When the following is entered into the Python shell, what does Python print?

1) >>> course = [1, 2, 2]
   a) course       b) NameError       c) [1, 2, 2]       d) nothing is printed

2) >>> course
   a) 'course'     b) course         c) [1, 2, 2]       d) NameError

3) >>> [1, 2, 2]
   a) 'course'     b) course         c) [1, 2, 2]       d) nothing is printed

4) >>> 'course'
   a) 'course'     b) course         c) [1, 2, 2]       d) nothing is printed

5) >>> course[1]
   a) 1            b) 2             c) [1, 2, 2]       d) IndexError

What is the result when the following Python code is executed?

```python
>>> greeting = 'Ciao'
>>> greeting = greeting * 2
>>> greeting
```
   a) TypeError       b) NameError       c) SyntaxError       d) 'Ciao'
   e) 'CiaoCiao'

Given the following Python code

```python
>>> a = 'carrot'
>>> b = 'radish'
```
What is the result when the following Python code is executed?

```python
7) a[:3] + b[4]
``` 
   a) 'cars'       b) 'rots'       c) 'roti'       d) TypeError       e) IndexError
8)
```python
count = 0
for letter in a:
    if letter == 'r':
        count += 1
print(count)
```

a) 1      b) 2      c) 5      d) 6      e) LogicError

9)
```python
len(a) == len(b)
```

a) 5      b) 6      c) True     d) False    e) TypeError

10)
```python
type(len(a) < len(b))
```

a) int    b) float   c) str      d) list     e) Boolean

(11) What will be the result when the following Python code is executed?
```python
>>> days_worked = int(input())
20
>>> type(days_worked)
```

a) int    b) float   c) str      d) list     e) Boolean

(12) What will be the result when the following Python code is executed?
```python
>>> shirts = 10
>>> green = shirts // 2
>>> yellow = shirts - green
>>> cost = green * 10 + 20 * yellow
>>> cost
```

a) 750    b) 750.0    c) 150    d) 150.00    e) 100

(13) What is the result when the following Python code is executed?
```python
>>> s = 'U of O'
>>> s[0] + s[-1]
```

a) 6      b) 'UO'     c) 'Oregon'    d) ' '    e) 'U'
(14-19) Given the following Python code:

```python
1 def q14(x, y, z):
  
  '''
  (number, number, number) -> ??
  '''
  wmax = min(x, y, z)
  return wmax

10 def q15(a, b, c, d, e):
  
  '''(number, number, number, number, number) -> ??
  '''
  max1 = q14(a, b, c)
  max2 = min(d, e)
  print('Maximum transport weight is:', max(max1, max2))
  return None

>>> q15(1, 2, 3, 4, 5)

14) At line 15 the value of a is
   a) undefined       b) 1       c) 3       d) 5

15) After line 15 executes, the value of max1 is
   a) undefined       b) None     c) 1       d) 2       e) 3

16) After line 16 executes, the value of max2 is
   a) undefined       b) None     c) 1       d) 4       e) 5

17) After line 17 executes, the value printed in the Shell after the string label is
   a) 1       b) 2       c) 3       d) 4       e) 5

18) When q15 is done executing,
   a) maximum transport weight has been printed  b) None is returned
c) maximum weight and None are returned       d) both a) and b)

19) Complete the type contracts for q14 and q15:
   a) number, None     b) str,number    c) str, None     d) None, None
20) What will be the result of executing the following code:

```python
>>> q20('CIS122') == q21('CIS122')
```

a) True  
b) False  
c) True, False  
d) 'e' and 'E'  
e) 'e' or 'E'

21) When q20('CIS122') is executing, the last value of ch will be

a) 'E'  
b) 'e'  
c) 'S'  
d) '1'  
e) '2'

22) In q21, what is the type of the variable lowere?

a) int  
b) str  
c) Boolean  
d) list  
e) None

23) In q20, indenting `return True` (to line up with `if`s) would

a) fix a bug  
b) cause a bug for some input strings  
c) cause a bug for all input strings  
d) cause an infinite loop

24) In q20, astring is

a) parameter  
b) argument  
c) function name  
d) Boolean expression
(25-28) Given the following UNTESTED Python code:

```python
def q25(astring):
    """(str) -> Boolean
    Returns True if astring includes one or more special characters.
    ""
    special = '!@#$%^&'
    for c in astring:
        if c in special:
            return True
    return False
```

25) What is the result of executing `>>> q25('CIS122!')`

a) 'CIS122!' b) '!' c) 'C' d) True e) False

26) What is the result of executing `>>> q25('!CIS122')`

a) 'CIS122!' b) '!' c) 'C' d) True e) False

27) What line(s) of code need(s) to be changed to fix the bug?

a) 1 b) 1, 2 c) 7 d) 12 e) 7, 12

28) Choose the best set of test cases for q25:

<table>
<thead>
<tr>
<th>a)</th>
<th>b)</th>
<th>c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>q25('122!')</td>
<td>q25('abc')</td>
<td>q25('CIS122!')</td>
</tr>
<tr>
<td>q25('#122')</td>
<td>q25('CIS122')</td>
<td>q25('CIS111')</td>
</tr>
<tr>
<td>q25('CIS&amp;122')</td>
<td>q25('122CIS')</td>
<td>q25('CIS115')</td>
</tr>
<tr>
<td>q25('122')</td>
<td>q25('ab')</td>
<td>q25('CIS!')</td>
</tr>
<tr>
<td>q25('$')</td>
<td>q25('a%b')</td>
<td>q25('122')</td>
</tr>
</tbody>
</table>
(29-31) Given the following UNTESTED Python code:

```python
def calc_avg(dataset):
    '''(list of ints) -> float

    returns average of values in input list, but zeros do not count at all
    >>> calc_avg([2, 3])
    2.5
    >>> calc_avg([2, 0, 3])
    2.5
    '''
    count = 0
    total = 0
    for value in dataset:
        if value != 0:
            total += value
            count += 1

    avg = total / count
    return avg
```

29) Executing `calc_avg` will <??> return the correct result.
   a) always   b) sometimes   c) never   d) cannot be determined

30) Which call to execute `calc_avg` will cause a runtime error?
   a) `calc_avg([2, 0, 3])`   b) `calc_avg([])`   c) `calc_avg([0])`
   d) a) and b)   e) b and c)

31) The runtime error will be
   a) NameError   b) IndexError   c) ZeroDivisionError   d) TypeError
   e) SyntaxError

(32) Given the following Python code:

```python
def q32(n):
    '''(int) -> ??

    Exam function.
    '''
    return (n % 2) == 1
```

Which best describes what `q32` does?
   a) returns `n` divided by 2   b) returns remainder of `n` divided by 2
   c) returns `n` if `n` is an even number   d) returns `True` if `n` is an even number
   e) returns `True` if `n` is an odd number
(33) Given the following Python code:

```python
def middle(L):
    '''(list) -> list item

    returns item from the middle of input list L, when L has an odd length, else return 999999.

>>> middle([0, 2, 4, 6, 8])
4
>>> middle([0, 2, 4, 6])
999999
>>> middle(['hello', 'goodbye', 'ciao'])
'goodbye'
>>> middle([])
999999
'''

Arrange the following lines of code in order to implement `middle` (ignore whitespaces):

```python
1 return L[num // 2]
2 num = len(L)
3 if odd(num):
4 else:
5 return 999999
```

a) 2,3,1,4,5  b) 3,2,4,1,5  c) 2,3,4,1,5  d) 3,2,4,1,5  e) 3,5,4,2,1

(34) Given the following Python code:

```python
def mySum(numlist):
    '''(list) -> number

    returns sum of numbers in numlist

>>> mySum([1, 2, 3])
6
>>> mySum([-5, 0, 5, 5])
5
>>> mySum([])
0
'''

Arrange the following lines of code in order to implement `mySum` (ignore whitespaces):

```python
1 accumulator = 0
2 for num in numlist:
3    accumulator += num
4 return accumulator
```

a) 2,1,3,4  b) 1,3,2,4  c) 2,3,1,4  d) 3,2,1,4  e) 1,2,3,4
Given the following Python code:

```python
1 def q35(compound):
2     '''(str) -> str
3     Exam function. '''
4     if compound == 'H2O':
5         cname = 'water'
6     elif compound == 'NH3':
7         cname = 'ammonia'
8     elif compound == 'CH4':
9         cname = 'methane'
10    else:
11        cname = 'unknown'
12    return cname
```

(35) Function parameter(s) is/are

a) q35, compound  b) q35  c) compound  d) str, str  e) def

When the following expression is entered in the Python Shell

```python
>>> q35('NH3')
```

36) the argument(s) is/are

a) 'NH3'  b) compound  c) 'H2O'  d) str, str  e) None

37) the expression on line 6 evaluates to

a) True  b) true  c) False  d) false  e) this line of code is not executed

38) At line 15 the value of cname is

a) 'water'  b) 'ammonia'  c) 'methane'  d) 'unknown'  e) cname
(39-40) Given the following Python code:

```python
1 def fbkup(fname):
2     '''(str) -> ??
3     Exam function.
4     '''
5     with open(fname, 'r') as examf:
6         examfli = examf.readlines()
7         newf = fname[:suffix_pos] + '-bkup' + suffix
8     with open(newf, 'w') as exambkupf:
9         for item in examfli:
10            exambkupf.write(item)
11     return newf

>>> fbkup('majors.txt')

39) Complete the type contract for `fbkup`:

a) file b) list c) str d) int e) None

40) After line 12 is executed, the value of `newf` is

a) 5 b) 6 c) 'majors.txt' d) 'majors' e) '.txt'

(41) What will be the result of executing the following Python code:

```python
>>> stuff = [2, 25, 80, 12]
>>> stuff[stuff[0]] = stuff[3]
>>> stuff
```

```plaintext
a) [2, 25, 80, 12] b) [2, 2, 12, 12] c) [2, 25, 12, 12] d) [] e) None
```

(42) What will be the result of executing the following Python code:

```python
>>> list1 = ['a', 'b', 99]
>>> list1 = list1.reverse()
>>> list1
```

```plaintext
a) ['a', 'b', 99] b) [99, 'b', 'a',] c) [99, 'a', 'b'] d) [] e) None
```
(43) What will be the result of executing the following Python code:

```python
>>> testli = [5, 4, 3]
>>> testli.sort()
>>> testli
```

*a) [5, 4, 3]  b) [3, 4, 5]  c) [5, 3, 4]*

d) []  e) None

44) If 'anyfile.txt' is a text file where the first four lines are a header describing the information in the file, which code will move the file pointer past the file header?

```python
with open('anyfile.txt', 'r') as examf:
```

*a) examf.read()  b) examf.readline()  c) for i in range(4):
    examf.readline()  d) examf.readlines()  e) examf.write()*

(45) What will be the result of executing the following Python code:

```python
def best(greeting):
    '''(str) -> None
    print a greeting
    ...
    start = greeting.find('Best')
    print(greeting[start:])
    return None

>>> best('CIS122: Best wishes for a pleasant summer. ')
```

*a) Best wishes for a pleasant summer.  b) Best wishes for a pleasant summer.
 c) Best wishes for a pleasant summer.  d) Best wishes for a pleasant summer.
 e) Best wishes for a pleasant summer.*