Language Basics

Check all syntactically correct statements.

1.) Which input statements are correct?

☐ `a = raw_input()`

☐ `a = raw_input("enter a number")`

☐ `a = raw_input(enter your name)`

2.) Which print statements are correct?

☐ print "9" + "9"

☐ print int("nine")

☐ print str(9) + "nine"

☐ print 9 + 9

☐ nine = 9
print nine

3.) Which are correct arithmetical operations?

☐ `a = 1 * 2`

☐ `2 = 1 + 1`

☐ `5 + 6 = y`

☐ `seven = 3 * 4`

4.) Which are correct variable names?

☐ `result`

☐ `my.result`

☐ `print`

☐ `result77`

5.) Which are correct type conversions?

☐ `int(7.0+0.1)`

☐ `str(1.2 * 3.4)`

☐ `float("77"+".0")`

☐ `str( 9 / 0 )`

6.) Which operations result in 8?

☐ `65 // 8`

☐ `17 % 9`

☐ `2 ** 4`

☐ `64 ** 0.5`

7.) Which lines are commented?

☐ `"""This is a comment"""`

☐ `# This is a comment`

☐ `// this is a comment`

☐ `'''This is a comment'''`

Data Types

Find the matching pairs of expressions and values.

1023

None

[2, 4, 8, 16]

True

17.54

('Roger', 1952)

"my fat cat"

{'name': 'Roger', 'birth':1952}

boolean

int

tuple

list

str

dict

NoneType

float

Manipulating Strings

Write the result of each operation into the fields.

```
s.split()
```

```
s[-6]
```

```
s.upper()
```

```
s.strip()
```

```
s = "my fat cat\n"
```

```
s.startswith("cat")
```

```
s.replace("c","r")
```

```
s.find("fat")
```

```
s[7:10]
```
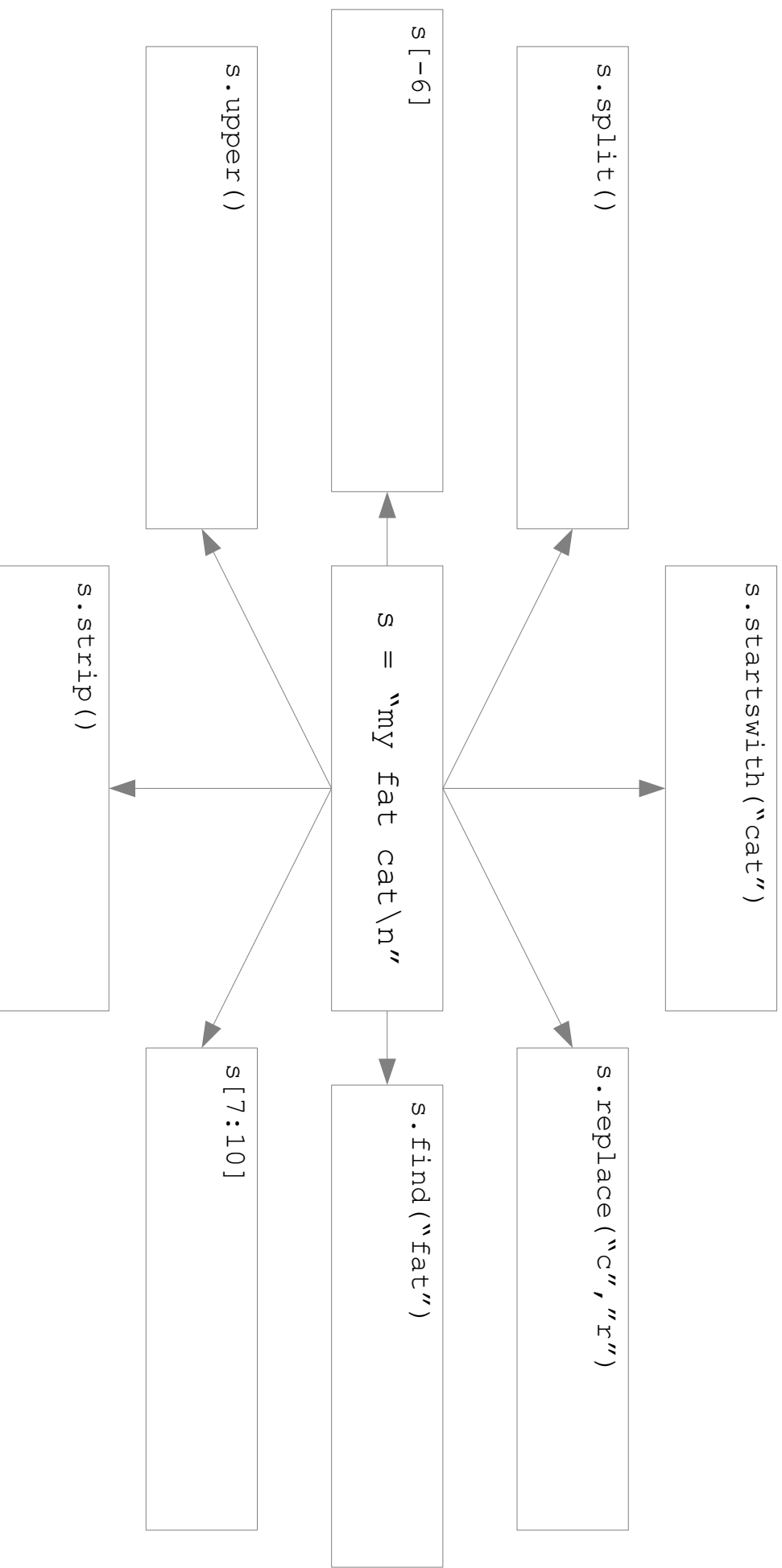
String formatting

Find the matching pairs of expressions and values.

expressions

"R.%s"%("GB")

"%2i. %4s"%(3, "RGB")

"%5.2f"%3.1415

'%s\t%s\tRG'%(B,'\t')

"%5i"%(3.1415)

"RG%%4iB"%(7)

values

3. RGB

3.14

R.GB

RG   7B

3

B   RG

Manipulating Lists

Write the result of each operation into the fields.

```
li.count(0)
```

```
li.append("A")
```

```
2 in li
```

```
len(li)
```

```
li = [2,0,1,0]
```

```
li[1:-1]
```

```
li.pop()
```

```
li.sort()
```

```
li[2]
```
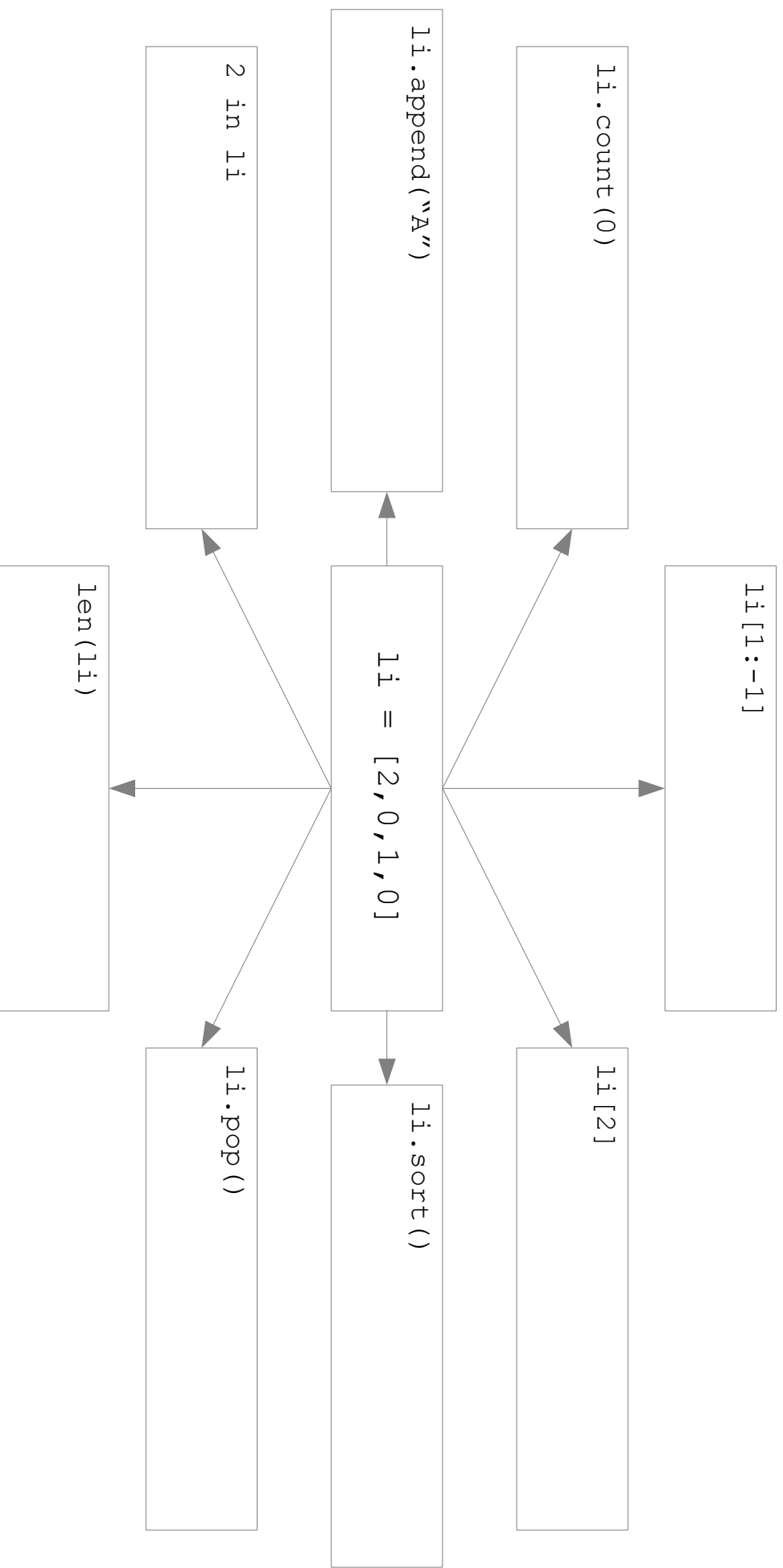
Functions operating on lists (1)

Write the correct operations to the arrows.

a = ['A','B','C',None,3]

a == ['A','B','B','C',None,3]

a == ['B','C',None]

a == ['C','C']

a == ['C','C','C']

a == ['C','C']
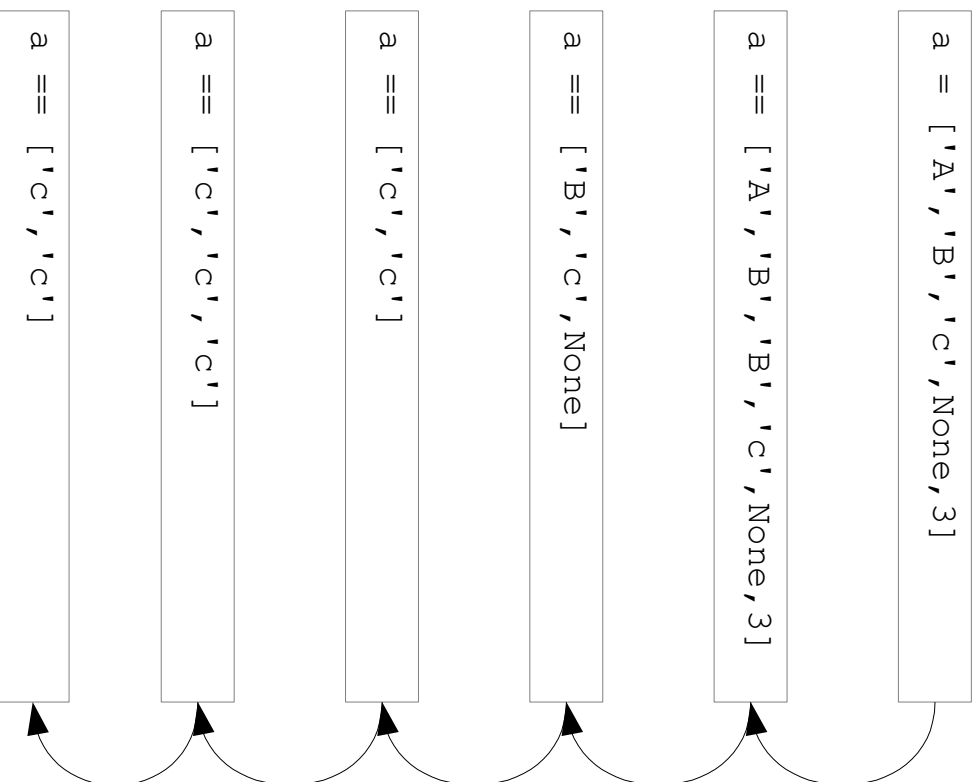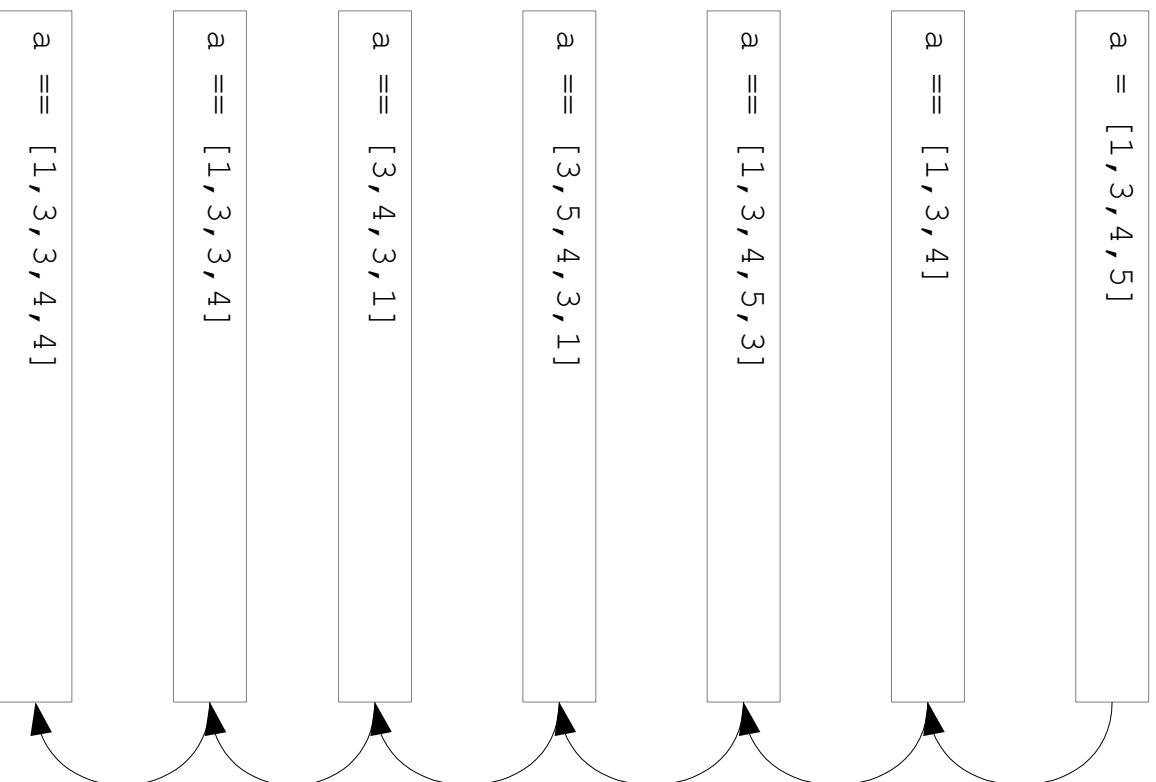
①   a = a[2:5]

②   a = [a[-2]] + [a[1]]

③   a = a[:2]

④   a = [a[-1]]*3

⑤   a = a[:2] + a[1:]

Functions operating on lists (2)

Write the correct operations to the arrows.

a = [1,3,4,5]

a == [1,3,4]

a == [1,3,4,5,3]

a == [3,5,4,3,1]

a == [3,4,3,1]

a == [1,3,3,4]

a == [1,3,3,4,4]

(1) a.reverse()

(2) a.sort()

(3) a.pop()

(4) a.append(4)

(5) a = a + [5,3]

(6) a.remove(5)

Manipulating Lists

Check the correct answer.

1.) What does the list *b* contain?

```
a = [8,7,6,5,4]
b = a[2:4]
```

☐ [7,6,5]

☐ [7,6]

☐ [6,5]

☐ [6,5,4]

2.) Which of the following code pieces results in

```
a == [2,4,6]
```

☐ `a = [1,2,3] * 2`

☐ `a = [int(s) for s in "246"]`

☐ `a = [x*2 for x in range(3)]`

☐ `a = [2**1]+[2**2]+[2**3]`

Working with Tuples

Check all correct answers.

1.) Which are correct tuples?

☐ `( 1, 2, 3 )`

☐ `("Jack" "Knife")`

☐ `('blue', [0,0,255])`

☐ `[ 1, "word" ]`

2.) What are tuples good for?

☐ Grouping data.

☐ Managing values that change.

☐ Running a for loop over them.

☐ Sorting.

3.) On what data types does the len() function work on?

☐ lists

☐ dictionaries.

☐ strings.

☐ tuples.

Loops and conditional statements

Check the correct statements.

1.) Which of these while commands are correct?

- [ ] `while a = 1:`
- [ ] `while a+7:`
- [ ] `while len(c)>10:`
- [ ] `while a and (b-2 == c):`
- [ ] `while b==1`
- [ ] `while s.find('c')>=0:`

2.) Which of these statements are correct?

- [ ] 'while' is also called a conditional loop.
- [ ] The expression after while may contain function calls.
- [ ] It is possible to write endless while loops.
- [ ] The colon after while may be omitted.
- [ ] The code block after while is executed at least once.

3.) What are possible structures of a conditional statement?

- [ ] if <expression> .. else
- [ ] if <expression> .. else if <expression>
- [ ] if <expression> .. else <expression>
- [ ] if <expression> .. elif <expression> .. else
- [ ] if <expression> .. elif <expression> ..else <expression>
- [ ] If <expression>.. else <expression> .. efli

4.) Which of these for commands are correct?

- [ ] `for char in "ABCD":`
- [ ] `for i in range(10):`
- [ ] `for num in (4,6,8):`
- [ ] `for k in 3+7:`
- [ ] `for (i=0; i<10; i+=1):`
- [ ] `for var in seq:`

5.) Which of these if statements are syntactically correct?

- [ ] `if a and b:`
- [ ] `if len(s) == 23:`
- [ ] `if a but not b<3:`
- [ ] `if a ** 2 >= 49:`
- [ ] `if a != 3:`
- [ ] `if (a and b) or (c and d):`

K. Rother, A. Philips
& K. Milanowska

While loops

Match the expressions for the while loops run the designated number of times

```
a = 5
while ☐ :
    a = a -1
```
5x

```
a = 2
while ☐ :
    a += 4
```
5x

```
a = 2
while ☐ :
    a = -a * 2
```
2x

```
a = 7
while ☐ :
    a -= 2
```
4x

```
a != 0
```

```
a >= 0
```

```
a < 19
```

```
abs(a) < 7
```