Week-6, Graded, Theory

Problem-1

Random Number Wheel

Questions 1 to 6 are based on a common theme.

Question-1 [2 marks]

We wish to populate a list of 10000 integers, where each integer is drawn at random from the range 1 to 10, both endpoints included. Which of the following code snippets can be used to achieve this?

Note

- As an example, a list of 20 integers drawn at random from this range would look like this:

```
[2, 1, 3, 5, 1, 4, 9, 5, 9, 4, 1, 3, 8, 6, 4, 7, 10, 1, 7, 5]
```

- The list shall be called `nums`. This list will be used in questions 1 to 6

(a)

```python
import random

nums = [ ]
for i in range(10000):
    nums.append(random.randint(1, 10))
```

(b)

```python
import random

nums = [ ]
for i in range(10):
    nums.append(random.randint(1, 10000))
```

(c)

```python
import random

nums = [ ]
for i in range(10000):
    nums.append(random.randint(0, 11))
```

(d)

```python
import random

nums = [ ]
for i in range(10000):
    nums.append(random.randint(1, 11))
```

Answer

(a)

Solution

```python
import random

nums = [ ]
for i in range(10000):
    nums.append(random.randint(1, 10))
```

`random.randint(a, b)` returns a random number between `a` and `b`, both endpoints inclusive. The loop runs 10,000 times. Each time, a number in this range is added to `nums`.

Question-2 [4 marks]

Using the list `nums` obtained in the previous question, we wish to find the frequency of occurrence of each of the numbers in the range 1 to 10, endpoints included, in the list `nums`. If `P` is a dictionary that stores this information, which of the following snippets of code is/are an appropriate choice? It is a Multiple Select Question (MSQ).

Note

- As an example, consider a list of size 20.

```
[2, 1, 3, 5, 1, 4, 9, 5, 9, 4, 1, 3, 8, 6, 4, 7, 10, 1, 7, 5]
```

Number	Frequency
1	4
2	1
3	2
...	...

- It is up to you to figure out what the keys and values represent in the dictionary `P` by scanning the options given below.

(a)

```python
P = { }
for num in range(1, 10):
    P[num] = 0

for num in nums:
    P[num] += 1
```

(b)

```python
P = { }
for num in range(1, 11):
    P[num] = 0

for num in nums:
    P[num] += 1
```

(c)

```python
P = { }

for num in nums:
    P[num] += 1
```

(d)

```python
P = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 0, 10: 0}

for num in nums:
    P[num] += 1
```

Answer

(b), (d)

Solution

Let us take one of the two solutions:

```python
1  P = { }
2  for num in range(1, 11):
3      P[num] = 0
4
5  for num in nums:
6      P[num] += 1
```

Here, an empty dict `P` is initialized in line-1. Then, lines 2 and 3 add the keys from 1 to 10 to `P`. The value corresponding to each of these keys is `0` to begin with. Lines 5-6 update the values corresponding to these keys. Every time the number `num` is encountered in the list `nums`, its count is incremented by 1.

Question-3 [2 marks]

Using the dictionary `P` obtained in the previous question and the list `nums` that was generated in question-1, consider the following code-block.

```
1 def check(P, N):
2     S = 0
3     for num in P:
4         S += P[num]
5     return S == N
6 
7 print(check(P, len(nums)))
```

What is the output of this code block?

(a) It is `True` if and only if `nums` is sorted in ascending order.

(b) It is `True` if and only if `nums` is sorted in descending order.

(c) It is always `True` and doesn't depend on the order in which elements appear in `nums`.

(d) It is always `False` and doesn't depend on the order in which elements appear in `nums`.

Answer

(c)

Solution

`S` holds the sum of the the frequency of occurrence of each of the numbers from 1 to 10 in `nums`. This sum is nothing but the total number of elements in `nums`. The function `check` is checking if this is indeed the case at line-5. So, it will always return `True`, irrespective of the order in which elements appear in `nums`.

Question-4 [2 marks]

A number is picked at random from the list `nums`. Which of the following expressions gives the probability of obtaining a `5`? Assume that any number in the list is equally likely to be chosen. Use the dictionary `P` obtained from question-2 if needed.

(a)

```
nums[5] / 10000
```

(b)

```
P[5] / 10000
```

(c)

```
5 / 10000
```

(d)

```
P[5] / nums[5]
```

Answer

(b)

Solution

`P[5]` is the number of times the number 5 occurs in `nums`. There are 10,000 numbers in total. Therefore, `P[5] / 10000` is the required probability.

Question-5 [4 marks]

Let us continue with the list `nums` and dictionary `P`. We wish to find the number which occurs the most number of times in `nums`. `most_freq` is a function which accepts the dictionary `P` as input and returns the number in the list `nums` that has the greatest frequency. Select the correct code fragment to achieve this.

Note

- `most_freq` should return the number which has the greatest frequency and not the frequency itself.
- If multiple numbers have the same maximum frequency, then return the largest such number.

(a)

```python
def most_freq(P):
    freq_num, freq = 1, P[1]
    for num in range(1, 11):
        if P[num] >= freq:
            freq_num, freq = num, P[num]
    return freq
```

(b)

```python
def most_freq(P):
    freq_num, freq = 1, P[1]
    for num in range(1, 11):
        if P[num] >= freq_num:
            freq_num, freq = num, P[num]
    return freq_num
```

(c)

```python
def most_freq(P):
    freq_num, freq = 1, P[1]
    for num in range(1, 11):
        if P[num] > freq:
            freq_num, freq = num, P[num]
    return freq_num
```

(d)

```python
def most_freq(P):
    freq_num, freq = 1, P[1]
    for num in range(1, 11):
        if P[num] >= freq:
            freq_num, freq = num, P[num]
    return freq_num
```

Answer

(d)

Solution

Let us look at the correct answer:

```
1  def most_freq(P):
2      freq_num, freq = 1, P[1]
3      for num in range(1, 11):
4          if P[num] >= freq:
5              freq_num, freq = num, P[num]
6      return freq_num
```

We see that `freq_num` holds the number that has the maximum frequency; `freq` holds the frequency of this particular number. We begin with `freq_num = 1` and `freq = P[1]`. In the loop, we go over each of the numbers from 1 to 10. If we see that for some `num`, `P[num]` exceeds the current maximum, then we update `freq_num` and `freq`. Notice that we use `>=` in the if-condition in line-4. This is because of the second point in the note. If there are two numbers which share the maximum frequency, then we need to return the greatest among them. Had we used `>` in line-4, we would have ended up returning the smallest number which has the maximum frequency.

Some points to note regarding the wrong answers:

- (a) is very similar to (d). (a) is wrong because it returns the frequency and not the number having the maximum frequency. This is as per the condition in note-1.
- (b) is wrong because, at line-4, it is checking if `P[num] >= freq_num`. It should be compared against `freq`.
- (c) is wrong because it uses `>` instead of `>=` at line-4.

Question-6 [4 marks]

Let us continue with the list `nums`. The list is said to feature a `streak` if the number `5` occurs at least five times in a row. `streak` is a function which accepts the list `nums` as input and returns `True` if it has a streak, and `False` otherwise. Select the most appropriate option. Some sample test cases for smaller lists are given below:

`nums`	`streak(nums)`
[1, 2, 5, 5, 5, 5, 5, 1, 2, 10]	True
[4, 5, 5, 2, 5, 5, 5, 2, 5]	False
[10, 5, 5, 5, 5, 5, 5, 5, 1, 6, 7, 8, 9]	True

Note

- Remember that your function should work for the list `nums` that we generated in question-1.

(a)

```python
def streak(nums):
    if nums == [1, 2, 5, 5, 5, 5, 5, 1, 2, 10]:
        return True
    if nums == [4, 5, 5, 2, 5, 5, 5, 2, 5]:
        return True
    if nums == [10, 5, 5, 5, 5, 5, 5, 5, 1, 6, 7, 8, 9]:
        return True
    return False
```

(b)

```python
def streak(nums):
    if 5 not in nums:
        return False
    count = 0
    for num in nums:
        if num == 5:
            count += 1
            if count == 5:
                return True
            else:
                return False
        else:
            count = 0
    return False
```

(c)

```
 1  def streak(nums):
 2      if 5 not in nums:
 3          return False
 4      count = 0
 5      for num in nums:
 6          if num == 5:
 7              count += 1
 8              if count == 5:
 9                  return True
10          else:
11              count = 0
12      return False
```

(d)

```
 1  def streak(nums):
 2      if 5 not in nums:
 3          return False
 4      count = 0
 5      for num in nums:
 6          if num == 5:
 7              count += 1
 8              if count == 5:
 9                  return True
10      return False
```

Answer

(c)

Solution

Let us look at the correct answer:

```
 1  def streak(nums):
 2      if 5 not in nums:
 3          return False
 4      count = 0
 5      for num in nums:
 6          if num == 5:
 7              count += 1
 8              if count == 5:
 9                  return True
10          else:
11              count = 0
12      return False
```

If 5 is not even present in `nums`, then there is no point checking for a streak. This is the use of lines 2 and 3. The loop starting at line-5 goes through each number in `nums`. Whenever we come across a 5, `count` is incremented. If it is not a 5, then `count` is reset to zero. This happens in the else block at line-10. Notice that the else in line-10 is paired with the if at line-6. The nested if at line-8 checks if the `count` has become 5. If yes, then we have a streak. Finally, if the code manages to come to line-12, then there is certainly no streak. So, we have to return False there.

Let us look at why other answers are wrong:

(a) It is obvious because it is checking for only the sample cases given in the question.

(b) It is wrong because of the else block at line-10. If `count` is not 5, then it is immediately going to return False without waiting for it to reach 5 ever.

(d) It is wrong because `count` is never reset to zero. So, even if 5 doesn't occur consecutively for five times, it will return True.

Problem-2

Question-7 [2 marks]

Which of the following declarations are valid for a Tuple? It is a Multiple Select Question (MSQ).

(a) `t = (1,2,3)`

(b) `t = (10)`

(c) `t = 20,`

(d) `t = tuple([10])`

(e) `t = tuple((10))`

(f) `t = 1, 2, 3`

Answer

(a), (c), (d) and (f)

Solution

In option (b) `t=(10)` ,the type of t is `int` and option (e) `t = tuple((10))` gives error because `int` object is not iterable. Other than (b) and (e) all options, (a), (c), (d) and (f) are valid for a `tuple`.

Problem 3

Question-8 [2 marks]

Which of the following declarations are valid for a Set. It is a Multiple Select Question (MSQ).

(a) `s = {}`

(b) `s = set([1])`

(c) `s = {1}`

(d) `s = set()`

(e) `s = set((1))`

(f) `s = set(1)`

Answers

(b), (c) and (d)

Solution

In option (a), the type of `s` is `set` and Option (e) and (f) gives error because `int` object is not iterable. Other than (a), (e) and (f) all options (b), (c) and (d) are valid for a `set`.

Problem 4

Common data for questions 9 to 12.

Each student has a unique ID, which is an integer. For example, the student Ramanujan's ID is 1 while the student Ravi's ID is 2.

```
report_card = { 1:{
                'name': 'Ramanujan',
                'age': 18,
                'school': 'KV',
                'marks': {'Physics':75, 'Math':80, 'Chemistry':60}
                },
                2:{
                'name': 'Ravi',
                'age': 19,
                'school': 'KV',
                'marks': {'Physics':95, 'Math':70, 'Chemistry':90}
                }
              }
```

Question-9 [2 marks]

Choose the correct statement to print Ravi's marks in Math.

(a) `print(report_card['Ravi']['marks']['Math'])`

(b) `print(report_card[2][marks][Math])`

(c) `print(report_card[2]['marks']['Math'])`

(d) `print(report_card['2']['marks']['Math'])`

Answer

(c)

Solution

`print(report_card[2]['marks']['Math'])` is Correct way to access Ravi's marks in math.

Question-10 [2 marks]

Choose the correct statement to change the key `school` to `school_name` for both the students. It is a Multiple Select Question (MSQ).

(a)

```
for i in range(1,3):
    report_card[i]['school_name'] = report_card[i].popitem('school')
```

(b)

```
for i in range(1,3):
    report_card[i]['school_name'] = report_card[i].pop('school')
```

(c)

```
report_card[1]['school_name'] = report_card[1].popitem('school')
report_card[2]['school_name'] = report_card[2].popitem('school')
```

(d)

```
for i in range(1,3):
    report_card[i]['school_name'] = report_card[i].remove('school')
```

(e)

```
report_card[1]['school_name'] = report_card[1].pop('school')
report_card[2]['school_name'] = report_card[2].pop('school')
```

Answer

(b) and (e)

Solution

`pop(key)` method is used for removing items from the dictionary and after removing it return corresponding value . So option (b) and (e) are the correct way to change the key `school` to `school_name` for both the students. Here the return value of `pop('school')` will be assigned to the new key `school_name` .

Question-11 [2 marks]

Choose the correct statement to add a new entry of student `Ram` whose student ID is 3. It is a Multiple Select Question (MSQ). Assume that we go back to the dictionary given in the common data. So, we will be using `school` and not `school_name` in this particular question.

Entry	Details
name	Ram
age	20
school	KV
marks	Physics: 96 Math: 85 Chemistry: 75

(a)

```
report_card.add({3:{'name':'Ram','age': 20,'school':'KV','marks':
    {'Physics':96,'Math':85,'Chemistry':75}}})
```

(b)

```
report_card[3]={'name':'Ram','age': 20,'school':'KV','marks':{'Physics':96,'Math':85,'Chemistry':75}}
```

(c)

```
report_card={3:{'name':'Ram','age': 20,'school':'KV','marks':{'Physics':96,'Math':85,'Chemistry':75}}}
```

(d)

```
report_card.update({3:{'name':'Ram','age': 20,'school':'KV','marks':
    {'Physics':96,'Math':85,'Chemistry':75}}})
```

(e)

```
report_card[3].update({'name':'Ram','age': 20,'school':'KV','marks':
    {'Physics':96,'Math':85,'Chemistry':75}})
```

Answer

(b) and (d)

Solution

The `update()` method inserts the specified items to the dictionary. The specified items can be a dictionary, or an iterable object with key value pairs or we can assign a new value with a new key like `dict[key]=value` to add new items in the dictionary. Hence, option (b) and (d) are correct.

Question-12 [2 marks]

Choose the correct statement to delete the all records of student `Ravi`. It is a Multiple Select Question (MSQ).

(a)

```
report_card.pop(2)
```

(b)

```
report_card.popitem(2)
```

(c)

```
report_card.remove(2)
```

(d)

```
del report_card[2]
```

(e)

```
report_card.discard(2)
```

Answer

(a) and (d)

Solution

Option (a) and (d) are correct ways to remove all information about Ravi.
`report_card.popitem(2)` gives an error because `popitem()` takes no argument.
`report_card.remove(2)` gives error because remove() is not supported by
`dictionary.report_card.discard(2)` gives error because discard() is not supported by dictionary.