

NAME:

EMAIL:

SIGNATURE:

CIRCLE COURSE SECTION: TTh 11-1 MW 1-3 TTh 4-6 MW 6-8
MW 4-6 MW 11-1 MW 9-11

Lehman College, CUNY

CMP 230 Exam 1, Version 1, Spring 2013

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1. What will the following code print:

```
presidents = "RichardXGeraldXJamesXRonaldXGeorgeXWilliamXGeorgeXBarack"  
print(presidents[12], presidents[9], presidents[4], presidents[26])  
num = presidents.count("X") + 1  
names = presidents.split("X")  
print("The last", num, "presidents are", names)  
message = names[-1].upper()  
print(message, "!!!")
```

Output:

2. Write a **function** that takes an input parameter a string of words separated by spaces returns a list of the words with each one capitalized.

3. What will the following code print:

(a)

```
s = "abc def ghi jkl"  
for i in [2,4,6,9]:  
    s = s[:i] + s[i+1] + s[i+2:]  
    print(s)
```

Output:

(b)

```
s = "fziizyt"  
for i in s:  
    n = ord('a')+(ord(i)-ord('a')+5)%26  
    print(i,chr(n))
```

Output:

4. What will the following program print:

```
def second():
    print("They're good shoes")
def first():
    print("These are my new shoes")
    second()
def repeat(qual):
    print("They won't make you",qual,"like me")
    return(1)
def end():
    print("They'll only make you have shoes like me")
def main4_1():
    first()
    count = repeat("rich") + repeat("rebound") + repeat("handsome")
    end()
    print("Count = ", count)
main4_1()
```

Output:

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

```
def main():
    welcome()           #Prints "Welcome" to the screen
    n,d = userInput()  #Asks user for 2 inputs and returns numbers entered
    r = calculate(n,d)  #Returns the remainder when n is divided by d
    displayResults(r)  #Prints the result to the screen
main()
```

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

6. What is returned when the function is invoked on the inputs below:

```
def mystery(s1):  
    n = s1.find(" ")  
    result = -1  
    for i in range(n):  
        result = result + i  
    return(result)
```

(a) `mystery("what does this do?")`

Return:

(b) `mystery("mystery, mystery?")`

Return:

(c) `mystery("I know, I know, I know!")`

Return:

7. What will the following code print:

```
s = "Little Jack Horner sat in the corner."  
m = ""  
M = ""  
for i in range(0,len(s)-1,2):  
    m = m + s[i]  
    M = M + s[i+1]  
print("m = ",m)  
print("M = ",M)
```

Output:

8. Write a function that takes as a parameter a list of strings and returns a list containing the last letter of each of the strings. That is, if the input parameter is ["This", "is", "an", "Example"], your function should return ["s", "s", "n", "e"].

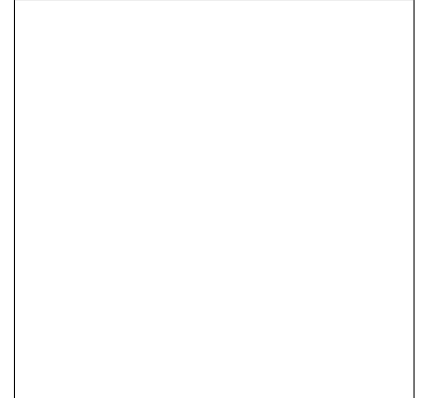
9. Given the following program and input file, what is printed:

```
def filter(words):  
    print(words[2])  
  
def main():  
    infile=open("lincoln.txt","r")  
    lines=infile.readlines()  
    for line in lines:  
        words=line.split(" ")  
        filter(words)  
main()
```

lincoln.txt

```
You can fool all  
the people some  
of the time and  
some of the people  
all the time but  
you cannot fool  
all the people  
all the time
```

Output:



10. Write a **program** that reads in a text file, `infile.txt`, and writes out the contents to another file, `outfile.txt`, with the lines in reverse order.

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.

Useful Unicode Ordinal Numbers

letter	Unicode
A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72
I	73
J	74
K	75
L	76
M	77
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

letter	Unicode
a	97
b	98
c	99
d	100
e	101
f	102
g	103
h	104
i	105
j	106
k	107
l	108
m	109
n	110
o	111
p	112
q	113
r	114
s	115
t	116
u	117
v	118
w	119
x	120
y	121
z	122

NAME:

EMAIL:

SIGNATURE:

CIRCLE COURSE SECTION: TTh 11-1 MW 1-3 TTh 4-6 MW 6-8
MW 4-6 MW 11-1 MW 9-11

Lehman College, CUNY

CMP 230 Exam 1, Version 2, Spring 2013

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1. What will the following code print:

```
presidents = "NixonXFordXCarterXReaganXBushXClintonXBushXObama"  
print(presidents[15], presidents[31], presidents[19], presidents[11], presidents[14])  
num = presidents.count("X") + 1  
names = presidents.split("X")  
print("The last", num, "presidents are", names)  
message = names[-1].upper()  
print(message, "!!!")
```

Output:

2. Write a **function** that takes a input parameter a string of words separated by commas and returns the list of the words in upper case.

3. What will the following code print:

(a)

```
s = "abc def ghi jkl"  
for i in [1,5,7,10]:  
    s = s[:i] + s[i+1] + s[i+2:]  
    print(s)
```

Output:

(b)

```
s = "hejykhj"  
for i in s:  
    n = ord('a')+(ord(i)-ord('a')+4)%26  
    print(i,chr(n))
```

Output:

4. What will the following program print:

```
def repeat(royal):
    print("The",royal,"of hearts", end=" ")
    return(1)
def tarts():
    return("tarts ")
def verses():
    count = repeat("queen")
    print("she made some "+ tarts() + "all on a summer's day")
    count = count + repeat("knave")
    print("he stole the "+ tarts() + "and took them clean away")
    count = count + repeat("king")
    print("called for the "+ tarts() + "and beat the Knave full sore")
    count = count + repeat("knave")
    print("brought back the "+ tarts() + "and vowed he'd steal no more")
    return count
def main4_2():
    print("Count = ", verses())
main4_2()
```

Output:



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

```
def main():
    welcome()           #Prints "My program" to the screen
    f,g = userInput()  #Asks user for 2 inputs and returns numbers entered
    p = calculate(f,g) #Returns the product of the parameters
    displayResults(p)  #Prints the result to the screen
main()
```

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

6. What is returned when the function is invoked on the inputs below:

```
def mystery(s1):  
    n = s1.rfind(" ")  
    result = 1  
    for i in range(n):  
        result = result * i  
    return(result)
```

(a) `mystery("what does this do?")`

Return:

(b) `mystery("mystery, mystery?")`

Return:

(c) `mystery("I know, I know, I know!")`

Return:

7. What will the following code print:

```
s = "So the poor little doggie had none."  
m = ""  
M = ""  
for i in range(0,len(s)-1,2):  
    m = m + s[i]  
    M = M + s[i+1]  
print("m = ",m)  
print("M = ",M)
```

Output:

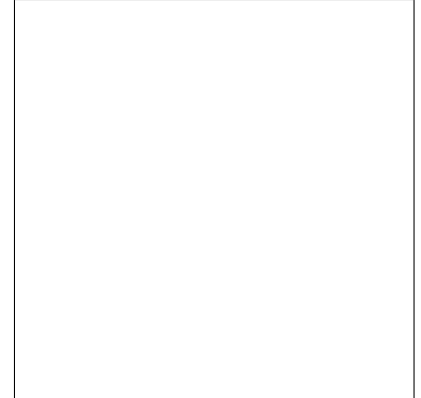
8. Write a function that takes as a parameter a list of strings and returns a list containing the first letter, in upper case, of each of the strings. That is, if the input parameter is ["This", "is", "an", "Example"], your function should return ["T", "I", "A", "E"].

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

```
def filter(words):  
    print(words[1])  
  
def main():  
    infile=open("kennedy.txt","r")  
    lines=infile.readlines()  
    for line in lines:  
        words=line.split(" ")  
        filter(words)  
main()
```

```
kennedy.txt  
My fellow Americans  
ask not what  
your country can  
do for you ask  
what you can do  
for your country.
```

Output:



10. Write a **program** that reads in a text file, `infile.txt`, and writes out the contents to another file, `outfile.txt`, with every instance of the word “and” replaced by the word “the”.

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.

Useful Unicode Ordinal Numbers

letter	Unicode
A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72
I	73
J	74
K	75
L	76
M	77
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

letter	Unicode
a	97
b	98
c	99
d	100
e	101
f	102
g	103
h	104
i	105
j	106
k	107
l	108
m	109
n	110
o	111
p	112
q	113
r	114
s	115
t	116
u	117
v	118
w	119
x	120
y	121
z	122

NAME:

EMAIL:

SIGNATURE:

CIRCLE COURSE SECTION: TTh 11-1 MW 1-3 TTh 4-6 MW 6-8
MW 4-6 MW 11-1 MW 9-11

Lehman College, CUNY

CMP 230 Exam 1, Version 3, Spring 2013

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1. What will the following code print:

```
presidents = "GeorgeXJohnXThomasXJamesXJamesXJohnXAndrewXMartin"  
print("w", presidents[13], presidents[47], presidents[12], presidents[5])  
num = presidents.count("X") + 1  
names = presidents.split("X")  
print("The first", num, "presidents are", names)  
message = names[-2].upper()  
print(message, "???)")
```

Output:

2. Write a **function** that takes as input a string of words separated by semicolons and returns the list of the words in lower case.

3. What will the following code print:

(a)

```
s = "abc def ghi jkl"  
for i in [0,4,6,9]:  
    s = s[:i] + s[i+1] + s[i+2:]  
    print(s)
```

Output:

(b)

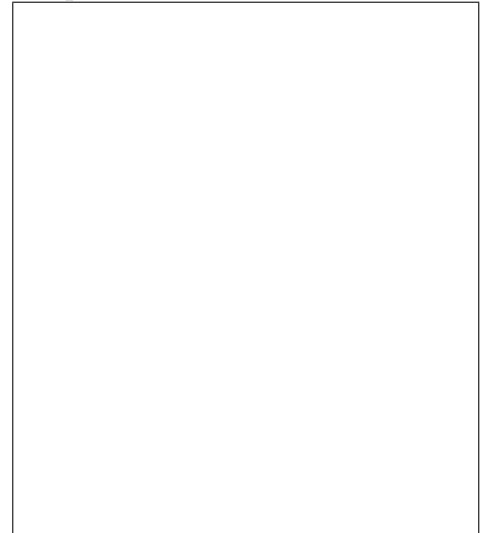
```
s = "hmflqml"  
for i in s:  
    n = ord('a')+(ord(i)-ord('a')+2)%26  
    print(i,chr(n))
```

Output:

4. What will the following program print:

```
def content():
    print(he()+"is content")
def governs():
    print(he()+"governs his passions")
def learns():
    print(he()+"learns from everyone")
def he():
    return("He that ")
def repeat(qual):
    print("Who is "+qual+"?")
    return(1)
def end():
    print("Nobody")
def main4_3():
    count = repeat("wise")
    learns()
    count = count + repeat("powerful")
    governs()
    count = count + repeat("rich")
    content()
    count = count + repeat("that")
    end()
    print("Count = ", count)
main4_3()
```

Output:



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

```
def main():
    welcome()           # Prints "Welcome" to the screen
    x,y = userInput()  # Asks user for 2 numbers and returns them
    z = calculate(x,y) # Raises x to the power y and returns the result
    displayResults(z)  # Prints z to the screen
main()
```

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

6. What is returned when the function is invoked on the inputs below:

```
def mystery(s1):  
    n = s1.find(",")  
    result = -1  
    for i in range(n):  
        result = result + i  
    return(result)
```

(a) `mystery("why, o why, why?")`

Return:

(b) `mystery("very lost")`

Return:

(c) `mystery("I am, you are, we are!")`

Return:

7. What will the following code print:

```
s = "And the dish ran away with the spoon."  
m = ""  
M = ""  
for i in range(0,len(s)-1,2):  
    m = m + s[i]  
    M = M + s[i+1]  
print("m = ",m)  
print("M = ",M)
```

Output:

8. Write a function that takes as a parameter a list of strings and returns a list containing the strings in reverse order. That is, if the input parameter is ["This", "is", "an", "Example"], your function should return ["Example", "an", "is", "This"].

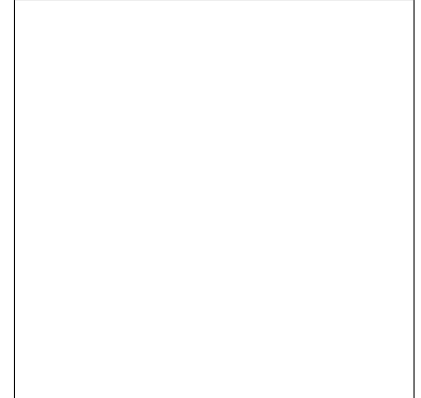
9. Given the following program and input file, what is printed:

```
def filter(words):  
    print(words[0])  
  
def main():  
    infile=open("jefferson.txt","r")  
    lines=infile.readlines()  
    for line in lines:  
        words=line.split(" ")  
        filter(words)  
main()
```

jefferson.txt

```
I believe that  
banking institutions  
are more  
dangerous to  
our liberties  
than standing  
armies
```

Output:



10. Write a **program** that reads in a text file, `infile.txt`, and writes out the lengths of each of the lines to a second file, `outfile.txt`, in order, 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(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.

Useful Unicode Ordinal Numbers

letter	Unicode
A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72
I	73
J	74
K	75
L	76
M	77
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

letter	Unicode
a	97
b	98
c	99
d	100
e	101
f	102
g	103
h	104
i	105
j	106
k	107
l	108
m	109
n	110
o	111
p	112
q	113
r	114
s	115
t	116
u	117
v	118
w	119
x	120
y	121
z	122

NAME:

EMAIL:

SIGNATURE:

CIRCLE COURSE SECTION: TTh 11-1 MW 1-3 TTh 4-6 MW 6-8
MW 4-6 MW 11-1 MW 9-11

Lehman College, CUNY

CMP 230 Exam 1, Version 4, Spring 2013

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

1. What will the following code print:

```
presidents = "WashingtonXAdamsXJeffersonXMadisonXMonroeXAdamsXJacksonXVanBuren"  
print(presidents[19], presidents[8], presidents[60], presidents[9], presidents[12])  
num = presidents.count("X") + 1  
names = presidents.split("X")  
print("The first", num, "presidents are", names)  
message = names[-3].upper()  
print(message, "???" )
```

Output:

2. Write a **function** that takes as input a string of words separated by periods and returns the list of the words with the leading white space stripped.

3. What will the following code print:

(a)

```
s = "abc def ghi jkl"  
for i in [1,5,7,10]:  
    s = s[:i] + s[i+1] + s[i+2:]  
    print(s)
```

Output:

(b)

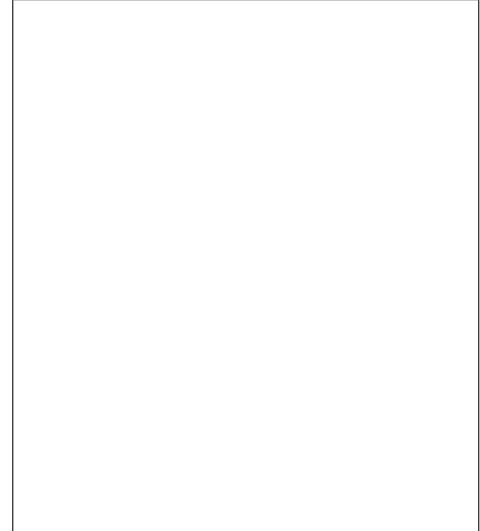
```
s = "gbccboplk"  
for i in s:  
    n = ord('a')+(ord(i)-ord('a')+3)%26  
    print(i,chr(n))
```

Output:

4. What will the following program print:

```
def repeat(s):
    print("Education is " + s)
    return(1)
def first():
    print("Some thoughts about education:")
def life():
    return("life")
def of():
    return("of a")
def end():
    print("What do you think?")
def verses():
    first()
    count = repeat("not preparation for "
                   + life())
    count = count + repeat(life() + " itself")
    count = count + repeat("not the filling "
                           + of() + " pail")
    count = count + repeat("the lighting "
                           + of() + " fire")
    count = count + repeat("a progressive discovery "
                           + "of our ignorance")
    end()
    return count
def main4_4():
    print("Count = ", verses())
main4_4()
```

Output:



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

```
def main():
    welcome()           #Prints "My Program" to the screen
    a,b = userInput()  #Asks user for 2 numbers and returns them
    c = calculate(a,b) # Returns the square root of (a * b)
    displayResults(c)  #Prints c to the screen
main()
```

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

6. What is returned when the function is invoked on the inputs below:

```
def mystery(s1):  
    n = s1.rfind(",")  
    result = 1  
    for i in range(n):  
        result = result * i  
    return(result)
```

(a) `mystery("why, o why, why?")`

Return:

(b) `mystery("very lost")`

Return:

(c) `mystery("I am, you are, we are!")`

Return:

7. What will the following code print:

```
s = "She sells sea shells on the sea shore."  
m = ""  
M = ""  
for i in range(0,len(s)-1,2):  
    m = m + s[i]  
    M = M + s[i+1]  
print("m = ",m)  
print("M = ",M)
```

Output:

8. Write a function that takes as a parameter a list of strings and returns a list containing the length of each of the strings in reverse order. That is, if the input parameter is ["This", "is", "an", "Example"], your function should return [7, 2, 2, 4].

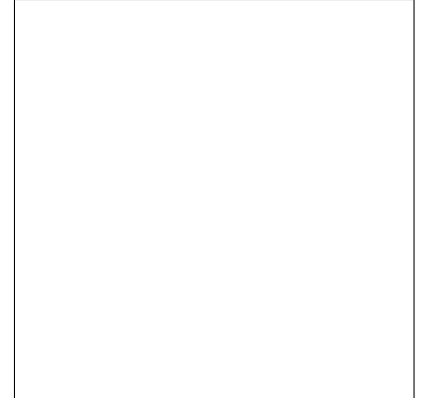
9. Given the following program and input file, what is printed:

```
def filter(words):  
    print(words[1])  
  
def main():  
    infile=open("johnson.txt","r")  
    lines=infile.readlines()  
    for line in lines:  
        words=line.split(" ")  
        filter(words)  
main()
```

johnson.txt

```
Until justice is  
blind to color  
until education is  
unaware of race  
emancipation will  
be a proclamation  
but not a fact
```

Output:



10. Write a **program** that reads in a text file, `infile.txt`, and writes out the first five lines and the last five lines to another file, `outfile.txt`.

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.

Useful Unicode Ordinal Numbers

letter	Unicode
A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72
I	73
J	74
K	75
L	76
M	77
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

letter	Unicode
a	97
b	98
c	99
d	100
e	101
f	102
g	103
h	104
i	105
j	106
k	107
l	108
m	109
n	110
o	111
p	112
q	113
r	114
s	115
t	116
u	117
v	118
w	119
x	120
y	121
z	122