1. TRUE / FALSE QUESTIONS

__T__ You can have more than one except clause in a `try/except` statement.

__T__ The `ZeroDivisionError` exception is raised when the program attempts to perform the calculation `x/y` if `y = 0`.

__T__ An exception handler is a piece of code that is written using the `try/except` statement.

__F__ The `else` suite in a `try/except` statement executes only if a statement in the try suite raises an exception.

__F__ The `finally` suite in a `try/except` statement executes only if no exceptions are raised by statements in the try suite.

__F__ Lists in Python are immutable.

__T__ Invalid indexes do not cause slicing expressions to raise an exception.

__F__ The index of the first element in a list is 1, the index of the second element is 2, and so forth.

__T__ The index -1 identifies the last element in a list.

__T__ In slicing, if the end index specifies a position beyond the end of the list, Python will use the length of the list instead.

2. COMPLETION QUESTIONS: Fill in the blanks.

a) A(n) __TRY__ block includes one or more statements that can potentially raise an exception.

b) The built-in function __LEN__ returns the length of a sequence.

c) A(n) __slice__ is a span of items that are taken from a sequence.

d) Lists are __mutable__, which means their elements can be changed in a program.

e) The __ValueError__ exception is raised when a search item is not in the list being searched.

f) A(n) __traceback__ gives information about the line number(s) that caused an exception.

3. ALGORITHM WORKBENCH QUESTIONS

a) Write a program that opens an output file with filename `things.txt`, writes the name of an animal, a fruit, and a country to the file on separate lines, then closes the file.

b) What will the following code display?

```
try:
      x = float('abc123')
      print('The conversion is complete.')
except IOError:
      print('This code caused an IOError.')
except ValueError:
      print('This code caused a ValueError.')
print('The end.')
```

c) What will the following code display?

```
try:
      x = float(abc123)
      print(x)
      except ValueError:
      print('This code caused a ValueError.')
except TypeError:
      print('This code caused a TypeError.')
except NameError:
      print('This code caused a NameError.')
print('The end.')
```

d) What will be the output after the following code is executed and the user enters **75** and **0** at the first two prompts?

```
def main():
    try:
        total = int(input("Enter total cost of items? "))
        num_items = int(input("Number of items "))
        average = total / num_items
    except ZeroDivisionError:
        print('ERROR: cannot have 0 items')
    except ValueError:
        print('ERROR: number of items cannot be negative')
main()
```

a) ERROR: cannot have 0 items
b) ERROR: number of items can't be negative
c) 0
d) Nothing; there is no **print** statement to display **average**.

e) What will be the output after the following code is executed and the user enters **75** and **-5** at the first two prompts?

```
def main():
    try:
        total = int(input("Enter total cost of items? "))
        num_items = int(input("Number of items "))
        average = total / num_items
    except ZeroDivisionError:
        print('ERROR: cannot have 0 items')
    except ValueError:
        print('ERROR: number of items cannot be negative')
main()
```

a) ERROR: cannot have 0 items
b) ERROR: cannot have 0 items
c) ERROR: number of items can't be negative
d) ERROR: number of items can't be negative
e) Nothing; there is no print statement to display average. The ValueError will not catch the error.

f) What is displayed when the following program is run?

```
def main():
    try:
        f()
        print("After the function call")
    except ZeroDivisionError:
        print("Divided by zero!")
    except:
        print("Exception")

def f():
    print(1 / 0)

main() # Call the main function
```

g) Assume the `names` variable references a list of strings. Write code that determines whether 'Ruby' is in the `names` list. If it is, display the message 'Hello Ruby'. Otherwise, display the message 'No Ruby'.

h) Assume the `numbers` variable references a list of strings. Write code that determines that determines the average of the numbers and displays the result with two digits after decimal point.

MULTIPLE CHOICE QUESTIONS

4. What are the data items in a list called?

a) data

b) elements

c) items

d) values

5. When working with multiple sets of data, one would typically use a(n)_____.

a) list

b) tuple

c) nested list

d) sequence

6. This is a number that identifies an item in a list.

a) element

b) index

c) bookmark

d) identifier

7. The primary difference between a tuple and a list is that

a) you don't use commas to separate elements in a tuple

b) a tuple can only include string elements

c) a tuple cannot include lists as elements

d) once a tuple is created, it cannot be changed

8. Which list will be referenced by the variable number after the following code is executed?

```
number = range(0, 9, 2)
```

a) `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`

b) `[1, 3, 5, 7, 9]`

c) `[2, 4, 6, 8]`

d) `[0, 2, 4, 6, 8]`

9. What will be the value of the variable list after the following code executes?

```
list = [1, 2]
list = list * 3
```

a) `[1, 2] * 3`

b) `[3, 6]`

c) `[1, 2, 1, 2, 1, 2]`

d) `[1, 2], [1, 2], [1, 2]`

10. Which method or operator can be used to concatenate lists?

a) `*`

b) `+`

c) `%`

d) `Concat`

11. This file object method returns a list containing the file's contents.

a) `to_list`

b) `getlist`

c) ~~`readline`~~

d) `readlines`

PROGRAMS

12. Write a program that generates 1,000 random integers between 0 and 9 and displays the count for each number.

Hint: Use a list of ten integers named `counts`. The `counts` list will store the counts for the number of 0s, 1s, ..., 9s.

13. Design a program that lets the user enter the total rainfall for each of 12 months into a list. The program should calculate and display the total rainfall for the year and *the average monthly rainfall*.

14. *(Count occurrence of numbers)* Write a program that reads some integers between 1 and 100 and counts the occurrences of each. Here is a sample run of the program:

```
Enter integers between 1 and 100: 2 5 6 5 4 3 23 43 2 ↵ Enter
2 occurs 2 times
3 occurs 1 time
4 occurs 1 time
5 occurs 2 times
6 occurs 1 time
23 occurs 1 time
43 occurs 1 time
```

15. *(Analyze scores)* Write a program that inputs a desired number of scores (from user as integer number) and determines how many scores are above or equal to the average and how many scores are below the average.

Hint: You can use a list to store the scores then process the list.

16. *(Count number of single digits)* Write a program that generates 1,000 random integers between 0 and 9 and displays the count for each number.

Hint: Use a list of ten integers, say counts, to store the counts for the number of 0s, 1s, ..., 9s.

17. *(Statistics: compute deviation)* You will write a program that computes the standard deviation of 10 numbers. The formula to compute the standard deviation of *n* numbers:

$$mean = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \ldots + x_n}{n} \qquad deviation = \sqrt{\frac{\sum_{i=1}^{n} (x_i - mean)^2}{n - 1}}$$

To compute the standard deviation with this formula, you have to store the individual numbers using a list, so that they can be used after the mean is obtained.

An example run of the program is shown below:

```
Enter numbers: 1.9 2.5 3.7 2 1 6 3 4 5 2 ↵ Enter
The mean is 3.11
The standard deviation is 1.55738
```

18. Write a program that reads the content of the `numbers_random.txt` file which contains an unknown number of random numbers. At the end, your program will display how many numbers are read from the file, total and the average of the numbers and how many of the numbers are above the average.

Hint: You must use a list to process the numbers and `readlines` file method can be used to read the data from the file.