Write statement to print 2nd element from list L to print e.

L=[‘h’, ‘e’, ‘l’, ‘l’, ‘o’]

**Print L[1]**

After following is executed, what is type of n[:3]

N=”abcde”

**String**

After following is executed, what is type of int(n[:3])

N=”abcde”

**Error**

What will following print?

Myvar=”hello”

Print myvar[-1]

**O**

What will following print out?

Myvar=”hello”

Print “myvar”[-1]

**R**

After following is executed, what is type of y?

Myvar=”the value of x is”

X=10

Y=myvar+x

**Error**

What will print?

X=3

Y=4

X==y+1

Print x

**3**

After following is executed, what is type of variable t?

S=”2014-10-02T20:12:28+0000”

T=split(“-“)

**List, error**

After following is executed, what is type of variable t?

S=”<published>2009-01-23T20:04:53z</published>”

T=s.find(“2009”)

**Integer**

What will print?

S=”2014-10-02T20:12:28+0000”

Print s[s.find(“14-“)]

**1**

What will print?

S=”2014-10-02T20:12:28+0000”

Print len(s.split(“:”)[2].split(“2”))

**2**

2 files, 1 has full poem, other has compressed version using lossless compression. Which has more info or do they have the same?

**Same- can reconstruct 1 from other bc lossless compression. 1 requires more storage than the other but same amount of info.**

Suppose a comp program convinces judges that it is human. Would Searle say program was “intelligent”?

**No, behaving intelligently does not equal understanding.**

Following unix is run, 1 is 1000 lines, 2 is 2000 lines, nothing in common, how many lines in file3.txt?

Diff file1.txt file2.txt > file3.txt

**About 3000**

Following is executed

L=[“First”, “Second”, “Third”]

For x in L:

Y=x[0]

Print y

**T**

For L in x:

Print L[0]

**Error**

For x in L:

Print x[2] in L[0]

**True**

**False**

**True**

What prints?

X=7

If x<10:

Print “one”

If x<20:

If x>15:

Print “two”

Else:

Print “three”

Elif:

**One**

**Three**

What will print?

X=27

If x<10:

Print “one”

If x<20:

If x>15:

Print “two”

Else:

Print “three”

Elif x<30:

Print “four”

**Four**

What prints?

L=[]

L.append(‘a’)

L.append(‘b’)

L.append(‘c’)

L[‘a’]=0

Print L

**Error**

What prints?

D={}

D[1]=’a’

D[2]=’b’

D[‘c’]=3

Print ‘a’ in d

**False**

What prints?

D={}

D[1]=’a’

D[2]=’b’

D[‘c’]=3

Print d[1]+d[2]

**ab**

Write 1 line of code that does what last 3 lines do

X=”some string”

Y=20

Z=x

X=y

Y=z

**(z, x, y)=(x, y, z)**

What prints?

Def g(x,y=0)

Y=y+x

Return y

Y=10

Z=g(y, 5)

Print z

**15**

What prints?

X=-1

Y=-2

Z=-3

Def h(x, y=2, z=3):

Print x, y, z

H(1,4)

**1 4 3**

Following has been executed

L=[{‘a’:1, ‘b’: 2, ‘d’:11}, {‘a’:4, ‘b’:5, ‘d’:12}, {‘a’:7, ‘b’:8, ‘f’:11}]

Write code to print each value associated with key d

**For diction in L:**

**If ‘d’ in diction:**

**Print diction[‘d’]**

Write code that creates 1 dictionary with 1 key for any key that appears in L with value of sum of keys

**Sums = {}**

**For diction in L:**

**For k in diction:**

**If k in sums:**

**Sums[k] += diction[k]**

**Else:**

**Sums[k]=diction[k]**

Write code that asks user to input numbers and keeps going until > 21

**Sum=0**

**Nums=[]**

**While sum < 21:**

**Nxt=int(raw\_input(“Insert a number”)**

**Sum += nxt**

**Nums.append(nxt)**

**Print nums**

Define function scrabble\_score. Takes a word as input and takes a dictionary that letters=number scores. Should return an integer.

**Def scrabble\_score(word, vals\_dict):**

**Tot=0**

**For c in word:**

**Tot += vals\_dict[c]**

**Return tot**