

## Basic type operations

What is the output of the following code snippet? Write your answer inside the box below.

```
print 'foo'
x = 'foo\n'
print(x * 2)
print(3/2)
print(len(x))
y = []
if y:
    print(1 + 2.0 / 4)
else:
    print((5 % 3) / 4.0)
```

## Loops

What is the output of the following code snippet? Write your answer inside the box below.

```
a = 5
while a > 0:
    c = ""
    for b in range(a):
        c = c + str(b)
    print(c)
    a -= 1
```

## Conditionals

What is the output of the following code snippet? Write your answer inside the box below.

```
for a in range(10):
    if a < 5:
        if a < 3:
            print('<')
        elif a == 3:
            print('=')
        else:
            print('>')
    else:
        if a >= 7:
            print('foo')
        if (a < 8) and (a > 6):
            print('bar')
```

## Indexing

What is the output of the following code snippet? Write your answer inside the box below.

```
a = range(10)
for b in a:
    print(a[b:(b%2)+1])
```

## Functions

Consider the function `f()` below.

```
def f(a=True, b=0, c=''):
    x = ["I'm just a", "poor boy", "nobody loves me",
         "He's just a"]
    if c:
        if c.lower().find('poor boy') == -1:
            z = c
        else:
            z = x[b]
    if a:
        print(z)
    else:
        print(z, end='')

```

Using calls to `f()` **only**, generate the following output. Write your answer in the box at the right.

```
I'm just a poor boy nobody loves me
He's just a poor boy from a poor family

```

Hints:

```
lower(...)
    S.lower() -> string

```

Return a copy of the string `S` converted to lowercase.

```
find(...)
    S.find(sub [,start [,end]]) -> int

```

Return the lowest index in `S` where substring `sub` is found, such that `sub` is contained within `s[start:end]`. Optional arguments `start` and `end` are interpreted as in slice notation.

Return `-1` on failure.

## Offline programming

Using the Python Standard Library only, complete the function below.

```
def itemfreq(a):
    """Returns a 2D list of item frequencies.

```

Column 1 contains item values, column 2 contains their respective counts.  
Assumes a 1D array is passed.

Parameters

-----

`a` : array

Returns

-----

A 2D frequency table (col `[0:n-1]`=scores, col `n`=frequencies)

Examples

-----

```
>>> a = [1, 1, 2, 2, 4, 5, 3, 4, 5, 3]
>>> itemfreq(a)
[[ 1,  2], [ 2,  2], [ 3,  2], [ 4,  2], [ 5,  2]]"""

```