Last/Family Name (please, \texttt{print}): \rule{17cm}{0.2mm}

First Name (please, \texttt{print}): \rule{17cm}{0.2mm}

Student Number: \rule{17cm}{0.2mm}

Signature: \rule{17cm}{0.2mm}

Instructor: Anne Lavergne

This examination has 6 pages. Let us verify that we have a complete paper.

\begin{center}
\begin{tabular}{|c|c|}
\hline
Q # & Marks \\
\hline
1 & / 14 \\
\hline
2 & / 12 \\
\hline
3 & / 19 \\
\hline
Total & / 45 \\
\hline
\end{tabular}
\end{center}

\textbf{Instructions:}

\begin{itemize}
\item One cheat sheet allowed. This cheat sheet must …
  \begin{itemize}
  \item be 8 1/2 by 11 inches (letter size) and one-sided,
  \item be handwritten,
  \item have our name and student number on its top right corner, and
  \item be handed in with this midterm examination.
  \end{itemize}
\item No books, calculators, computers, cell phones, or other materials may be used.
\item Read each question carefully before answering it.
\item We must use \texttt{Python 3}.
\item Let us always \texttt{comment our code} and use the \textbf{Good Programming Style} (GPS) discussed in class and used in our labs and assignments.
\item We can assume that all Python code fragments given in this exam are syntactically correct, unless stated.
\item The marks for each question are given in [ ]. Let us use this to manage our time:
  \begin{itemize}
  \item 1 mark correspond to 1 minute of work.
  \end{itemize}
\end{itemize}

Good luck!
1. [14 marks in total– See individual question for marks - No part marks] Write the output each of the following Python code fragments produce (on the computer monitor screen) in the box to its right. If we encounter an error, **we do not have to fix it.** All we have to do is write “error” in the box to the right.

<table>
<thead>
<tr>
<th>a) [2 marks] Python Code Fragment:</th>
</tr>
</thead>
<tbody>
<tr>
<td>numbers = list(range(0,9))</td>
</tr>
<tr>
<td>aSlice = numbers[0::2]</td>
</tr>
<tr>
<td>print(aSlice)</td>
</tr>
<tr>
<td>[0, 2, 4, 6, 8]</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>b) [4 marks] Python Code Fragment:</td>
</tr>
<tr>
<td>Name = &quot;Xiao&quot;</td>
</tr>
<tr>
<td>name = &quot;Xiao&quot;</td>
</tr>
<tr>
<td>taxable = 45900.5124</td>
</tr>
<tr>
<td>rate = 15</td>
</tr>
<tr>
<td>age = 26</td>
</tr>
<tr>
<td>record = &quot;Name: %s\n\n\n\ntaxable: $%0.2f\n\nrate: %d%%&quot; % (name, age)</td>
</tr>
<tr>
<td>record += &quot;\n\n\n\ntaxable: $%0.2f&quot; % taxable</td>
</tr>
<tr>
<td>record += &quot;\n\n\n\nrate: %d%%&quot; % rate</td>
</tr>
<tr>
<td>print(&quot;{}&quot;.format(record))</td>
</tr>
<tr>
<td>The first statement was meant to be:</td>
</tr>
<tr>
<td>name = &quot;Xiao&quot;</td>
</tr>
<tr>
<td>which would have produced:</td>
</tr>
<tr>
<td>Name: Xiao</td>
</tr>
<tr>
<td>age: 26</td>
</tr>
<tr>
<td>taxable: $45900.51</td>
</tr>
<tr>
<td>rate: 15%</td>
</tr>
<tr>
<td>but Microsoft Word capitalized the variable, so the first statement became:</td>
</tr>
<tr>
<td>Name = &quot;Xiao&quot;</td>
</tr>
<tr>
<td>which produced an error.</td>
</tr>
<tr>
<td>So, both answers are accepted.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>c) [2 marks] Python Code Fragment:</td>
</tr>
<tr>
<td>grades = [1, 2, 3, 4]</td>
</tr>
<tr>
<td>grades.remove(2)</td>
</tr>
<tr>
<td>print(grades)</td>
</tr>
<tr>
<td>[1, 3, 4]</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>d) [2 marks] Python Code Fragment:</td>
</tr>
<tr>
<td>print(str(16 - 6 // 2 * 3) * 3)</td>
</tr>
<tr>
<td>777</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
e) [4 marks] Python Code Fragment:

```python
symbol = '&'  
stop = 6  
aVar = list()  
for number in range(1, stop+1):
    aChar = str(number)  
    if aChar in "13579":
        aVar.append(symbol)  
    else:
        aVar.append(aChar)  
print("%s" % aVar)"
```

`['&', '2', '&', '4', '&', '6']`
2. [12 marks] Consider the following Python code fragment:

```python
grade = int(input("Please, enter a grade between 0 and 100 : "))
if grade < 0:
    print("Negative grade.")
elif grade > 100:
    print("Grade > 100.")
elif 0 <= grade <= 49 :
    print("F")
elif 50 <= grade <= 69 :
    print("C")
elif 70 <= grade <= 89 :
    print("B")
else :
    print("A")
```

In the table below, create the minimum number of test cases we would need to completely test this Python code fragment.

<table>
<thead>
<tr>
<th>Test Case #</th>
<th>Test Data</th>
<th>Expected Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-6</td>
<td>Negative grade.</td>
</tr>
<tr>
<td>2</td>
<td>106</td>
<td>Grade &gt; 100.</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>57</td>
<td>C</td>
</tr>
<tr>
<td>5</td>
<td>72</td>
<td>B</td>
</tr>
<tr>
<td>6</td>
<td>94</td>
<td>A</td>
</tr>
</tbody>
</table>
3. [19 marks] **Problem Statement:** Write a **complete Python program** that, given a list of string elements, figures out how many of these string elements contain words (i.e., contain only letters) and how many of these string elements contain numbers (i.e., contain only digits).

To exemplify the behaviour and the output of this Python program, here are some **sample runs:**

1. If `aList = ['apple', 'banana', 'pear']`, then our Python program produces:
   
   ```
   ['apple', 'banana', 'pear'] contains 3 word(s) and 0 number(s)
   ```

2. If `aList = ['me', '2', 'you', '3', 'her', 'them', '78']`, then our Python program produces:
   
   ```
   ['me', '2', 'you', '3', 'her', 'them', '78'] contains 4 word(s) and 3 number(s)
   ```

3. If `aList = [ ]`, then our Python program produces:
   
   ```
   [ ] contains 0 word(s) and 0 number(s)
   ```

4. If `aList = ['23', '24', '25', '26']`, then our Python program produces:
   
   ```
   ['23', '24', '25', '26'] contains 0 word(s) and 4 number(s)
   ```

5. If `aList = ['sun']`, then our Python program produces:
   
   ```
   ['sun'] contains 1 word(s) and 0 number(s)
   ```

6. If `aList = ['9']`, then our Python program produces:
   
   ```
   ['9'] contains 0 word(s) and 1 number(s)
   ```

We do not have to create function(s) in our Python program.

We must not ask the user to enter a list. Instead, we can simply assign a list to our list variable in our program as follows:

```python
aList = ['apple', 'banana', 'pear']
```

Words containing a mixture of letters, digits and/or symbols such as '2s*u$n' or '5 cats' cannot/will not be used in our Python program.

If we find it useful to start by designing an algorithm, let’s feel free to do so.

Let’s use the next page to answer this question.
Possible Solution:

# MT_1174_Question_3.py
#
# This Python program, given a list of string elements, figures out
# how many of these string elements contain words
# (i.e., contain only letters) and how many of these string elements
# contain numbers (i.e., contain only digits).
#
# <my name>
# <today’s date>

# You only needed to use 1 test case!

# Test Case 1

# Set variables
aList = ["apple", "banana", "pear"]
countDigit = 0
countAlpha = 0

# For each element from the list
for element in aList:
    # Figure out whether it is a number or a word
    if element.isnumeric( ) :
        # Running count of numbers
        countDigit += 1
    else :
        # Running count of words
        countAlpha += 1

# Print the result in required format
print("%s contains %d word(s) and %d number(s)" %(aList, countAlpha, countDigit ))

# Test Case 2

# Set variables
bList = ["me", "2", "you", "3", "her", "them", "78"]
countDigit = 0
countAlpha = 0

# For each element from the list
for element in bList:
    # Figure out whether it is a number or a word
    if element.isnumeric( ) :
        # Running count of numbers
        countDigit += 1
    else :
        # Running count of words
        countAlpha += 1
# Running count of words
    countAlpha += 1

# Print the result in required format
print("%s contains %d word(s) and %d number(s)" %(bList, countAlpha, countDigit ))

#------------------

# Test Case 3

# Set variables
cList = []
countDigit = 0
countAlpha = 0

# For each element from the list
for element in cList:
    # Figure out whether it is a number or a word
    if element.isnumeric( ) :
        # Running count of numbers
        countDigit += 1
    else :
        # Running count of words
        countAlpha += 1

# Print the result in required format
print("%s contains %d word(s) and %d number(s)" %(cList, countAlpha, countDigit ))

#------------------

# Test Case 4

# Set variables
dList = ["blue", "4", "white", "black"]
countDigit = 0
countAlpha = 0

# For each element from the list
for element in dList:
    # Figure out whether it is a number or a word
    if element.isnumeric( ) :
        # Running count of numbers
        countDigit += 1
    else :
        # Running count of words
        countAlpha += 1

# Print the result in required format
print("%s contains %d word(s) and %d number(s)" %(dList, countAlpha, countDigit ))

#------------------
# Test Case 5

# Set variables
eList = ['9']
countDigit = 0
countAlpha = 0

# For each element from the list
for element in eList:
    # Figure out whether it is a number or a word
    if element.isnumeric() :
        # Running count of numbers
        countDigit += 1
    else :
        # Running count of words
        countAlpha += 1

# Print the result in required format
print('%s contains %d word(s) and %d number(s)'
      % (eList, countAlpha, countDigit))

#------------------

# Test Case 6

# Set variables
fList = ['sun']
countDigit = 0
countAlpha = 0

# For each element from the list
for element in fList:
    # Figure out whether it is a number or a word
    if element.isnumeric() :
        # Running count of numbers
        countDigit += 1
    else :
        # Running count of words
        countAlpha += 1

# Print the result in required format
print('%s contains %d word(s) and %d number(s)'
      % (fList, countAlpha, countDigit))