

# data-structure-ds-set-4 (mcqmate.com).pdf

Author:

McqMate.com

Data Structure (DS) MCQs [set-4]

76. A circular linked list can be used for

- A. stack
- B. queue
- C. both stack & queue
- D. neither stack or queue

Answer: C

77. In doubly linked lists

- A. a pointer is maintained to store both next and previous nodes.
- B. two pointers are maintained to store next and previous nodes.
- C. a pointer to self is maintained for each node.
- D. none of the above

Answer: B

78. The disadvantage in using a circular linked list is .....

- A. it is possible to get into infinite loop
- B. last node points to first node.
- C. time consuming
- D. requires more memory space

Answer: A

79. A linear list in which each node has pointers to point to the predecessor and successors nodes is called as

- A. singly linked list
- B. circular linked list
- C. doubly linked list
- D. linear linked list

Answer: C

80. The situation when in a linked list START=NULL is

- A. underflow
- B. overflow
- C. housefull
- D. saturated

Answer: A

81. In doubly linked lists, traversal can be performed?

- A. only in forward direction
- B. only in reverse direction
- C. in both directions
- D. none of the above

Answer: C

82. What differentiates a circular linked list from a normal linked list?

- A. you cannot have the 'next' pointer point to null in a circular linked list
- B. it is faster to traverse the circular linked list
- C. you may or may not have the 'next' pointer point to null in a circular linked list
- D. head node is known in circular linked list

Answer: C

83. How do you count the number of elements in the circular linked list?

- A. `public int length(node head) { int length = 0; if( head == null) return 0; node temp = head.getnext(); while(temp != head) { temp = temp.getnext(); length++; } return length; }`
- B. `public int length(node head) { int length = 0; if( head == null) return 0; node temp = head.getnext(); while(temp != null) { temp = temp.getnext(); length++; } return length; }`
- C. `public int length(node head) { int length = 0; if( head == null) return 0; node temp = head.getnext(); while(temp != head && temp != null) { temp = head.getnext(); length++; } return length; }`
- D. `public int length(node head) { int length = 0; if( head == null) return 0; node temp = head.getnext(); while(temp != head && temp == null) { temp = head.getnext(); length++; } return length; }`

Answer: A

84. `public int function()`

```
{
if(head == null)
return Integer.MIN_VALUE;
int var;
Node temp = head;
```

View all MCQ's at [McqMate.com](http://McqMate.com)