

1. What will the following code print:

```
days = "mondaytuesdaywednesdayetc"
print(days)
x = days[6:9]
print(x)
one, two, three, four = days.split("day")
print(one, two, three, four)
t = two.capitalize()
print(t)
n = ",".join(days.split("day"))
print(n)
```

2. Trace the **program** below, indicating the state of the memory and the display on screen as it executes:

```
def something(x):
    for i in x :
        print("CMP {0}".format(i))

def myfunc(x, y):
    return x * x + y * y

def intro():
    print("This is an example of my code tracing skills:")

def practice():
    intro()
    list = ["230", "326", "338"]
    something(list)
    z = myfunc(2, 3)
    print("I think the final value is {0}".format(z))

practice()
```

3. Write a **program** that reads in a text file as indicated by the user, determines the longest word in the file, and prints that word out to the screen. Assume the file has one word per line. In the event that more than one word qualifies as the longest, print out the first such word only.
4. Write a **function** that returns the number of even integers in a list of integers.
5. Write a **program** that asks for the name of a file. Assume the file contains numbers, one per line. The program should print out how many numbers in the file are above average.
6. A certain CS professor gives 5-point quizzes that are graded on the scale 5-A, 4-B, 3-C, 2-D, 1-F, 0-F. Write a **program** that accepts an integer quiz score as an input and prints out the corresponding grade.
7. Write a **function** that converts a list of ASCII codes into a list of the corresponding characters.
8. Write a **program** that prompts the user for the name of three files—let's call them one, two and three. Files one and two contain numbers, separated by spaces. Both files contain the same number of

numbers. Your program reads a number from each of files one and two adds the numbers, and writes the result to file three. Then the next numbers, the next, and so on, until all numbers have been processed.

Useful String Methods: (from p 140 of textbook)

Function	Meaning
<code>s.capitalize()</code>	Copy of <code>s</code> with only the first character capitalized.
<code>s.center(width)</code>	Copy of <code>s</code> is centered in a field of given width.
<code>s.count(sub)</code>	Count the number of occurrences of <code>sub</code> in <code>s</code> .
<code>s.find(sub)</code>	Find the first position where <code>sub</code> occurs in <code>s</code> .
<code>s.join(list)</code>	Concatenate <code>list</code> into a string using <code>s</code> as a separator.
<code>s.ljust(width)</code>	Like <code>center</code> , but <code>s</code> is left-justified.
<code>s.lower()</code>	Copy of <code>s</code> with all characters converted to lowercase.
<code>s.lstrip()</code>	Copy of <code>s</code> with leading whitespace removed.
<code>s.replace(oldsub,newsub)</code>	Replace all occurrences of <code>oldsub</code> in <code>s</code> with <code>newsub</code> .
<code>s.rfind(sub)</code>	Like <code>find</code> , but returns rightmost position.
<code>s.rjust(sub)</code>	Like <code>center</code> , but <code>s</code> is right-justified.
<code>s.rstrip()</code>	Copy of <code>s</code> with trailing whitespace removed.
<code>s.split()</code>	Split <code>s</code> into a list of substrings.
<code>s.title()</code>	Copy of <code>s</code> with first character of each word capitalized.
<code>s.upper()</code>	Copy of <code>s</code> with all characters converted to uppercase.

1. What will the following code print:

```
all = "Uno:Dos"
l = all.split(":")
one = l[0]
two = l[1]
print(one)
print(two)
both = one.upper() + ", " + two.lower()
print(both)
part = both[5:8]
print(part)
```

2. Trace the **program** below, indicating the state of the memory and the display on screen as it executes:

```
def doCalculation(list):
    x = list[0]
    y = list[0]
    for i in range(len(list)):
        if list[i] > x :
            x = list[i]
        if list[i] < y :
            y = list[i]
    return x - y

def Prog2():
    list = [18, 12, 45, 17, 94, 34, 7, 56]
    x = doCalculation(list)
    print("The property in question has a value of {}".format(x))
```

Prog2()

3. Write a **program** that reads in a text file as indicated by the user, determines the number of words in the file, and prints that number out to the screen.
4. Write a **function** that returns the number of vowels in a string. Vowels are 'a', 'e', 'i', 'o', 'u'. You may assume all letters in the string are lowercase.
5. Write a **program** that inputs the names of two files. Assume both files contain numbers, one per line. The program should print out the name of the file containing the most numbers.
6. Write a **program** that takes a number between 1 and 7 and prints out the corresponding day of the week. For example, if the input is 1, your program should print "Sunday".
7. Write a **function** that converts a list of strings of lower case letters into a list of the corresponding strings with upper case characters.
8. Write a **program** that prompts the user for the name of a file. Your program should then print to the screen if there are more A's than B's in the file.

Useful String Methods: (from p 140 of textbook)

Function	Meaning
<code>s.capitalize()</code>	Copy of <code>s</code> with only the first character capitalized.
<code>s.center(width)</code>	Copy of <code>s</code> is centered in a field of given width.
<code>s.count(sub)</code>	Count the number of occurrences of <code>sub</code> in <code>s</code> .
<code>s.find(sub)</code>	Find the first position where <code>sub</code> occurs in <code>s</code> .
<code>s.join(list)</code>	Concatenate <code>list</code> into a string using <code>s</code> as a separator.
<code>s.ljust(width)</code>	Like <code>center</code> , but <code>s</code> is left-justified.
<code>s.lower()</code>	Copy of <code>s</code> with all characters converted to lowercase.
<code>s.lstrip()</code>	Copy of <code>s</code> with leading whitespace removed.
<code>s.replace(oldsub,newsub)</code>	Replace all occurrences of <code>oldsub</code> in <code>s</code> with <code>newsub</code> .
<code>s.rfind(sub)</code>	Like <code>find</code> , but returns rightmost position.
<code>s.rjust(sub)</code>	Like <code>center</code> , but <code>s</code> is right-justified.
<code>s.rstrip()</code>	Copy of <code>s</code> with trailing whitespace removed.
<code>s.split()</code>	Split <code>s</code> into a list of substrings.
<code>s.title()</code>	Copy of <code>s</code> with first character of each word capitalized.
<code>s.upper()</code>	Copy of <code>s</code> with all characters converted to uppercase.

1. What will the following code print:

```
groceries = "milk,eggs,butter,bread,fruit"
print(groceries)
list = groceries.upper().split(",")
print(list)
odd1, odd2, odd3 = groceries.split("i")
print(odd1, odd2, odd3)
newList = " ; ".join(list)
print(newList)
c = odd1.center(6)
print(c)
```

2. Trace the **program** below, indicating the state of the memory and the display on screen as it executes:

```
def welcome():
    print("Welcome to Bank 230")

def calculate(amount, rate)
    return amount*(1+rate)

def addInterest(balances, rate):
    for i in range(len(balances)):
        balances[i] = calculate(balances[i],rate)

def test():
    amounts = [100,1000,10000]
    rate = 0.10
    addInterest(amounts, rate)
    print(amounts)

test()
```

3. Write a **program** that reads in a text file as indicated by the user, determines the number of lines with 10 or fewer characters, and prints that number out to the screen.
4. Write a **function** that returns the number of spaces in a list of strings.
5. Write a **program** that inputs the name of a file. Assume the file contains numbers, one on each line. The program prints out the word “even” or “odd” depending on which are more numerous in the file.
6. Write a **program** that takes a number between 1 and 12 and prints out the corresponding month of the year. For example, if the input is 1, your program should print "January".
7. Write a **function** that converts a list of numbers into a list of the corresponding squares of those numbers. For example, if the input list is [2,3,1,4], then the output list is [4,9,1,16].
8. Write a **program** that prompts the user for the name of a file. Your program should then print out how many numbers are in the file and also print the maximum number in the file. The file is formatted with one number per line.

Useful String Methods: (from p 140 of textbook)

Function	Meaning
<code>s.capitalize()</code>	Copy of <code>s</code> with only the first character capitalized.
<code>s.center(width)</code>	Copy of <code>s</code> is centered in a field of given width.
<code>s.count(sub)</code>	Count the number of occurrences of <code>sub</code> in <code>s</code> .
<code>s.find(sub)</code>	Find the first position where <code>sub</code> occurs in <code>s</code> .
<code>s.join(list)</code>	Concatenate <code>list</code> into a string using <code>s</code> as a separator.
<code>s.ljust(width)</code>	Like <code>center</code> , but <code>s</code> is left-justified.
<code>s.lower()</code>	Copy of <code>s</code> with all characters converted to lowercase.
<code>s.lstrip()</code>	Copy of <code>s</code> with leading whitespace removed.
<code>s.replace(oldsub,newsub)</code>	Replace all occurrences of <code>oldsub</code> in <code>s</code> with <code>newsub</code> .
<code>s.rfind(sub)</code>	Like <code>find</code> , but returns rightmost position.
<code>s.rjust(sub)</code>	Like <code>center</code> , but <code>s</code> is right-justified.
<code>s.rstrip()</code>	Copy of <code>s</code> with trailing whitespace removed.
<code>s.split()</code>	Split <code>s</code> into a list of substrings.
<code>s.title()</code>	Copy of <code>s</code> with first character of each word capitalized.
<code>s.upper()</code>	Copy of <code>s</code> with all characters converted to uppercase.

1. What will the following code print:

```
list = "cmp230;cmp326;cmp334;cmp338"
print(list)
list.replace("cmp", "CMP ")
print(list)
n = list.count("cmp")
print(n)
one, two, three, four = list.split(";")
print(one, two, three, four)
c = two.upper()
print(c)
```

2. Trace the **program** below, indicating the state of the memory and the display on screen as it executes:

```
def welcome(num):
    for i in range(num):
        print("This is the way the world")

def end():
    return("but a whimper.")

def middle(symbol, num):
    s = symbol + " with a "
    for i in range(num):
        s = s + symbol
    print(s)

def Problem2():
    welcome(3)
    middle("!", 4)
    s = end()
    print(s)

Problem2()
```

3. Write a **program** that reads in a text file as indicated by the user, determines the number of occurrences of 'a', 'an', and 'the' in the file, and prints that number out to the screen.
4. Write a **function** that returns the number of positive integers in a list of integers.
5. Write a **program** that inputs the name of a file. Assume the file contains numbers separated by spaces. The program prints out the smallest and largest number.
6. Write a **program** that takes a number between 1 and 9 and prints out the corresponding word for that number. For example, if the input is 1, your program should print "One".
7. Write a **function** that converts a list of numbers into a list of the corresponding absolute value of those numbers. For example, if the input list is $\{-2, 3, -1, 4\}$, then the output list is $\{2, 3, 1, 4\}$.
8. Write a **program** that prompts the user for the names of two files. Your program should then print the name of the longer file.

Useful String Methods: (from p 140 of textbook)

Function	Meaning
<code>s.capitalize()</code>	Copy of <code>s</code> with only the first character capitalized.
<code>s.center(width)</code>	Copy of <code>s</code> is centered in a field of given width.
<code>s.count(sub)</code>	Count the number of occurrences of <code>sub</code> in <code>s</code> .
<code>s.find(sub)</code>	Find the first position where <code>sub</code> occurs in <code>s</code> .
<code>s.join(list)</code>	Concatenate <code>list</code> into a string using <code>s</code> as a separator.
<code>s.ljust(width)</code>	Like <code>center</code> , but <code>s</code> is left-justified.
<code>s.lower()</code>	Copy of <code>s</code> with all characters converted to lowercase.
<code>s.lstrip()</code>	Copy of <code>s</code> with leading whitespace removed.
<code>s.replace(oldsub,newsub)</code>	Replace all occurrences of <code>oldsub</code> in <code>s</code> with <code>newsub</code> .
<code>s.rfind(sub)</code>	Like <code>find</code> , but returns rightmost position.
<code>s.rjust(sub)</code>	Like <code>center</code> , but <code>s</code> is right-justified.
<code>s.rstrip()</code>	Copy of <code>s</code> with trailing whitespace removed.
<code>s.split()</code>	Split <code>s</code> into a list of substrings.
<code>s.title()</code>	Copy of <code>s</code> with first character of each word capitalized.
<code>s.upper()</code>	Copy of <code>s</code> with all characters converted to uppercase.

1. What will the following code print:

```
mo = "january february march april may"
print(mo)
t = mo.title()
print(t)
n = mo.count("ary")
print(n)
months = mo.split(" ")
print(months)
c = months[2].upper()
print(c)
```

2. Trace the **program** below, indicating the state of the memory and the display on screen as it executes:

```
def doCalculation(list):
    x = list[0]
    y = list[0]
    for i in range(len(list)):
        if list[i] > x :
            x = list[i]
        if list[i] < y :
            y = list[i]
    return x, y

def Prog2():
    list = [21, 15, 48, 19, 92, 3, 17, 6]
    x,y = doCalculation(list)
    print("The property in question has values of {}".format(x))
    print("and of {}".format(y));
```

```
Prog2()
```

3. Write a **program** that reads in a text file as indicated by the user, determines the number of characters in the file, and prints that number out to the screen.
4. Write a **function** that returns the number of integers less than 10 in a list of integers.
5. Write a **program** that inputs the names of two files and prints out the name of the file whose numbers have the largest sum. Assume that there is one number per line in the files.
6. Write a **program** that takes a number between 1 and 9 and prints out the corresponding ordinal word for that number. For example, if the input is 1, your program should print "First".
7. Write a **function** that converts a list of strings of lower case letters into a list of the corresponding strings with upper case characters.
8. Write a **program** that prompts the user for the name of a files. Your program should report how many numbers in the file are larger than average. The file is formatted with one number per line.

Useful String Methods: (from p 140 of textbook)

Function	Meaning
<code>s.capitalize()</code>	Copy of <code>s</code> with only the first character capitalized.
<code>s.center(width)</code>	Copy of <code>s</code> is centered in a field of given width.
<code>s.count(sub)</code>	Count the number of occurrences of <code>sub</code> in <code>s</code> .
<code>s.find(sub)</code>	Find the first position where <code>sub</code> occurs in <code>s</code> .
<code>s.join(list)</code>	Concatenate <code>list</code> into a string using <code>s</code> as a separator.
<code>s.ljust(width)</code>	Like <code>center</code> , but <code>s</code> is left-justified.
<code>s.lower()</code>	Copy of <code>s</code> with all characters converted to lowercase.
<code>s.lstrip()</code>	Copy of <code>s</code> with leading whitespace removed.
<code>s.replace(oldsub,newsub)</code>	Replace all occurrences of <code>oldsub</code> in <code>s</code> with <code>newsub</code> .
<code>s.rfind(sub)</code>	Like <code>find</code> , but returns rightmost position.
<code>s.rjust(sub)</code>	Like <code>center</code> , but <code>s</code> is right-justified.
<code>s.rstrip()</code>	Copy of <code>s</code> with trailing whitespace removed.
<code>s.split()</code>	Split <code>s</code> into a list of substrings.
<code>s.title()</code>	Copy of <code>s</code> with first character of each word capitalized.
<code>s.upper()</code>	Copy of <code>s</code> with all characters converted to uppercase.