

Test s14

What will be the output of the following code sequences? Explain your answer.

1.

```
a, b = 1, 2
l = [a, b]
a = 7
l2 = [a]
l.append(l2)
l2[0] = 8
print(a, l2, l)
```

#-----

2.

```
def f(i):
    return i + 1

def g(h):
    return h(1)

def h(x):
    return x + 10

i = lambda x: x + 2
print(f(f(f(g(i)))))
```

#-----

3.

```
def f():
    return 1
def g(x=1):
    return x + 1
def h(x=1, y=2):
    return x + y
l = [f, g, h]
for e in l:
    print(e())
h = lambda x = 1, y = 2: x * y
print(l[2](3), h(), h(3), h(x=3), h(y=3))
print(h([2, 3]))
print(h(*[2, 3]))
```

#-----

4.

```
#module m1
a = 1
def f(b):
    a = b
    return a
def h():
    return a

#module m2
from ro.ubb.t.ts14 import m1
a = 2
def g():
    global a
    a = m1.f(3) + 1
    return m1.h()

print(g(), a)
```

5.

```
class A:
    def f(self):
        return 1
def g():
    return 2
a = A()
a.f = g
a2 = A()
A.f = lambda x: 3
l = [a, a2, A()]
for e in l:
    print(e.f())
```

#-----

6.

```
s = 0
def f(x):
    global s
    if x > 1:
        s = s + x
        f(x - 1)
    s = s + x

f(3)
print(s)
```

#-----

7.

```
class A:
    def __init__(self, k):
        self.__k = k
        self.__items = []
    def add(self, a):
        self.__items.append(a)
        return self
    def f(self, l):
        if l == 0:
            print(self.__k)
            for a in self.__items:
                a.f(l - 1)

a = A(1)
a.add(A(2))
a.add(A(3).add(A(5).add(A(7))))
a.add(A(4).add(A(6)))
a.f(2)
```

8. Specify and test the following function:

```
def f(l):
    if len(l) == 0:
        raise ValueError
    s = 0
    for i in range(len(l)):
        if i % 2 == 0:
            s+=l[i]
        else:
            s-=l[i]
    return s
```

9. Given a sorted list *l*, specify and test a function that searches a given value *v* in the list *l*.

10. Write a recursive algorithm for the problem from point 9.

11. Write an algorithm with the complexity $O(\log n)$ for the function from point 9.

12. Using the divide and conquer method, compute the product of all even elements from a list.

