1. What are 2 python sequence types implemented in C language?
   1. Container sequences Can hold items of different types, including nested containers. Someexamples:list,tuple,and collections.deque. A container sequence holds references to the objects it contains, which may be of any type, while a flat sequence stores the value of its contents in its own memory space, not as distinct Python objects.
   2. Flat sequences Hold items of one simple type. Some examples: str, bytes, and array.array

Note: If you are not gonna do anything with the produced list of list comprehension it’d better not to use it and always remember that list comprehension produces a new list.

1. Why tuples are playing such important role in python standard library?
   1. Clarity: When you see a tuple in code, you know its length will never change.
   2. Performance: A tuple uses less memory than a list of the same length, and it allows Python to do some optimizations.
2. Note: be aware that the immutability of a tuple only applies to the references contained in it. References in a tuple cannot be deleted or replaced. But if one of those references points to a mutable object, and that object is changed, then the value of the tuple changes.
3. What are advantages of using tuple over list?
   1. Given a tuple t, tuple(t) simply returns a reference to the same t. There’s no need to copy. In contrast, given a list l, the list(l) constructor must create a new copy of l
   2. Because of its fixed length, a tuple instance is allocated the exact memory space it needs. Instances of list, on the other hand, are allocated with room to spare, to amortize the cost of future appends.
   3. The references to the items in a tuple are stored in an array in the tuple struct, while a list holds a pointer to an array of references stored elsewhere. The indi‐ rection is necessary because when a list grows beyond the space currently alloca‐ ted, Python needs to reallocate the array of references to make room. The extra indirection makes CPU caches less effective.