

## Problem 1: string comparison

Hint: the following is True: "" < "0" < "9" < "A" < "Z" < "a" < "z"

Circle the expressions that are True:

"a" < "z"	"ax" < "ay"	"abc" < "abCd"
"a" < "Z"	"x2" < "x1"	"zero" < "999"
"x" < "x"	"abcX" < "abcY"	"10" < "999"
"0" < "x"	"abcX" < "aBcY"	"1000" < "999"
"1" < "0"	"abc" < "abcd"	"888888888888" < "9"

## Problem 2: string functions

Functions: upper, lower, strip, rstrip, lstrip, format, startswith, endswith, find.

Expression:	Value (put in quotes):
"dog".upper()	"DOG"
"Dog".lower()	"dog"
" paint ".strip()	"paint"
" paint ".rstrip()	" paint"
"val: {}".format(99)	"val: 99%"
"{} {}".format("X", "Y")	"X Y"

Expression:	Value
"abcd".startswith("ab")	True
"abcd".endswith("bc")	False
"abcd".find("a")	0
"abcd".find("c")	2
"abcd".find("B")	-1
"Python".find("th")	2

## Problem 3: sequence indexing

Assume **msg** is "Hello" and **x** is "num= 13". Some expressions cause an error.

Expression	Result
"abc"[0]	a
"abc"[2]	c

Expression	Result
msg[4]	0
msg[5]	error

Expression	Result
x[len(x) - 1]	3
x[3]	=

"abc"[-1]	c
-----------	---

msg[len(msg)]	error
---------------	-------

x[1] + x[2]	um
-------------	----

## Problem 4: sequence slicing

Assume *msg* and *x* are as before, and *p* is "=".   
*msg* = "Hello"   
*x* = "num = 13"

Expression	Result
"abcde"[0:2]	ab
"abcde"[2:6]	cde
"abcde"[2:9]	cde

Expression	Result
msg[:2]	He
msg[2:]	llo
msg[-2:]	lo

Expression	Result
x[:x.find('=')]	num
x[x.find('')+1:]	13
x[x.find(p)+len(p):]	13

3      2

## Problem 5: for loop over sequence

What does the following code print?

```
msg = "301"
A = ""
B = ""
```

301  
301  
301

```
for character in msg:
    print(msg)
    A = A + character + "."
    B = character + B
```

What is in A afterwards? 3.0.1.

What is in B afterwards? 103

## Problem 6: for loop over range

What does this code print?

```
s = "PYTHON"
for i in range(len(s)):
    print(s[:i+1])
```

P  
PY  
PYT  
PYTH  
PYTHO  
PYTHON  
PYTHON