Week-3 Graded Assignment (Theory)

Problem 1

Question

What does the following code block print?

```
for i in 'We are in question one':
    if i == 'a' or i == 'e' or i == 'i' or i == 'o' or i == 'u':
        continue
    print(i, end = '')
```

(a) `W r n qstn n`

(b) `Wrnqstnn`

(c) `We are in question one`

(d) None of the above

Answer

(a)

Solution

The `for` loop checks whether each character is a vowel or not. If the character is a vowel (a', 'e', 'i', 'o', 'u'), the character is skipped and not printed on the console due to `continue` statement.

Problem 2

Question

Is it possible to get the below output without using any loop and without repeating the string `Hello Python!` in Python?

```
Hello Python!
Hello Python!
Hello Python!
Hello Python!
Hello Python!
Hello Python!
Hello Python!
Hello Python!
Hello Python!
Hello Python!

```

(a) True
(b) False

Answer

(a) True

Solution

Multiplying a string with a positive number `n` is equivalent to the string repeated `n` times one after another.

```
print(10*'Hello Python!\n')
```

Please find the below code for Question (3 & 4)

```
x = int(input())
i = 0
while x % 10**i != x:
    i = i + 1
```

Problem 3

Question

What will the variable `i` represent at the end of execution where `x` is a positive integer?

(a) Number of zeros in `x`
(b) Number of ones in `x`
(c) Number of digits in `x`
(d) Number of non-zero digits

Answer

(c) Number of digits of `x`

Solution

The above code is equivalent to the below

```
x = int(input())
i = 0
j = x % 10**i
while j != x:
    i = i + 1
    print("i = ", i)
    j = x % 10**i
```

The remainder of `x` is checked on each iteration. `x` is divided by 1, 10, 100,.. in each iteration, where the remainder of `x` will be last 1, 2, 3,.. digits respectively. When the divisor `10 ** i` becomes greater than `x`, the remainder `x % 10**i` will be same as `x`.

The variable, `i` counts the number of times the power of `10` is increased, which in other terms is the number of digits in `x`.

For example, `x` is 1987 then loop exits at `i` = 4 when the expression `1987 % 10**4` gives the value of `x`.

Problem 4

Question

What will be output if a negative value is given as input ?

(a) Number of digits in `x`

(b) Number of digits in `x` -1

(c) Number of digits in `x` +1

(d) Infinite loop

Answer

(d) Infinite loop

Solution

If `x` holds the negative integral value, the expression `x % 10**i` can be expressed as `(-a * 10**i + b) % 10**i` where `a` is `x // 10**i` and `b` is the remainder. The remainder `b` is always a positive integer and smaller than the absolute value of `x`. Hence, the remainder will never be equal to `x`, and the loop continues infinitely.

For example, if `x` is equal to -10 then `x % 6` will be expressed as `(-2 * 6 + 2) % 6` which gives the remainder as 2.

Problem 5

Question

How many times do the break statements get executed? It is a Numerical Type Question (NAT).

```
1 for i in range(10):
2     for j in range(10):
3         break
4     break
```

Answer

2

Solution

First `break` statement (in line-3) makes the program to exit from the inner loop regardless of how many iterations remaining. Hence, first `break` will exits from the inner for-loop and next `break` (in line-4) will make the program to exit from the outer for-loop. Thus, `break` is executed two times in the above code snippet.

Problem 6

Question

```
for i in range(10, 0, 1):
    print(i)
```

A programmer wants to print a decreasing sequence. How many times does the `print` statement get executed? And why?

Select the most appropriate statement

(a) One time because `i` takes only the value 10 and thereafter it will be decremented
(b) One time because `i` takes only the value 9 and thereafter it will be decremented
(c) `print` statement will not be executed due to invalid end points
(d) `print` statement will not be executed due to incompatible step size

Answer

(d) `print` statement will not be executed due to incompatible increment

Solution

The `start`, `end` and `step` parameters in `range()` are 10, 0 and 1. The variable `i` starts from 10. It should be incremented at each iteration by 1 and should end at 0, which is not possible. The `range(10, 0, 1)` returns no values, so the loop does not run and therefore `print` statement will not be executed.

Problem 7

Question

```
for i in range(1231, -12420, -7):
    print(i)
```

How many times the `print` statement get executed? It is a Numerical Type Question (NAT).

Answer

1951

Solution

The variable `i` starts at 1231 and ends at -12419. This can be counted using another variable say `c`, as shown in the below code:

```
c = 0
for i in range(1231, -12420, -7):
    c += 1
    print(i)
print(c)
```

The final printed value will give the number of times the `print` get executed.

An alternate approach will be to use `math.ceil()` function.

```
import math
print(math.ceil((-12420-1231)/-7))
```

Problem 8

Code-1

```
x = 0
x_ = 1
for i in range(10):
    x, x_ = x_, x + x_
print(x)
```

Code-2

```
x = 0
x_ = 1
for i in range(10):
    x = x_
    x_ = x + x_
print(x)
```

Question

Code-1 and Code-2 will return the same value.

(a) True
(b) False

Answer

(b) False

Solution

In Code-1 the value of `x` and `x_` are assigned simultaneously from `x_` and `x + x_` respectively. Thus, both variable store different values, the pattern leads to the Fibonacci series since the initial value are 0 and 1 for `x` and `x_`.

In Code-2 the value of `x_` is assigned to `x` and `x_` is assigned with the value of expression `x + x_` which is nothing but the twice the value of `x`. Hence, it prints the sequence of value powers of 2.