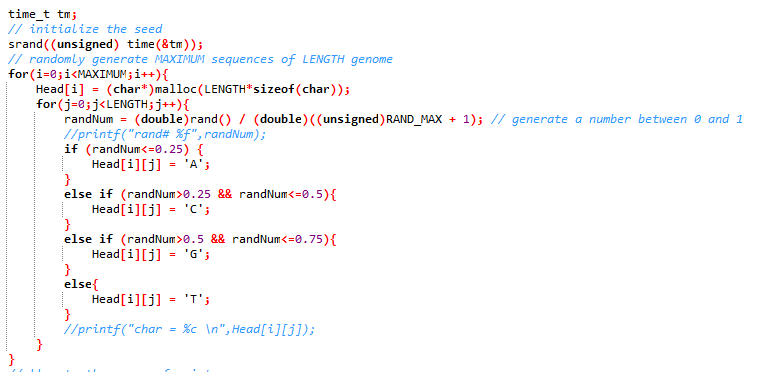
HW#2 INF503

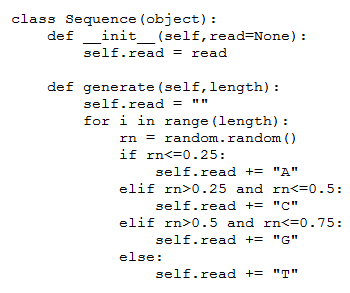
Jiaming Chen

Problem#1

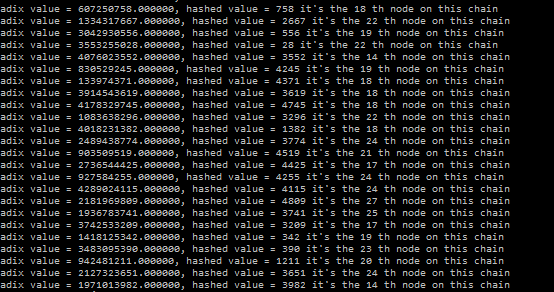
1A: code in C:



Code in Python:



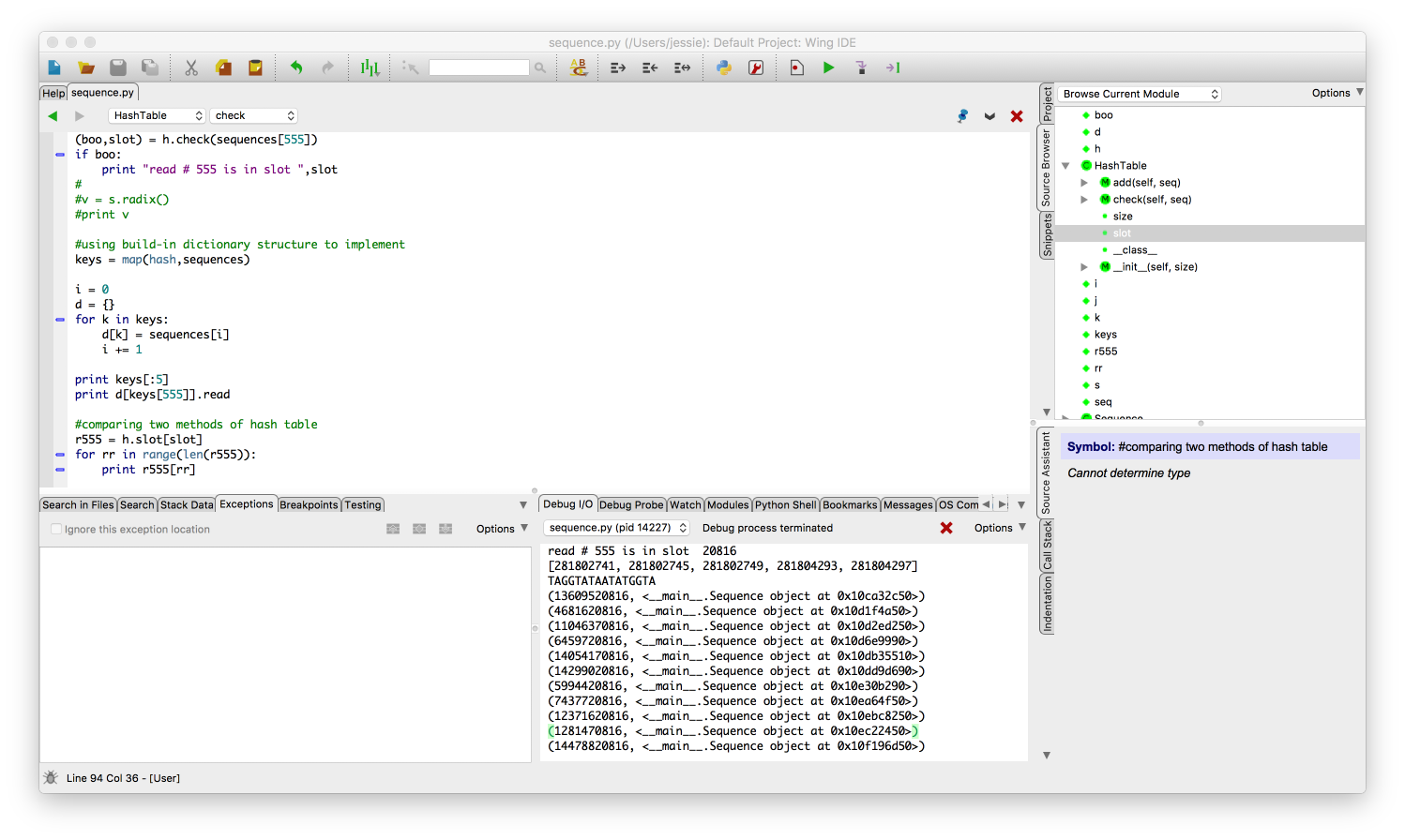
1B. see hw2.c:



1C. I

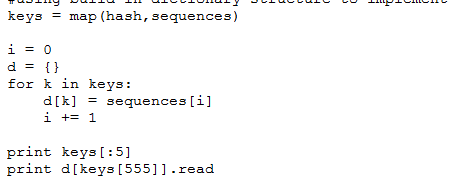
The first part of this python program is to implement the chaining and radix/division method as I’ve done with C.

The data-structure is exactly same as linked-list



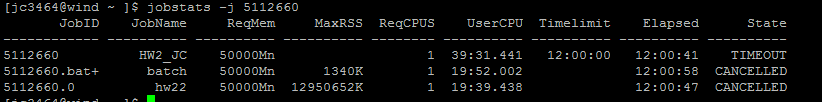
the print-out result shown above actually demonstrate a chain in the hash table. The nodes in this chain are composed with the radix value and a “pointer” to the stored sequence.

The second part is to use dictionary and build-in hash function to implement hash table:



Problem#2

1. See function “All” in hw22.c It works for 5 char long sequences. I tested the program with size of 16 on Monsoon. It ran for 12 hours and ….. timeout.



1. Using char array to implement bit array. The initial value here is NULL, because NULL’s ASCII value is 0. Every 8 bits is a “chunk”, in every chunk the exact index of each bit if “precise”. If we want to flip a certain bit, we’ll need to add 2^precise to the chunk char, in my program this value is shown as “temp”.
2. Below is the result of generating all sequences of size 4 and INSERT the 60th sequences. The chunk(or char) # 7 became “16” indicate that a sequence of radix value 60 is INSERT in the table.

