

1. What will the following code print:

```
aves = "Madison,Park,Lexington"
s = aves.split(",")
print(s)
a = s[0]
b = s[1].lower()
c = s[2].upper()
print(a,b,c)
print("{0} and {1} Avenues".format(a,b))
last = c[0:3] + " Ave."
print(last)
```

2. Write a **function** that takes as a parameter a list of numbers and returns a list with each number replaced by the corresponding string.
3. What will the following program print:

```
def first():
    print("Exam 2, Version 1")
def count(m):
    return len(m)
def last(n,x):
    print(n[0:x])
def test():
    first()
    m = "I love python"
    c = count(m)
    print(m,c)
    last(m,6)
test()
```

4. Fill in the missing function definitions for this program:

```
def main():
    welcome()           #Prints "Hello, world" to the screen
    x = userInput()     #Asks user for input and returns number entered
    d = calculate(x)    #Returns twice the parameter
    displayResults(x,d) #Prints the two inputs, on separate lines
main()
```

(That is, write the functions `welcome()`, `userInput()`, `calculate()` and `displayResults()`.)

5. What is printed when the function is invoked on the inputs below:

```
def weather(rain,snow,clear):
    if rain < clear and snow < clear:
        print("Sunny skies")
    elif snow < rain:
        print("Snow!")
    else:
        print("Rain?")
```

- (a) `weather(.4, .3, .3)`
- (b) `weather(.1, .1, .8)`
- (c) `weather(.25, .25, .5)`

6. Write a **function** that takes as a parameter the number of credits a student has completed and prints out the student's class standing ('freshman', 'sophomore', 'junior', or 'senior'). Students with less than 30 credits are a freshman. Students with more than 30 but less than 60 credits are sophomores. Students with more than 60 but less than 90 credits are juniors. Students with 90 and more credits are seniors.

7. What will the following code print:

```

nums = [100,90,83,80,55,92,81,98,99,70]
countA = 0
countB = 0
for i in nums:
    if i >= 90:
        countA = countA + 1
    elif i >= 80:
        countB = countB + 1
print(countA, countB)

```

8. Write a **function** that takes as a parameter a list of numbers and returns a number of the items that are negative.

9. Given the following program and input file, what is printed:

<pre> def main():     infile = open("in.txt", "r")     for w in infile.readlines():         n = eval(w)         if n%10 == 0:             print(n)     infile.close() </pre>	<p><b>in.txt</b></p> <p>21</p> <p>30</p> <p>2011</p> <p>230</p> <p>8</p> <p>1959</p> <p>10</p>
--	--

10. Write a **program** that reads in a text file, `infile.txt`, and writes out the contents to another file, `outfile.txt`, in reverse order (that is the last items in the file occur first and the first items occur last in the new file). You may assume that every line of the input file consists of exactly one number.

**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:

```
stops = "GCT,125,Yankee Stadium,University Heights,Marble Hill"
s = stops.split(",")
print(s)
a = s[0].lower()
b = s[1]
c = s[2].upper()
d = s[3]
print("{0} and {1}th Street Stations".format(a,b))
print(c,d)
count = stops.count(" ")
print(count)
```

2. Write a function that takes as a parameter a list of strings and returns a list with each string in upper case.
3. What will the following program print:

```
def first():
    print("Version 2")
def count(m):
    return m.count(" ")
def last(n,x):
    print(n[x:])
def test():
    first()
    first()
    m = "I love python"
    c = count(m)
    print(c)
    last(m,7)
test()
```

4. Fill in the missing function definitions for this program:

```
def main():
    welcome()           #Prints "Greetings!" to the screen
    x = userInput()     #Asks user for input and returns number entered
    d = calculate(x)    #Returns the square of the parameter
    displayResults(x,d) #Prints the two inputs, on separate lines
main()
```

(That is, write the functions `welcome()`, `userInput()`, `calculate()` and `displayResults()`.)

5. What is printed when the function is invoked on the inputs below:

```
def weather(temp,humidity):
    if temp < 32:
        print("Brr...")
    elif temp < 90 and humidity < 90:
```

```

    print("Nice Day")
else:
    print("Too hot")

```

- (a) `weather(50,50)`
- (b) `weather(20,70)`
- (c) `weather(95,50)`

6. Write a **function** that takes as a parameter income and returns a tax rate. If income is less than 8,350, the tax rate is 10%. If income is between 8,351 and 33,950, the tax rate is 15%. If income is between 33,951 and 82,250, the tax rate is 25%. If income is between 82,251 and 171,550, the tax rate is 28%. If income is between 171,551 and 372,950, the tax rate is 33%. If income is higher than 372,951, the tax rate is 35%.

7. What will the following code print:

```

s = "the quick brown fox jumped over the lazy dog"
countE = 0
countZ = 0
for c in s:
    if c == 'e':
        countE = countE + 1
    elif c == 'z':
        countZ = countZ + 1
print(countE, countZ)

```

8. Write a **function** that takes as a parameter a list of numbers and returns the number of items larger than 7.

9. Given the following program and input file, what is printed:

<pre> def main():     infile = open("in.txt", "r")     for w in infile.readlines():         n = eval(w)         if n//100 == 0:             print(n)     infile.close() </pre>	<pre> <b>in.txt</b> 21 30 2011 230 8 1959 10 </pre>
--	---

10. Write a **program** that reads in a text file, `infile.txt` and writes out the even numbers to another file, `outfile.txt`. You may assume that every line of the input file consists of exactly one number.

**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:

```
stops = "4,14,34,42,59,125"
s = stops.split(",")
print(s)
a = s[0]
b = s[1]
print("{0}th and {1}th Street Stations".format(a,b))
print(c,d)
count4 = stops.count("4")
print(count4)
```

2. Write a function that takes as a parameter a list of strings and returns a list containing the first letter of each string in the list.
3. What will the following program print:

```
def first():
    print("Version"+3*"3")
def count(m):
    return m.count("!")
def last(n,x):
    print(n[0:x])
def test():
    first()
    m = "Python loves me!"
    c = count(m)
    print(c)
    last(m,6)
test()
```

4. Fill in the missing function definitions for this program:

```
def main():
    welcome()          #Prints "Greetings!" to the screen
    x = userInput()    #Asks user for input and returns the string entered
    d = calculate(x)   #Returns the the string in upper case letters
    displayResults(x,d) #Prints the two inputs, on separate lines
main()
```

(That is, write the functions `welcome()`, `userInput()`, `calculate()` and `displayResults()`.)

5. What is printed when the function is invoked on the inputs below:

```
def change(price,given):
    if price <= 0 or given <= 0:
        print("Enter positive numbers only!")
    elif price < given:
        print("You have change!")
    elif price == given:
        print("Thank you for exact change!")
    else:
        print("You owe money!")
```

- (a) `change(10,10)`
  - (b) `change(15,20)`
  - (c) `change(15,10)`
6. Write a **function** that takes as a parameter the number of axles on a truck and returns the toll to cross the Triborough (Robert F. Kennedy) Bridge. If the number of axles is 2, the toll is 13. If the number of axles is 3, the toll is 21. If the number of axles is 4, the toll is 27. If the number of axles is 5, the toll is 35. If the number of axles is 6, the toll is 41. If the number of axles is 7, the toll is 50.
7. What will the following code print:

```
s = "012345012345012345"
count0 = 0
count5 = 0
for c in s:
    if c == '0':
        count0 = count0 + 1
    elif c == '5':
        count5 = count5 + 1
print(count0, count5)
```

8. Write a **function** that takes as parameter a list of strings and returns the number of strings of that begin with 'A'.
9. Given the following program and input file, what is printed:

<pre>def main():     infile = open("in.txt", "r")     for w in infile.readlines():         n = eval(w)         if n//1000 == 0:             print(n)     infile.close()</pre>	<p><b>in.txt</b></p> <p>2021</p> <p>300</p> <p>2011</p> <p>230</p> <p>8</p> <p>1959</p>
---	---

10. Write a **program** that reads in a text file, `infile.txt` and writes out the positive numbers to another file, `outfile.txt`. You may assume that every line of the input file consists of exactly one number.

**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(width)</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:

```
random = "XHxeXlXLXo8XWxoXRXLXD"
list = random.split("X")
count = len(list)
print(count)
s = "".join(list)
print(s)
list2 = s.split("8")
print(list2[0],list2[1])
```

2. Write a function that takes as a parameter a list of strings and returns a list containing the first letter of each string in the list.
3. What will the following program print:

```
def first():
    print("4"*4, "Version")
def count(m):
    return m.count("o")
def last(n,x):
    print(n[4:x])
def test():
    first()
    m = "Python loves me!"
    c = count(m)
    print(c)
    last(m,8)
test()
```

4. Fill in the missing function definitions for this program:

```
def main():
    welcome()          #Prints "Hello!!!" to the screen
    x,y = userInput() #Asks user for 2 input and returns the numbers entered
    d = calculate(x,y) #Returns the the difference of the number
    displayResults(d) #Prints the result
main()
```

(That is, write the functions `welcome()`, `userInput()`, `calculate()` and `displayResults()`.)

5. What is printed when the function is invoked on the inputs below:

```
def refund(price,paid):
    if price <= 0 or paid <= 0:
        print("Enter positive numbers only!")
    elif price < paid:
        print("You have change!")
    elif price == paid:
        print("Thank you for exact change!")
    else:
        print("You owe money")
```

- (a) `refund(100,100)`
  - (b) `refund(195,200)`
  - (c) `refund(195,190)`
6. Write a **function** that takes as a parameter the speed and returns the corresponding fine. If the speed is less than or equal to 55, there is no fine (return 0). If the speed is greater than 55 but less than or equal to 65, the fine is 50. If the speed is greater than 65 but less than or equal to 75, the fine is 150. If the speed is greater than 75 but less than or equal to 85, the fine is 200. If the speed is greater than 85, then the fine is 300.
7. What will the following code print:

```

nums = [30,35,100,89,39,25,75]
count = 0
for i in range(len(nums) - 1):
    if nums[i] < nums[i+1]:
        count = count+1
        print(i)
print(count)

```

8. Write a **function** that takes as a parameter a list of strings and returns the number of strings of that begin with 'A'.
9. Given the following program and input file, what is printed:

<pre> def main():     infile = open("in.txt", "r")     for w in infile.readlines():         n = eval(w)         if n%100 == 0:             print(n)     infile.close() </pre>	<p style="text-align: right;"><b>in.txt</b></p> <p>200</p> <p>309</p> <p>2011</p> <p>2300</p> <p>8</p> <p>1959</p> <p>1100</p>
---	--

10. Write a **program** that reads in a text file, `infile.txt`, and writes out the even numbers to another file, `outfile.txt`. You may assume that every line of the input file consists of exactly one number.

**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.