

## Data Structures

1. Which of the following data structures is most appropriate for storing the list of *return addresses* for a chain of **recursive function calls**?  
(a) array      (b) linked list      (c) stack      (d) queue
2. Which of the following data structures is most appropriate for storing **how many days are in each month** from January through December?  
(a) array      (b) linked list      (c) stack      (d) queue
3. Which of the following data structures is most appropriate to represent a **teller line in a bank**?  
(a) array      (b) linked list      (c) stack      (d) queue
4. Which of the following data structures is most appropriate for a sequence that **efficiently** maintains an **ascending order** when **adding a new element**?  
(a) array      (b) linked list      (c) queue      (d) priority queue
5. Which of the following data structures is most appropriate for a list of currently running processes maintained by an operating system? When allocating CPU cycles to the processes, the operating system uses "urgency" criteria to select the next process.  
(a) queue      (b) linked list      (c) stack      (d) priority queue
6. Which of the following data structures allows **direct access** to an element at any position?  
(a) array      (b) linked list      (c) stack      (d) queue
7. Which of the following data structures is easy to use for **inserting or deleting** any element in the container?  
(a) array      (b) linked list      (c) stack      (d) queue
8. Which of the following data structures can be accessed at **only one end** of the list?  
(a) array      (b) linked list      (c) stack      (d) queue
9. The order of accessing a **stack** is indicated by  
(a) First-In-First-Out      (b) Last-In-First-Out      (c) Last-In-Last-Out
10. The order of accessing a **queue** is indicated by  
(a) First-In-First-Out      (b) Last-In-First-Out      (c) First-In-Last-Out