Congratulations! You passed!
TO PASS 70% or higher

Keep Learning

GRADE
100%

Lecture 4 Quiz

LATEST SUBMISSION GRADE
100%

1. The following expression is true when rnatype is 'ncRNA' and length is at least 200, or rnatype is 'ncRNA' and length is 22: **1 / 1 point**

```
(rnatype is 'ncRNA' and length>=200) or (rnatype is 'ncRNA' and length==22)
```

What Boolean expression below represents a negation of the above expression?

- rnatype is not 'ncRNA' and (length <200 or length != 22)
- rnatype is not 'ncRNA' or (length <200 and length != 22)
- (rnatype is not 'ncRNA' and length < 200) and (rnatype is not 'ncRNA' and length != 22)
- (rnatype is not 'ncRNA' and length < 200) or (rnatype is not 'ncRNA' and length != 22)

Correct

2. For what values of the variable fold would the following code print 'condition B'? **1 / 1 point**

```
if fold > 2 : print('condition A')

elif fold>100: print('condition B')

if fold> 2 or fold<2 : print('condition A')

else : print('condition B')
```

- if fold is bigger than 100 or fold is 2
- if fold is less than 2
- never
- if fold is 2

Correct

3. How many times will Python execute the code inside the following while loop? **1 / 1 point**

```
i=1
while i< 2048 :
    i=2*i
```

- 1024
- 12
- 11
- 2048

Correct

4. What sequence of numbers does the range(1,-23,-3) expression evaluate to? **1 / 1 point**

- 1, -2, -5, -8, -11, -14, -17, -20
- 1, -1, -3, -5, -7, -9, -11, -13, -15, -17, -19, -21
- -23, -20, -17, -14, -11, -8, -5, -2
- -23, -20, -17, -14, -11, -8, -5, -2, 1

Correct

5. A substring in programming represents all characters from a string, between two specified indices. Given a variable string called seq, a student writes the following program that will generate all nonempty substrings of seq: **1 / 1 point**

```
for i in range(len(seq)) :     # line 1
    for j in range(i) :        # line 2
        print(seq[j:i])     # line 3
```

Which of the following changes make the above program correct?

A. Program is correct as it is.

B. Change line 1 to:

for i in range(len(seq)+1) :

C. Change line 3 to:

print(seq[j:i+1])

D. Change line 2 to:

for j in range(i+1) :

- (●) Only B
- ( ) Only A
- ( ) Only C
- ( ) Only D

✓ **Correct**

6. While and for loops are equivalent: whatever you can do with one you can do with the other. Given the for loop written by the student in problem 5, which of the following while loops are equivalent to it: **1 / 1 point**

A.

```
i=0
while i<len(seq) :
    j=0
    while(j<i) :
        print(seq[j:i])
```

B.

```
i=1
while i<len(seq) :
    j=1
    while(j<i) :
        print(seq[j:i])
        j=j+1
    i=i+1
```

C.

```
i=0
while i<len(seq) :
    j=0
    while(j<i) :
        print(seq[j:i])
        j+=1
    i+=1
```

D.

```
i=0
while i<len(seq)+1 :
    j=0
    while(j<i+1) :
        print(seq[j:i])
        j=j+1
    i=i+1
```

E.

```
i=1
while i<len(seq)+1 :
    j=1
    while(j<i+1) :
        print(seq[j:i])
```

F.

```
i=0
while i<len(seq) :
```

```
    j=i
    while(j>0) :
            print(seq[j:i])
            j=j+1
i=i+1
```

- ○ B only
- ○ B, and C only
- ◉ C only
- ○ A, C, D and E only

✓ **Correct**

7. A student writes a program that for any two lists L1 and L2, computes a list L3 that contains only the elements that are common between the two lists *without duplicates*. Which following statement makes the following portion of code that computes L3 correct:

1 / 1 point

```
L3 = []                 # line 1
for elem in L1:         # line 2
 if elem in L2:         # line 3
  L3.append(elem)       # line 4
```

- ○ The following two lines are introduced with the correct indentation after line 2:

  ```
  if elem in L3:
    pass
  ```

- ○ Add the following line (with the correct indentation) between lines 2 and 3:

  ```
  if elem not in L3:
  ```

- ◉ Change line3 to be:

  ```
  if elem in L2 and elem not in L3:
  ```

- ○ Change line 4 to:

  ```
  L3=L3+elem
  ```

✓ **Correct**

8. Study the following two Python code fragments:

1 / 1 point

Version 1.

```
d = {}
result = False
for x in mylist:
        if x in d:
           result=True
           break
        d[x] = True
```

Version 2.

```
d = {}
result = False
for x in mylist:
       if not x in d:
          d[x]=True
          continue
       result = True
```

Both versions should determine if there is any element that appears more than once in the list mylist. If there is such an element than the variable result should be True, otherwise it should be False. For instance, if mylist=[1,2,2,3,4,5] the result variable should be True.

Which of the following statements is True for any value of the list mylist after the execution of both versions of code?

- Both the result and d variables have the same value.
- Neither Version 1 or Version 2 are computing the value of the result variable correctly.
- Version 2 is not computing the result variable correctly.
- The value of the result variable is the same, but the variable d is different.

✓ **Correct**

9. Study the following if statement:

1 / 1 point

```
if x>10 or x<-10: print('big')
elif x>1000000: print('very big')
elif x<-1000000: print('very big')
else : print('small')
```

For what values of x will the above code print 'very big'?

- For x < -1000000
- For no value
- For x > 1000000
- For x > 1000000 or x < -1000000

✓ **Correct**

10. What will be the value of the variable i after the following code is executed:

1 / 1 point

```
i = 1
while i < 100:
        if i%2 == 0 : break
        i += 1
else:
    i=1000
```

- 2
- 1
- 99
- 98

✓ **Correct**