This practice exam is due the day of the midterm 2 exam. The solutions will be posted the day before the exam but we encourage you to look at the solutions only after you have tried to solve the exam first.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Max</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Given the initial statements:

   s1 = “spam”
   s2 = “ni!”

   Show a python expression that could construct each of the following results by performing string operations on s1 and s2.

   (a) “NI!”
       
       s2.upper()

   (b) “ni!spamni!”
       
       s2+s1+s2

   (c) “Spam Ni! Spam Ni! Spam Ni!”
       
       s3 = s1[0].upper()+s1[1:]
       s4 = s2[0].upper()+s2[1:]
       
       (s3+””+s4)*3

   (d) “spam”
       
       s1

   (e) [“sp”, “m”]
       
       [s1[0:2],s1[-1]]

   (f) “spm”
       
       s1[0:2]+s1[-1]

2. 
The following is a (silly) decision structure:

```python
a,b,c = eval(input('Enter three numbers: '))
if a > b:
    if b > c:
        print("Spam Please!")
    else:
        print("It's a late parrot!")
elif b > c:
    print("Cheese Shoppe")
    if a >= c:
        print("Cheddar")
elif a < b:
    print("Gouda")
elif c == b:
    print("Swiss")
else
    print("Trees")
    if a == b:
        print("Chestnut")
    else:
        print("Larch")
print("Done")
```

Show the output that would result from each of the following possible inputs:

(a) 3,4,5

```
Trees
Larch
Done
```

(b) 3,3,3

```
Trees
Chestnut
Done
```

(c) 5,4,3

```
Spam Please!
Done
```

(d) 3,5,2

```
Cheese Shoppe
Cheddar
Done
```
3. Write a while loop fragment that calculates the following values:

(a) Sum of the first $n$ counting numbers: $1 + 2 + 3 + 4 + \ldots + n$

```
sum = 0
i = 0
while i < n:
    sum = sum + i
    i = i + 1
```

(b) Sum of the first $n$ odd numbers: $1 + 3 + 5 + \ldots + 2n+1$

```
sum = 0
i = 0
while i < n:
    sum = sum + i* 2 + 1
    i = i + 1
```
4. Given the initial statements

\[ s1 = [2,1,4,3] \]
\[ s2 = ['c','a','b'] \]

show the result of evaluating each of the following sequence expressions:

(a) \( s1 + s2 \)

\[ [2,1,4,3,'c','a','b'] \]

(b) \( 3 \ast s1 + 2 \ast s2 \)

\[ [2,1,4,3,2,1,4,3,2,1,3,4,'c','a','b','c','a','b'] \]

(c) \( s1[1] \)

\[ 1 \]

(d) \( s1[1:3] \)

\[ [1,4] \]

(e) \( s1 + s2[-1] \)

Error ( int + string)

(f) \( s1.sort() \)

\[ [1,2,3,4] \]
5. Answer the following short questions about classes, objects, and methods:

a) What Python reserved word starts a class definition?
   class

b) In a method in a class, what is always the name of the first parameter?
   self

c) Within a method definition the instance variable “x” could be accessed via which expression?
   self.x

d) The term applied to hiding details inside class definitions is called?
   Encapsulation
6. A certain CS professor gives 100 point exams that are graded on the scale 90-100:A, 80-89:B, 70-79:C, 60-69:D, <60:F. Write a program that accepts an exam score as input and prints out the corresponding grade.

Example input/output:

Exam Score: 86
Grade: B

```python
def main():
    n = eval(input("Exam Score:"))
    g = ""
    if n >= 90:
        g = "A"
    elif n >= 80:
        g = "B"
    elif n >= 70:
        g = "C"
    elif n >= 60:
        g = "D"
    else:
        g = "F"
    print("Grade:" + g)
main()
```
7. Write a program that reads a sentence entered by the user, and then it prints the number of words, and the average word length in the sentence.

Example Input/Output:

Sentence: Purdue University is in West Lafayette Indiana
Number of Words: 7
Average Length: 5.71

```python
# words.py

def main():
    s = input("Sentence:"))
    words = s.split(" ")

    print("words=",words)

    sum = 0
    for w in words:
        sum = sum + len(w)

    print("Number of Words:",len(words))
    print("Average Length:", sum/len(words))

main()
```
8. A positive number $n > 2$ is prime if no number between 2 and $n-1$ evenly divides $n$. Write a program that accepts a value of $n$ as input and determines if the value is a prime. If $n$ is not prime, your program should quit as soon as it finds a value that evenly divides $n$.

Example input/output:

n? 17
17 is prime

n? 18
18 is not prime

```python
def main():
    n = eval(input("n?"))
    p = True
    for i in range(2,n-1):
        if n % i == 0:
            p = False
            break
    if p:
        print(n, "is prime")
    else:
        print(n, "is not prime")

main()
```
9. Write a class to represent a cube. Your class should implement the following methods:

- `__init__(self, side)`: Creates a cube with a given “side” length.
- `getSide(self)`: Returns the “side” length of a cube.
- `getSurfaceArea(self)`: Returns the surface of a cube using the formula
  \[ \text{surface} = 6 \times \text{side}^2 \]
- `getVolume(self)`: Returns the volume of a cube using the formula
  \[ \text{volume} = \text{side} \times \text{side} \times \text{side} \]

```python
class Cube:
    def __init__(self, side):  # Creates a cube with a given “side” length.
        self.side = side

    def getSide(self):  # Returns the “side” length of a cube.
        return self.side

    def getSurfaceArea(self):  # Returns the surface of a cube using the formula
        area = 6 * self.side * self.side
        return area

    def getVolume(self):  # Returns the volume of a cube using the formula
        volume = self.side * self.side * self.side
        return volume

def main():
    cube = Cube(10)
    area = cube.getSurfaceArea()
    volume = cube.getVolume()
    print("side=", cube.getSide(), "area=", area, "volume=", volume)

main()
```
10. A palindrome is a sentence that you can read the same from left to right and from right to left without regard of spaces or punctuation. For example, the following sentences are palindromes: “Never odd or even”, “Madam, I’m Adam”. Wrote a program that tells you if a sentence is a palindrome or not.

Example input/output:

Sentence? Never odd or even
It is a palindrome.
Do you want to continue? (yes/no) yes
Sentence? Hello world
It is not a palindrome.

#palindrome.py

def main():
    while True:
        s = input("Sentence?")

        # Remove punctuation and spaces
        # Iterate over all characters ch in s.
        # Add ch to s2 only if ch is a letter.
        #
        s2 = 
        for ch in s:
            if ch.isalpha():
                s2 = s2 + ch
        s = s2

        # Convert s to lower case
        s = s.lower()

        print(s)

        # Obtain reverse
        r = 
        for ch in s:
            r = ch + r

        # Compare reverse with s
        if s == r:
            print("It is a palindrome")
else:
    print("It is not a palindrome.")

yesno=input("Do you want to continue? (yes/no)")
if yesno=="no":
    break

main()