: We can read the above code as print ‘love’ if the argument passed to the Boolean function is zero else print ‘python’. The argument passed to the Boolean function in the above code is ‘gfg’, which does not amount to zero and hence the output is: “python”.

4. What is the output of the code:

mylist =[0, 5, 2, 0, 'gfg', '', []]

print(list(filter(bool, mylist)))

1. [0, 0, ]

2. [0, 5, 2, 0, ‘gfg’, ”, []]

3. Error

4. [5, 2, ‘gfg’]

Output:

4. [5, 2, 'gfg']

Explanation: The code above returns a new list containing only those elements of the list mylist which are not equal to zero. Hence the output is: [5, 2, ‘gfg’].

5. What is the output of the code:

if (7 < 0) and (0 < -7):

print("abhi")

elif (7 > 0) or False:

print("love")

else:

print("geeksforgeeks")

1. geeksforgeeks

2. love

3. abhi

4. Error

Output:

2. love

Explanation: The code shown above prints the appropriate option depending on the conditions given. The condition which matches is (7>0), and hence the output is: “love”.

[Output of Python Programs | Set 22 (Loops)](https://www.geeksforgeeks.org/output-python-programs-set-22-loops/) =>>

1. What is the output of the following?

mylist = ['geeks', 'forgeeks']

for i in mylist:

i.upper()

print(mylist)

1. [‘GEEKS’, ‘FORGEEKS’].

2. [‘geeks’, ‘forgeeks’].

3. [None, None].

4. Unexpected

Output:

2. [‘geeks’, ‘forgeeks’]

Explanation: The function upper() does not modify a string in place, it returns a new string which isn’t being stored anywhere.

2. What is the output of the following?

mylist = ['geeks', 'forgeeks']

for i in mylist:

mylist.append(i.upper())

print(mylist)

1. [‘GEEKS’, ‘FORGEEKS’].

2. [‘geeks’, ‘forgeeks’, ‘GEEKS’, ‘FORGEEKS’].

3. [None, None].

4. None of these

Output:

4. None of these

Explanation:The loop does not terminate as new elements are being added to the list in each iteration.

3. What is the output of the following?

i = 1

while True:

if i % 0O7 == 0:

break

print(i)

i += 1

1. 1 2 3 4 5 6.

2. 1 2 3 4 5 6 7.

3. error.

4. None of these

Output:

1. 1 2 3 4 5 6

Explanation: The loop will terminate when i will be equal to 7.

4. What is the output of the following?

True = False

while True:

print(True)

break

1. False.

2. True.

3. Error.

4. None of these

Output:

3. Error

Explanation: SyntaxError, True is a keyword and it’s value cannot be changed.

5. What is the output of the following?

i = 1

while True:

if i % 3 == 0:

break

print(i)

i + = 1

1. 1 2 3.

2. 1 2.

3. Syntax Error.

4. None of these

Output:

3. Syntax Error

Explanation: SyntaxError, there shouldn’t be a space between + and = in +=.

[Output of Python Programs | Set 23 (String in loops)](https://www.geeksforgeeks.org/output-python-programs-set-23-string-loops/) =>>

1. What is the output of the following?

my\_string = "geeksforgeeks"

i = "i"

while i in my\_string:

print(i, end =" ")

1. None

2. geeksforgeeks

3. i i i i i i …

4. g e e k s f o r g e e k s

Output:

1. None

Explanation: ‘i’ is not present in string ‘geeksforgeeks’

2. What is the output of the following?

i = 0

while i < 3:

print(i)

i += 1

else:

print(0)

1. 0 1 2 3 0

2. 0 1 2 0

3. 0 1 2

4. Error

Output:

2. 0 1 2 0

Explanation: The else part is executed when the condition in the while statement is false.

3. What is the output of the following?

my\_string = 'geeksforgeeks'

for i in range(my\_string):

print(i)

1. 0 1 2 3 … 12

2. geeksforgeeks

3. None

4. Error

Output:

4. Error

Explanation: range(str) is not allowed.

4. What is the output of the following?

my\_string = 'geeksforgeeks'

for i in range(len(my\_string)):

my\_string[i].upper()

print (my\_string)

1. GEEKSFORGEEKS

2. geeksforgeeks

3. Error

4. None

Output:

2. geeksforgeeks

Explanation: Changes do not happen in-place, rather it will return a new instance of the string.

5. What is the output of the following?

my\_string = 'geeksforgeeks'

for i in range(len(my\_string)):

print (my\_string)

my\_string = 'a'

1. gaaaaaaaaaaaa

2. geeksforgeeks a a a a a a a a a a a a

3. Error

4. None

Output:

2. geeksforgeeks a a a a a a a a a a a a

Explanation: String is modified only after ‘geeksforgeeks’ has been printed once.

[Output of Python Programs | Set 24 (Sets)](https://www.geeksforgeeks.org/output-python-programs-set-24-sets/) =>>

1. What is the output of the code shown below?

sets = {1, 2, 3, 4, 4}

print(sets)

Options:

1. {1, 2, 3}

2. {1, 2, 3, 4}

3. {1, 2, 3, 4, 4}

4. Error

Output:

2. {1, 2, 3, 4}

Explanation : Duplicate values are not allowed in sets. Hence, the output of the code shown above will be a set containing the duplicate value only once. Hence output will be {1, 2, 3, 4}.

2. What is the output of the code shown below?

sets = {3, 4, 5}

sets.update([1, 2, 3])

print(sets)

Options:

1. {1, 2, 3, 4, 5}

2. {3, 4, 5, 1, 2, 3}

3. {1, 2, 3, 3, 4, 5}

4. Error

Output:

1. {1, 2, 3, 4, 5}

Explanation: The method update adds elements to a set.

3. What is the output of the code shown below?

set1 = {1, 2, 3}

set2 = set1.copy()

set2.add(4)

print(set1)

Options:

1. {1, 2, 3, 4}

2. {1, 2, 3}

3. Invalid Syntax

4. Error

Output:

2. {1, 2, 3}

Explanation: In the above piece of code, set2 is barely a copy and not an alias of set1. Hence any change made in set2 isn’t reflected in set1.

4. What is the output of the code shown below?

set1 = {1, 2, 3}

set2 = set1.add(4)

print(set2)

Options:

1. {1, 2, 3, 4}

2. {1, 2, 3}

3. Invalid Syntax

4. None

Output:

4. None

Explanation: add method doesn’t return anything. Hence there will be no output.

5. What is the output of the code shown below?

set1 = {1, 2, 3}

set2 = {4, 5, 6}

print(len(set1 + set2))

Options:

1. 3

2. 6

3. Unexpected

4. Error

Output:

4. Error

Explanation: unsupported operand type(s) for +: ‘set’ and ‘set’.