1. True or False for each block of proceeding code: The code will output “this is lowercase now”

```
phrase = “ThIs Is LoWeRCaSe NoW”
phrase.lower()
print(phrase, end=’’) # False → Output: “ThIs Is LoWeRCaSe NoW”

-------------

phrase = “ThIs Is LoWeRCaSe NoW”
for c in phrase:
     phrase[c] = phrase[c].lower()
print(phrase, end=’’) # False → TypeError: string indices must be integers

-------------

phrase = “ThIs Is LoWeRCaSe NoW”
new_phrase = “”
for c in phrase:
     if c>= “a” and c <= “z”:
          new_phrase += c
     else:
          new_phrase += c.lower()
print(new_phrase, end=’’) # True → Output: “this is lowercase now”
```

2. True or False for each block of code: The code will result in a runtime error

```
p = 41
try:
    f = "scarlet"
    p += 24
except:
    f = input("Enter a number: ") #user enters x
    f *= p
    f[5] = "s"
else:
    f += "pink"
print(f[3], end ="") # False → Output: “r”
-------------
d = "dragon"
try:
    d *= 10
except:
    d *= 3.0
else:
    d += "fruit"

print(format(d, ".2f"), end = "") # True → ValueError:
Unknown format code 'f' for object of type 'str'
-------------
p = "peanut"
try:
    p *= 3.0
except:
    p *= 4
else:
    p += "butter"

print(format(p, "<100s"), end = "") # False → Output:
“peanutpeanutpeanutpeanut”
```

3. Use the following list and dictionary to answer the questions below.

```
c = [['x',100], ['y', 200]]
d = {'x':100, 'y':200}
```

a) How could we access the number 100 from list c? `c[0][1]`

b) Write two ways of retrieving the value at key, 'x' from the dictionary

```
d.get('x')
d['x']
```

c) Write two ways of adding ['z', 300] to the list c.

```
c.append(['z', 300])
c += [['z', 300]]
```

d) Write the correct way of adding a new key value pair, 'z':300, to the dictionary, d.

```
d['z'] = 300
```

e) Write the correct way of removing the element that contains x in list c.

```
for i in range(len(c)):
    if c[i][0] == "x":
        index_to_remove = i
del c[index_to_remove]
```

f) Write the correct way of removing the element, 'x', from the dictionary

```
del d["x"]
```

g) Convert list c into dictionary d

```
new_dict={}
for i in range(len(c)):
   new_dict[c[i][0]] = c[i][1]
```

h) Write 3 ways to iterate over the dictionary, d.

```
for i in d:
for k in d.keys():
for v in d.values():
for k,v in d.items():
```

4. Given the data file below (datafile.txt), what will be stored in the file at the end of this program?

datafile.txt
snow
snowflake
sled
sledding

```
file_object = open("datafile.txt", "r")
data = file_object.read()
file_object.close()
x = data.split("\n")

file_object_2 = open("datafile.txt", "w")

for i in x:
     if len(i) % 2 == 0:
          file_object_2.write(i + "!\n")
     else:
          file_object_2.write(i + "?\n")

file_object_2.close()
```

datafile.txt
snow!
snowflake?
sled!
sledding!

5. For this problem you will need to use Python to compute the sum of all VALUES in a list that occupy INDEXES that are both (a) EVEN and (b) DIVISIBLE 3.

For example - if the list is holding [3, 8, 12, 27, 29] your program would compute the value 18 which was computed by adding 3 and 12 which occupy the indexes 0 and 2. Note that 0 is evenly divisible by any integer.

```
the_list = [496, 323, 1290, 3704, 3669, 3302, 3691, 3786, 1099, 2063,
1601, 1284, 1127, 1127, 1067, 2632, 1551, 2378, 2521, 1880, 179,
1005, 531, 1514, 695, 2749, 3813, 1108, 1270, 2242, 880, 646, 852,
2557, 186, 1507, 547, 3608, 1965, 1092, 2894, 1702, 1385, 3606, 2573,
2328, 104, 434, 636, 2471, 3491, 2197, 3411, 1686, 2523, 599, 967,
787, 3422, 3369, 2799, 1068, 2728, 2791, 2107, 2373, 3372, 3180,
2627, 2146, 1317, 1328, 2602, 3802, 3406, 1447, 3003, 2827, 1435,
2465, 3014, 2559, 547, 1235, 2401, 3366, 3246, 1198, 2711, 1073, 953,
2880, 328, 2780, 3679, 1385, 1055, 1466, 671, 374, 2316, 2774, 2595,
2768, 3320, 244, 1944, 101, 1369, 3289, 267, 2525, 3153, 390, 1576,
2302, 156, 2751, 1711, 1756, 1039, 2736, 368, 2495, 2658, 2939, 96,
1774, 1137, 2400, 3202, 1356, 3275, 3663, 3265, 3782, 3309, 1970,
1409, 45, 1441, 1560, 2438, 795, 1591, 323, 121, 896, 2627, 2283,
1426, 1819, 484, 1953, 3369, 2058, 198, 665, 3804, 3374, 3169, 1516,
1139, 2987, 487, 2557, 738, 2419, 2037, 3262, 974, 691, 2709, 2886,
2405, 836, 1837, 1749, 1734, 3003, 3208, 1568, 3274, 3339, 1008,
1644, 3414, 1720, 2106, 3507, 2780, 2686, 1171, 1943, 1112, 2167,
300, 95, 579, 1454]
```

```
total = 0
for index in range(len(the_list)):
    if index % 2 == 0 and index % 3 == 0:
        total += the_list[index]
print(total) #57645
```

6. For this problem you will need to use Python to analyze a dictionary. The keys in this dictionary are all strings, and the values are all integers. You will need to compute the sum of all VALUES in the dictionary that are associated with KEYS that contain a ODD number of characters.

For example - if the dictionary is {'r': 27, 'TT': 5, 'RR': 22, 'pp': 24, 'z': 11} your program would compute the value 38 which was computed by adding 27 + 11 which are associated with keys ('r', 'z').

The dictionary you are analyzing can be found below (copy this dictionary into your Python editor) - you only need to submit your final answer (an integer)

```
the_dict = {'LL': 470, 'YYYYY': 3438, 'PPPPP': 1496, 'fffff': 2234,
'ppppp': 3404, 'HHH': 3516, 'yyy': 3461, 'rrr': 2112, 'bbbbb': 3813,
'wwwww': 1529, 'ZZZZ': 1093, 'MMM': 3191, 'dddd': 3431, 'W': 1202,
'p': 623, 'BBBB': 748, 'VVV': 2432, 'WW': 810, 'd': 2842, 'ss': 2239,
'vvvvv': 3175, 'PPPP': 283, 'HHHHH': 3874, 'xx': 2840, 'FF': 3825,
'yyyy': 449, 'xxxxx': 587, 'LLL': 1318, 'C': 2319, 'mm': 3031,
'JJJJJ': 3466, 'TTTT': 1328, 'kkkk': 3824, 'hh': 2175, 'j': 979, 'tt':
2396, 'H': 2038, 'VV': 1390, 'KKK': 3808, 'r': 1498, 'CC': 2840, 'SS':
956, 'K': 254, 'ddd': 3242, 'JJJJ': 1099, 'qqq': 2675, 'MM': 2143,
'c': 2207, 'bb': 1572, 'RRRR': 1377, 'ZZZZZ': 2865, 'DDDD': 3221, 'x':
1291, 'b': 1154, 'FFFF': 1271, 'BBB': 2484, 'zzz': 27, 'PPP': 2260,
'vvvv': 3479, 'FFF': 2125, 'pppp': 1051, 'cc': 3423, 'mmm': 82, 's':
2656, 'WWWW': 2656, 'V': 2810, 'RR': 2522, 'qqqq': 1422, 'zz': 187,
'ff': 3135, 'hhhh': 1436, 'ffff': 786, 'L': 951, 'VVVV': 1247,
'kkkkk': 3396, 'jjjjj': 408, 'KKKK': 283, 'qq': 2769, 'VVVVV': 1268,
'LLLLL': 1085, 'dd': 529, 'bbb': 3500, 'vv': 3591, 'Q': 1077, 'ccc':
1600, 'yyyyy': 3482, 'KKKKK': 3574, 'KK': 1454, 'h': 472, 'zzzzz':
2094, 'TTT': 3535, 'hhh': 2528, 'CCCCC': 1147, 'DDD': 3752, 'XXX':
1754, 'Y': 659, 'sssss': 1040, 'XX': 132, 'WWW': 2448, 'm': 2207}
```

```
total = 0
for key in the_dict:
    if len(key) % 2 == 1:
        total += the_dict[key]
print(total) #126464
```

Copy the class in the box below into your Python editor and write a program that performs the following operations in the order that you see below. Note that your program SHOULD NOT modify the class in any way. Simply write a main program that does the following:

```
class Yellow:

    def __init__(self, L):
        self.L = L
        self.W = ''
        self.S = 0
    def zebra(self):
        return self.W
    def beetle(self):
        d = str(self.L*self.L)
        V = int(d[len(d)//2-3:len(d)//2+3])
        s = int(V/999999 * 26) + 65
        self.L = V
        self.W += chr(s)
        self.S += 1
        return s
    def sardine(self):
        return self.S
    def carrot(self):
        d = str(self.L*self.L)
        V = int(d[len(d)//2-3:len(d)//2+3])
        s = int(V/999999 * 100)
        self.L = V
        self.W += str(s)
        self.S += 1
        return s
    def squirrel(self):
        try:
            return int(self.W)
        except:

            return self.W
```

1. Create 3 objects of type Yellow. Each object should be initialized with a different constructor argument. The 1st object should be initialized with a constructor argument of 825856 (integer). The 2nd object should be initialized with a constructor argument of 814162 (integer). The 3rd object should be initialized with a constructor argument of 635018 (integer).

2. Have the 1st object call its 'beetle' method 7 times
3. Have the 2nd object call its 'carrot' method 2 times
4. Have the 3rd object call its 'carrot' method 8 times
5. Have the 1st object call its 'squirrel' method and store the result.
6. Have the 2nd object call its 'squirrel' method and store the result.
7. Have the 3rd object call its 'squirrel' method and store the result.
8. Concatenate all results returned by the 'squirrel' method that are of a string data type and enter them into the blank below. Add these together based on the order in which the objects were constructed (i.e. if all results are strings you would compute this as 1st object return value + 2nd object return value + 3rd object return value). If only one return value is a string then type that value into the blank below.

```
yel1 = Yellow(825856)
yel2 = Yellow(814162)
yel3 = Yellow(635018)

for i in range(7):
    yel1.beetle()
for i in range(2):
    yel2.carrot()
for i in range(8):
    yel3.carrot()

result1 = yel1.squirrel()
result2 = yel2.squirrel()
result3 = yel3.squirrel()
print(result1, result2, result3)
# AODXCXC 8519 2443857217854397

results = [result1, result2, result3]
output = ""
for r in results:
    if str(r).isalpha():
        output += str(r)
print(output) #AODXCXC
```