Section 1

Question 1 (1 point)

What would this code print when run?

```
x = [1, 2, 3]
y = x
y.append(4)
print(x)
```

- ◯ `[4, 1, 2, 3]`
- ◯ There is an error when this code is run
- ◯ `[4]`
- ◯ `[1, 2, 3, 4]`
- ◯ `[1, 2, 3]`

Question 2 (1 point)

What would this code print when run?

```
x = 0

def foo():
    global x
    x = 'Hello, world!'

foo()
print(x)
```

- `0`
- There is an error when this code is run
- `Hello, world!`
- Some other answer not given here

Question 3 (1 point)

What would this code print when run?

```
x = 0

def foo():
    x = 'Hello, world!'

foo()
print(x)
```

- There is an error when this code is run
- `Hello, world!`
- Some other answer not given here
- `0`

Question 4 (1 point)

What would this code print when run?

```
x = 0

def foo():
    global x
```

```
    x = 'Hello, world!'
print(x)
```

- 0
- There is an error when this code is run
- Hello, world!
- Some other answer not given here

Question 5 (1 point)

What would this code print when run?

```
x = 0

def foo():
    global x
x = 'Hello, world!'

foo()
print(x)
```

- Some other answer not given here
- Hello, world!
- There is an error when this code is run
- 0

Question 6 (1 point)

What does the following code draw when run?

```
import turtle

L = [(0, 0), (50, 0), (50, 75), (0, 0), (50, 75)]
for coord in L:
    x = coord[0]
    y = coord[1]
    turtle.goto(x, y)
```

- Some other answer not given here
- An icosahedron
- A square
- A pentagon
- A triangle

Question 7 (1 point)

What does this code print when it is run?

```
def foo(mydict):
    mydict['ghi'] = 789

D = {'abc': 123, 'def': 456}
foo(D)
print(D)
```

- ◯ `{'abc', 'def'}`
- ◯ `{'abc', 'def', 'ghi'}`
- ◯ `{'abc': 123, 'def': 456}`
- ◯ `{'abc': 123, 'def': 456, 'ghi': 789}`

Section 2

Use the following definitions to answer the questions in this section.

```
i = 3
L = [7, 7, 2, 3, 6, 7, 9]
D = {6: 'spy3', i: 'om', 7: 'baz'}
T = (2, 1, 5, 7, 4)
```

Question 8 (1 point)

What would this code print when run?

```
print(len(L), len(D))
```

- ◯ 7 6
- ◯ 7 12
- ◯ 6 3
- ◯ 6 6
- ◯ 7 3

Question 9 (1 point)

How many of these variables have immutable types?

- 3
- 4
- 1
- 2
- 0

Question 10 (1 point)

How many of the following evaluate to `True`?

```
i in L
7 in T
'baz' in D
'6' not in D
```

- 4
- 3
- 1
- 0
- 2

Question 11 (1 point)

How many of the following evaluate to `True`?

```
3 in D
i != L[3]
len(T[3:-1]) > 1
T[L[len(D[i])]] == len(T)
```

- 0
- 2
- 3
- 1
- 4

Question 12 (1 point)

What would be printed if this code were run with the above definitions?

```
D2 = { }
for n in L:
    D2[n] = True
print(len(D2))
```

- 7
- Nothing - there is an error when this code is run
- 6
- 5
- Some other answer not given here

Question 13 (1 point)

What is `L[2:-2]`?

- `[2, 3]`
- `[7, 2, 3, 6, 7]`
- Some other answer not given here
- `[2, 3, 6]`
- `[7, 2, 3, 6]`

Question 14 (1 point)

What is `T[:3]`?

- Some other answer not given here
- `(2, 1, 5)`
- `(2, 1, 5, 7)`
- `[2, 1, 5, 7]`
- `[2, 1, 5]`

Question 15 (1 point)

What would this code print when run with the above definitions?

```
for n in range(0, len(T), 3):
    print(T[n])
```

- 7, then 7
- Some other answer not given here
- 0, then 3
- 2, then 7

Question 16 (1 point)

What would this code print when run with the above definitions?

```
for n in range(0, len(T), 3):
    print(n)
```

- 2, then 7
- 0, then 3
- Some other answer not given here
- 7, then 7

Question 17 (1 point)

What would this code print when run with the above definitions?

```
for n in range(0, len(T), 3):
    print(T[i])
```

- Some other answer not given here
- 2, then 7
- 0, then 3
- 7, then 7

Section 3

The program in this section should read data from a datafile into a 2D list of lists and manipulate it. The datafile consists of tab-separated fields, and each line of data has the same number of fields. The end of the datafile is denoted by the sentinel `END`. A sample datafile is below.

```
a       b       c
d       e       f
g       h       i
END
```

When the sample datafile is given as input to the program in this section, by typing

```
    spy3 section3.py < datafile
```

as you've seen before, the output from the program is:

```
a b c
d e f
g h i

g d a
h e b
i f c
```

Starting with the following code:

AAA

```
def print2d(L):
    for row in L:
```

```
        for col in row:
            print(BBB, CCC)
        print()

def readdata():
    L = []
    while DDD:
        line = input()
        if EEE:
            FFF
        fields = GGG
        L.append(fields)

def rotatecw(oldL):
    newL = []
    for oldcol in range(len(HHH[0])):
        newrow = []
        for oldrow in range(III):
            newrow.append(JJJ)
        newL.append(newrow)
    KKK

# read in and print data
LLL
print2d(L)
print()

# rotate clockwise by 90 degrees and print
MMM
print2d(L)
```

Question 18 (1 point)

What should AAA be replaced with?

- Nothing
- `SENTINEL == END`
- `SENTINEL = END`
- `SENTINEL == 'END'`
- `SENTINEL = 'END'`

Question 19 (1 point)

What should BBB be replaced with?

- `col`
- `L[row][col]`
- `L[row]`
- `L[col]`
- `row`
- `L[col][row]`

Question 20 (1 point)

What should CCC be replaced with?

- `end='\t'`
- `end='\n'`
- Nothing
- `end='/t'`
- `end=' '`

Question 21 (1 point)

What should `DDD` be replaced with?

- `input()`
- `True`
- `0`
- `False`
- `line != 'END'`

Question 22 (1 point)

What should EEE be replaced with?

- `line = SENTINEL`
- `'END'`
- `line = 'END'`
- `line == SENTINEL`
- `line == 'END'`
- `SENTINEL`

Question 23 (1 point)

What should `FFF` be replaced with?

- `return L`
- `return`
- `continue`
- `break`
- `exit()`

Question 24 (1 point)

What should `GGG` be replaced with?

- `split()`
- `line.split('/t')`
- `line.split()`
- `line.split('\t')`
- `split('/t')`
- `split('\t')`

Question 25 (1 point)

What should `HHH` be replaced with?

- `newrow`
- `newL`
- `L`
- `oldL`

Question 26 (1 point)

What should `III` be replaced with?

- `len(oldL), -1, -1`
- `len(oldL)-1, 0, -1`
- `len(oldL)-1, -1, -1`
- `len(oldL)-1, 0`
- `len(oldL)-1, -1`

Question 27 (1 point)

What should `JJJ` be replaced with?

- `oldL[oldrow]`
- `oldL[oldcol]`
- `oldL[oldrow][oldcol]`
- `oldL[oldcol][oldrow]`
- `oldL`

Question 28 (1 point)

What should `KKK` be replaced with?

- `return oldL`
- Nothing
- `return L`
- `return`
- `return newL`

Question 29 (1 point)

What should `LLL` be replaced with?

- `readdata()`
- `readdata`
- `L = readdata`
- `L = readdata()`

Question 30 (1 point)

What should `MMM` be replaced with?

- `L = rotatecw`
- `rotatecw()`
- `L = rotatecw()`
- `rotatecw(L)`
- `rotatecw`
- `L = rotatecw(L)`

Section 4

The program in this section should read student data from a variant of the *completely fictitious* classlist data you've seen me use before. The classlist data begins with a line containing the number of students, followed by that many lines of data, one student per line. Fields in each line are separated by commas. Here is a sample classlist:

```
4
784799,Aycock,John,ARKY,toy.blanche@hotmail.com
954552,Loblaw,Bob,ENEL,carole@hotmail.com
005867,Kile,Alice,STAT,ali.vandusen@yahoo.com
676502,Nicely,Jane,GEOG,virgie@gmail.com
```

The program should calculate and output the frequency that letters are used in students' first names. Uppercase and lowercase letters should be treated as equivalent. For example, the output for the above classlist file should be:

```
j 2
o 2
h 1
n 2
b 2
a 2
l 1
```

```
i 1
c 1
e 2
```

Start with the following code:

```
n = AAA

D = BBB

for i in CCC:
    line = input()
    fields = DDD
    firstname = EEE
    for ch in firstname:
        FFF
        if ch in D:
            GGG
        else:
            HHH

for ch in D:
    print(III)
```

The code is run by typing

```
spy3 section5.py < classlist
```

as you've seen in class.

Question 31 (1 point)

What should AAA be replaced with?

- `input`
- `int(input())`
- `input()`
- `int(input)`
- 4

Question 32 (1 point)

What should BBB be replaced with?

- `[ ]`
- `( )`
- `{ }`
- `input()`
- 4

Question 33 (1 point)

What should CCC be replaced with?

- `range(n-1)`
- `n`
- `D`
- `range(n)`
- `range(n+1)`

Question 34 (1 point)

What should `DDD` be replaced with?

- `line.split('\t')`
- `split(',')`
- `line.split()`
- `line.split('/t')`
- `line.split(',')`
- `split()`

Question 35 (1 point)

What should `EEE` be replaced with?

- `fields[1]`
- `fields[2]`
- `fields`
- `input()`
- `line`
- `fields[3]`

Question 36 (1 point)

What should FFF be replaced with?

- `ch.islower()`
- `ch = ch.lower()`
- Nothing
- `ch.lower()`
- `ch = lower()`
- `ch = ch.islower()`

Question 37 (1 point)

What should GGG be replaced with?

- `0`
- `D[ch] = D[ch] + 1`
- `1`
- `ch = ch + 1`
- `D[ch] + 1`
- `D[ch] = 1`

Question 38 (1 point)

What should `HHH` be replaced with?

- `D[ch] = D[ch] + 1`
- `D[ch] = 1`
- `D[ch] + 1`
- `1`
- `ch = ch + 1`
- `0`

Question 39 (1 point)

What should `III` be replaced with?

- `ch`
- `D[ch], ch`
- `D`
- `ch, D[ch]`
- `D[ch]`

Section 5

Use the following code to answer the questions in this section:

```
p = input()
for i in p:
    if i == '0':
        n = 0
    elif i == '1':
        n = n + 1
    elif i == 'x':
        break
    elif i == 'p':
        print(n)
```

Question 40 (1 point)

The input p results in the output...

- `1`
- `0`
- There is an error when that input is entered
- There is no output

Question 41 (1 point)

The input `0111p` results in the output...

- There is no output
- There is an error when that input is entered
- `0`
- `3`

Question 42 (1 point)

The input `0111xp` results in the output...

- `3`
- `0`
- There is an error when that input is entered
- There is no output

Question 43 (1 point)

The input 0111110p results in the output...

- ◯ There is an error when that input is entered
- ◯ `0`
- ◯ `5`
- ◯ There is no output

Section 6

With Picologo being such an up-and-coming language, there's naturally a demand to create tools that work with the language. The program in this section will read a Picologo program as input, and output the corresponding Python turtle module program. For example, given the Picologo program:

```
f
b
l
r

u
d

x
```

the output should be:

```
import turtle
turtle.fd(1)
turtle.bk(1)
turtle.lt(1)
turtle.rt(1)

turtle.pu()
turtle.pd()
```

Assume that the Picologo program has no errors or repeat counts. The program is run as

```
spy3 section6.py < input.pico
```

Starting with the following:

AAA

```
TRANSLATION = {
    BBB
}

while CCC:
    line = input()
    if DDD:
        print(TRANSLATION[line])
    elif line == 'x':
        EEE
    else:
        print(line)
```

Question 44 (1 point)

What should AAA be replaced with?

- `import turtle`
- `print('import turtle')`
- Nothing

Question 45 (1 point)

What should BBB be replaced with?

```
'turtle.fd(1)': 'f',
'turtle.bk(1)': 'b',
'turtle.lt(1)': 'l',
'turtle.rt(1)': 'r',
'turtle.pu()': 'u',
'turtle.pd()': 'd',
```

```
'f': 'turtle.fd(1)',
'b': 'turtle.bk(1)',
'l': 'turtle.lt(1)',
'r': 'turtle.rt(1)',
'u': 'turtle.pu()',
'd': 'turtle.pd()',
```

```
'f': 'fd(1)',
'b': 'bk(1)',
'l': 'lt(1)',
'r': 'rt(1)',
'u': 'pu()',
'd': 'pd()',
```

```
'f': turtle.fd(1),
'b': turtle.bk(1),
'l': turtle.lt(1),
'r': turtle.rt(1),
'u': turtle.pu(),
'd': turtle.pd(),
```

Question 46 (1 point)

What should CCC be replaced with?

- `True`
- `input()`
- `False`
- `0`

Question 47 (1 point)

What should `DDD` be replaced with?

- `line not in TRANSLATION`
- `line in TRANSLATION`
- `line == TRANSLATION`
- `line = TRANSLATION`
- `line != TRANSLATION`

Question 48 (1 point)

What should `EEE` be replaced with?

- `break`
- `return`
- Nothing
- `print(TRANSLATION[line])`
- `continue`

Submit Quiz *0 of 48 questions saved*