1. Implement direct addressing in a table.
2. Implement hashing with chaining.
3. Implement hashing by probing using different schemes. Take care of deleted slots and differentiate them from slots never occupied.
4. Write a code for implementing a stack, a queue, and a queue simulated by two stacks and a stack simulated by two queues, all using prespecified size arrays. You need to implement the push, pop, enqueue, dequeue operations and error checks.
   1. Write a routine for changing a sequence of elements in a queue according to a specified input permutation using two temporary queues. Do the same using a single temporary queue.
   2. Write a routine for changing a sequence of elements in a stack according to a specified permutation using two temporary stacks.
5. Implement a list data structure which behaves as a queue or a stack

(conceptually meaning FIFO or LIFO) depending on the size of the

data structure. For concreteness implement it such that deletion hap-

pens according to the logic of a queue if the size of the data structure

at the time of deletion is <= 8 and according to the logic of stack when

the size is > 8 at the time of deletion.

1. Write a routine for transferring the elements of a stack to a second

stack in the same order, using a third stack as working space. The

extra restriction is that the relative order of no two elements can be

reverse of their original on any stack at any stage of the procedure.