Exam 1 Review – 1.a, 1.b

```
1.a      bin(len(str(10<20))) →
         bin(len(str(True))) →
         bin(len('True')) →
         bin(4) →
         '0b100'   string

1.b   4L*len("Problem1")/(len("Exam1")%3)+2.0 →
      4L*8/(5%3)+2.0 →
      4L*8/2+2.0 →
      32L/2+2.0 →
      16L+2.0 →
      18.0      float
```

Exam 1 Review – 1.c, 1.d

```
1.c      len(‘twelve’+‘four’)/(len(‘2L’)+2*(-1)**3) ➔
         len(‘twelvefour’)/(len(‘2L’)+2*(-1)**3) ➔
                   10/(2+2*(-1)**3) ➔
                     10/(2+2*-1) ➔
                       10/(2-2) ➔
                         10/0 ➔
                  Error – divide by zero

1.d      round(math.pi, 2) < math.trunc(math.pi) ➔
                        3.14 < 3 ➔
                      False    bool
```

Exam 1 Review – 1.e, 1.f

1.e 20 < 30 and 50 < 25 and 4 < 100/0 ➔

True and False and 4 < 100/0 ➔

False and 4 < 100/0 ➔

False bool

1.f str(12 + 4.0/2**2) ➔

str(12 + 4.0/4) ➔

str(12 + 1.0) ➔

str(13.0) ➔

"13.0" str

Exam 1 Review – 1.g, 1.h

1.g 3.0 * input("Enter an integer: ") ➔

Input is 10/2 + 1

3.0 * 6 ➔

18.0

1.h not bool(0.1) or not (100 and not False) ➔

not True or not (100 and True) ➔

False or not True ➔

False or False ➔

False

Exam 1 Review – 1.i, 1.j

1.i 2*float("sys.maxint") ➔

Error - "sys.maxint" does not convert to a float

1.j len(str(random.randint(1,4)*22))➔

len(str(one of [1,2,3,4]*22))➔

len(str(one of [22,44,66,88]))➔

len(one of ['22','44','66','88']))➔

2 int

Exam 1 Review – 2a

```
x = 10
while x < 120:
    y = x + 40
    while (x < y) :
        if (y % 10) == 5 :
            y = y + 15
        else :
            print x, y
            y = y - 35
    x = x + 30
```

```
10 50
10 30
40 80
40 60
70 110
70 90
100 140
100 120
```

x	y
10	50
	15
	30
	-5
40	80
	45
	60
	25
70	110
	75
	90
	65
100	140
	105
130	

Exam 1 Review – 2b

```
x = 0
while x < 10:
    if x % 2 == 0:
        y = x + 1
        while (y < 11) :
            print "Line 1:", x, y
            y = y + 4
    else :
        y = 11-x
        while (y > 1) :
            print "Line 2:", x, y
            y = y - 3
    x = x + 3
```

```
Line 1: 0 1
Line 1: 0 5
Line 1: 0 9
Line 2: 3 8
Line 2: 3 5
Line 2: 3 2
Line 1: 6 7
Line 1: 9 2
```

x	y
0	1
	5
	9
	13
3	8
	5
	2
	-1
6	7
	11
9	2
	-1
12	

Exam 1 Review – 2c

```
n = int(raw_input(“Please type something.”))

answer = “”
while n > 0:
    if n % 10 > 0:
            answer = str(n) + answer
    print answer
    n = n/10
```

n	answer
2079	“”
	“2079”
207	“2072079”
20	
2	“22072079”
0	

```
2079
2072079
2072079
```

Exam 1 Review – 2d

```
print "Type positive integers, one per line and then 0 to be done."

n = 1
while n:
    n = int(raw_input())
    if (n/1000 == n%10) and ((n/100)%10 == (n%100)/10):
        print n
```

n
1
8033
3443
2117
2332
0

```
Type positive integers, one per line and then 0 to be done.
3443
2332
0
```

Exam 1 Review – 3a

```
b = int(raw_input("Type an integer in binary. "))
bCopy = b
answer = 0
placeValue = 1 # tracks place values, from right to left

while b:
    bit = b % 10 # extract the right most bits

    # update the answer using the right most bit (Blank 1)
    answer = answer + placeValue*bit

    #update b (Blank 2)
    b = b/10

    # update placeValue
    placeValue = 2*placeValue

print "The decimal equivalent of", bCopy, "is", answer
```

Exam 1 Review – 3b

```
import sys

# Variable used to read input numbers
current = int(raw_input("Type a positive int (zero if done)."))

# tracks the number just prior to the most recently read number
previous = sys.maxint

# counter to track the number of consecutive, increasing pairs
numIncreasingPairs = 0

while current:
    if current > previous:
        numIncreasingPairs = numIncreasingPairs + 1

    # Update previous (Blank 1)
    previous = current

    # Update current (Blank 2)
    current = int(raw_input("Type a positive int (zero if done)."))

print numIncreasingPairs
```

Exam 1 Review – 4

4a)

```
def startsWithDigit(word):

    n = 0    # digit to be checked
    # check each digit
    while n < 10 :
       # if word starts with n, return True
       if startsWith(word, str(n)) :
            return True
        n = n + 1

  # Word does not start with a digit, return False
  return False
```

4b)

```
 password = raw_input("Please enter your password:”)
 if len(password) >= 6 and startsWithDigit(password) :
    print "Password accepted"
else :
    print "Your password is not valid”
```