5. Assume we have a class `Tree` that implements a height-balanced search tree with the usual functions `isempty()`, `isleaf()`, `find(v)`, `insert(v)` and `delete(v)`. You can assume the search tree representation discussed in class, where each node has fields `value`, `left` and `right`, the empty tree has all three fields set to `None` and leaf nodes have empty trees as left and right children.

   We want to add the following functions to the class `Tree` to find the predecessor and successor of a node in a tree.

   - `pred(v)` should return the value `w`, where `w` in the largest node in the tree such that `w < v`. If `v` is not a node in the tree, or if `v` is the smallest node in the tree, return `None`.
   - `succ(v)` should return the value `w`, where `w` in the smallest node in the tree such that `v < w`. If `v` is not a node in the tree, or if `v` is the largest node in the tree, return `None`.

   (a) Write Python code for `pred(v)` and `succ(v)`. *(8 marks)*

   (b) What is the asymptotic complexity of the two functions in your implementation in terms of the size of the tree? Justify your answer. *(3 marks)*

6. Assume we have a class `Heap` that implements a max-heap. The values on the heap are stored level by level, as usual, in an internal list `hlist`. The constructor takes an input sequence of values and heapifies them into `hlist`. The class has functions `isempty()`, `insert(v)` and `deletemax()` with the standard interpretation. We assume that all values in a heap are distinct.

   We wish to add a function `update(v,w)` that changes the value `v` on the heap to the value `w`. For simplicity, we assume that `update(v,w)` is always called in a context where `v` is present in the heap and `w` is not present in the heap and `v != w`, so that the outcome of the operation changes the heap while maintaining the property that all values in the heap are distinct.

   (a) Write Python code to implement `update(v,w)`. Observe that if `v > w`, the new value may violate the heap property with respect to its children, and, if `v < w`, the new value may violate the heap property with respect to its parent. Note that we have to first locate `v` on the heap. For now, scan `hlist` sequentially to find the value `v`. *(6 marks)*

   (b) The update operation can be improved by maintaining an internal dictionary `valuetonode` such that `valuetonode[v] = i` if `hlist[i] = v`. Rewrite `update(v,w)` so that it uses `valuetonode` to locate the value to update and modifies `valuetonode` appropriately as part of the update process. (Recall that `del(d[k])` removes key `k` and its associated value from dictionary `d`.) *(6 marks)*

   (c) What is the asymptotic complexity of the two versions of `update(v,w)`? Explain your answer. *(3 marks)*

7. You are playing an old-style video game in which you have to shoot down alien spaceships as they fly across the screen from left to right. Each spaceship flies across the screen at a specified height. You have an antiaircraft gun set to shoot down all spaceships at a certain height. Spaceships fly one at a time, so if your gun is set to fire at the correct height, it will shoot down the spaceship currently flying across the screen.

   You can set the initial height at which the gun fires. As the game progresses, you can reset the height, but only to a lower value. You are given in advance the height at which each spaceship flies. There are N spaceships numbered $1, 2, \ldots, N$ in the order in which they fly across the screen. For $1 \leq i \leq N$, $h[i]$ denotes the height at which spaceship i flies.

   Let $V[i]$ denote the maximum number of spaceships from $i, i+1, \ldots, N$ that you can shoot down with a single gun, assuming you shoot spaceship i.

   (a) Write a recursive formulation of $V[i]$ in terms of $V[j]$, for $i < j$. *(5 marks)*

   (b) Write Python code to compute $V[1]$ using dynamic programming. *(5 marks)*

Programming and Data Structures using Python 28 November 2023
Final Exam, Aug–Nov 2023 Marks: 50, Weightage: 40%

Note: *In any question that asks for Python code, minor syntax errors will not be penalized.*

1. In the following code, which is the first statement where Python will flag an error. Explain your answer.

```
x = [1,[7,8],2,"efgh",[3,4]]   # Statement 1
y = x[0:50]                    # Statement 2
z = y                          # Statement 3
w = x                          # Statement 4
x[1] = "abcd"                  # Statement 5
y[1][1] = "y"                  # Statement 6
w[1][1] = "9"                  # Statement 7
y[2] = 4                       # Statement 8
z[0] = 0                       # Statement 9
w[4][0] = 1000                 # Statement 10
```

(3 marks)

2. What is the value of `pairs` after the following assignment? Explain your answer.

```
pairs = [ (x,y) for x in range(5) for y in range(4) if (x+y)%2 == 0 ]
```

(3 marks)

3. Explain what the function `mystring` given below computes.

```
def mystery(s):
  for i in range(len(s),len(s)//2,-1):
    s = s[:len(s)-i] + s[i-1] + s[len(s)-i+1:i-1] + s[len(s)-i] + s[i:]
  return(s)
```

(4 marks)

4. Consider the following Python code.

```
def f(l):
  l = l + [l[0]]
  return(l)

def g(l):
  l.append(l[0])
  return(l)

list1 = [4,3,9,7]
list2 = g(list1)
list3 = f(list2)
```

What are the values of `list1`, `list2`, `list3` at the end of the execution? Explain your answers.
(4 marks)