

Programming and Data Structures in Python

Duration: 1 Hour and 15 minutes

Maximum Marks: 15

1. Consider the following python code and describe its output. (there was a typo in the version used in the test, alas)

```
x = [12,[13,14],15]
y = [9]
z = (x,y)
w = x + []
w[1].append(5)
z[0][1].append(19)
print(x)
print(y)
print(z)
print(w)
print(x[1] is w[1])
print(x[1] is z[0][1])
```

(3 marks)

2. Consider the following python code and describe its output:

```
x = 10

def f():
    def g():
        nonlocal x
        x = 20
        print(x)
    def h():
        global x
        x = 30
        print(x)
    x = 15
    h()
    g()
    print(x)

f()
print(x)
```

(3 marks)

3. An aliases list is a list of pairs where the first component of each pair is a string called "alias" and the second is a list of strings (representing say "usernames"). Here is an example

```
[ ("BscM", ["aa", "bb", "cc"]),
  ("BscP", ["ef", "gh", "ij"]),
  ("New", ["aa", "BscP"]),
  ("Bsc", ["New", "bb", "cc"]),
  ("Rest", ["zz", "yy"])]
```

Each line defines the alias as the collection of usernames given in the second component. No alias has multiple definitions. Your aim is to write a function `expandOut` is to take such a list and return one where all occurrences in of aliases in the second components have been removed via *expansion*. For the above list the answer would be

```
[ ("BscM", ["aa", "bb", "cc"]),
  ("BscP", ["ef", "gh", "ij"]),
  ("New", ["aa", "ef", "gh", "ij"]),
  ("Bsc", ["aa", "ef", "gh", "ij", "bb", "cc"]),
  ("Rest", ["zz", "yy"]) ]
```

You may assume that any use of a alias can only come after its definition (so, you cannot have cyclic definitions). (5 marks)

If your program does not require that the definition occurs before usage (but of course there is no cyclic dependency permitted) then you will get 2 bonus marks.

4. Write a function `rearrange` with 3 arguments: Its first argument is a number N , the second argument P is a permutation of $[0, 1, \dots, N-1]$ and the third argument C is a list containing a sequence of distinct values from $[0, \dots, N-1]$. The list C tells you how to rearrange the given permutation P . Say $N = 5$ and $P = [3, 1, 2, 0, 4]$. If the third argument is $[2, 4, 3]$ then it means the following: The value sitting at position 2 must now be moved to position 4, the value sitting at position 4 must be moved to position 3 and finally the value at position 3 should be moved to position 2 (all done simultaneously). So, the result is $[3, 1, 0, 4, 2]$. In general if the sequence is i_1, \dots, i_k then we move the element in position i_1 in P to position i_2 , the one at i_2 to i_3 and so on and finally the one at i_k to position i_1 .

```
rearrange(5, [3,1,2,0,4], []) = [3,1,2,0,4]
rearrange(5, [3,1,2,0,4], [0,4]) = [4,1,2,0,3]
rearrange(5, [3,1,2,0,4], [2,4,3]) = [3,1,0,4,2]
```

(5 marks)

If you solve this problem *in place*, i.e., within the list P without creating a new list then you will get 2 bonus marks.