Week-4, Practice, Theory

Problem 1

Question

Match the statements with respective output in the below table.

Statement	OUTPUT
(a) `type(float("10.2"))`	(1) Returns `ValueError`
(b) `int("2.3")`	(2) Returns `str`
(c) `del x`	(3) Returns `Float`
(d) `type(str(100))`	(4) `SyntaxError`
(e) `bool('False')`	(5) Deletes variable `x` from memory if exists, otherwise throws `NameError`
(f) `"12345".index(1)`	(6) Returns `True`
(g) `from random import randint()`	(7) Returns `TypeError`

Select the correct option. This is a MCQ type question.

(A) (a) - (1), (b) - (3), (c) - (5), (d) - (2), (e) - (6), (f) - (7), (g) - (4)

(B) (a) - (3), (b) - (1), (c) - (5), (d) - (2), (e) - (6), (f) - (7), (g) - (4)

(C) (a) - (3), (b) - (5), (c) - (1), (d) - (6), (e) - (2), (f) - (7), (g) - (4)

(D) (a) - (3), (b) - (5), (c) - (1), (d) - (2), (e) - (6), (f) - (4), (g) - (7)

Answer

(B)

Solution

(a) , (d) `type()` function gives the data type of the value passed to this function.

(b) `int()` function converts an integer in `str` form to `int` data type. To change a string of `float` value to an `int`, it should be first converted a `float` value using `float()` first and then `int()` should be applied. Hence, `ValueError` is shown to the user.

(c) `del` is used to delete a variable in python.

(e) `bool` function converts the value passed to it to a `bool` type. Since the value inside is a non empty `string`, it is converted to `True`.

(f) `index()` function returns the index of first occurrence of the character in the given string. Since, the string `'12345'` contains string `'1'` but does not contain the integer `1` , it throws `TypeError`.

(g) Correct syntax is: `from random import randint`, hence it shows `SyntaxError`.

Problem 2

Question

What is the output of the following code? The input is a string of odd length. This is a MSQ type question.

```
input_string = input()
x = input_string[len(input_string)//2]
while x in input_string:
    input_string = input_string[:-1]
    print(x, end = "")
```

(a) It prints a string where each character is the first character of `input_string`

(b) It prints a string of the length `len(input_string) - input_string.index(x)`

(c) It prints `input_string` in the reverse order

(d) It prints a string where each character is the middle character of the `input_string`

(e) It prints a string of almost same length as the `input_string`

Answer

(b), (d)

Solution

Option (b) and (d) are correct. Since the `input_string` is of odd length, the position `len(input_string)//2` indicates the middle index and `x` represents the character at this index. The `while` loop checks if `x` exists in the `input_string`. Inside the body of the loop, in each iteration `input_string` is trimmed from right by one character till the character stored in `x` is found in this trimmed `input_string`.

Hence, it prints a string where each character is the middle character of the `input_string`. The length of this string is `len(input_string) - input_string.index(x)`.

Problem 3

Question

Python has __ statement as a null statement. This is a "Fill in the blank" type question.

Answer

pass

Solution

The `pass` keyword is used to represent an empty / blank statement. It does not perform anything in the program. It is sometimes used as a placeholder statement so that real real code can be filled later.

Problem 4

Question

`L` is a non-empty list of integers and `x` is an integer. Assume that both `L` and `x` have already been defined. The following code does not throw any error when executed. Lines 1 and 6 will be used to refer to the state of the variables before and after key sections of the code are executed.

```
1  pass                # before
2  count = 0
3  while len(L) > 0:
4      L.remove(x)
5      count += 1
6  pass                # after
```

If the value of `count` is 10 at the end of execution of the code (line-6), which of the following statements are true? It is a Multiple Select Question.

(a) `x` is an element of `L` at line-1.

(b) `L` has at least two different (unequal) elements in it at line-1

(c) Length of `L` is 10 at line-1

(d) Length of `L` is 0 at line-6

Answers

(a), (c), (d)

Solution

The first observation is that the loop goes through 10 iterations. This can be inferred from the value of `count`. Next, we know that the code doesn't throw an error. The only line which could introduce a bug in the code is line-4. Combining this with the first observation we see that `x` is certainly present in `L` at line-1. In each iteration, the element `x` is removed from `L`. This means that the element `x` occurs at least 10 times. Can it occur more number of times? No, then the value of `count` would have been more than `10`. Can `L` have any other element in it other than `x` at line-1? This is also not a possibility as the list becomes empty after 10 elements are removed from it.

Problem 5

Question

`P` is a non-empty list of distinct integers that has already been defined. Which of the following statements are true at the end of execution of the code given below? It is a Multiple Select Question (MSQ).

```
1  Q, R = [ ], [ ]
2  for x in P:
3      Q.append(-x)
4  Q.sort()
5  for x in Q:
6      R.append(-x)
```

(a) `P` is always equal to `R`, i.e., `P == R` returns the value `True`

(b) Every element in `P` is present in `R`, but not necessarily in the same sequence.

(c) `R` is a list of integers sorted in descending order.

(d) `R` is a list of integers sorted in ascending order.

Answer

(b), (c)

Solution

The negative of each element in `P` is added to the list `Q` . Then `Q` is sorted in ascending order. Finally, the negative of each element in `Q` is added to `R`. This means that `R` will be a list sorted in descending order. Every element in `P` would be present in `R`. Let us take the simple example of `P = [5, 2, 4, 1, 3]`. Just before line-4 is executed, `Q` will be `[-5, -2, -4, -1, -3]`. After line-4 is executed, `Q` will be `[-5, -4, -3, -2, -1]`. After the entire code is executed, `R` will be `[5, 4, 3, 2, 1]`.

Problem 6

Question

Let `M` be a matrix of numbers that has already been defined and populated. We wish to find the sum of the elements in each row and store all such row-sums in a list called `rsum`. For the i^{th} row, `rsum[i]` should be the sum of all elements in that row. Select the correct code snippets that achieve this. It is a Multiple Select Question (MSQ).

(a)

```
dim = len(M)
rsum = [ ]
for i in range(dim):
    # one element for each row
    rsum.append(0)
    # we will now go through all cells in the ith row
    for j in range(dim):
        # rsum[-1] is the same as rsum[i] in this case
        # add the jth element in the ith row to this sum
        rsum[-1] += M[i][j]
```

(b)

```
dim = len(M)
for i in range(dim):
    # this is WRONG; rsum should be initialized outside the nested loop
    rsum = [ ]
    rsum.append(0)
    for j in range(dim):
        rsum[-1] += M[i][j]
```

(c)

```
dim = len(M)
rsum = [ ]
for i in range(dim):
    rsum.append(0)
for i in range(dim):
    for j in range(dim):
        # this is WRONG; rsum[j] should be rsum[i] or rsum[-1]
        rsum[j] += M[i][j]
```

(d)

```
dim = len(M)
rsum = [ ]
# different from option (a)
# rsum is first initialized with zeros
# one zero for every row
for i in range(dim):
    rsum.append(0)
for i in range(dim):
    for j in range(dim):
        rsum[i] += M[i][j]
```

Answers

(a), (d)

Solution

Check comments in the code.