Q1. Using an array of size n to implement a queue Q (circular queue). Which of the following Enqueue x to the queue (Assuming the queue is not empty)?

a. Q->front = (Q->front+1)%n; Q->data[Q->front]=x;
b. Q->data[Q->rear]=x; Q->rear = (Q->rear+1)%n;
c. Q->rear = (Q->rear+1)%n; Q->data[Q->rear]=x;
d. Q->data[Q->front]=x, Q->front = (Q->front+1) %n;

Q2. The minimum number of fields in each node of a Circular linked list is:

a. 4 fields
b. 2 fields
c. 1 field
d. 3 fields

Q3. Let: pointer h point to node x in a linked list (the head of the linked list), *h->next* point to the node y, pointer p point to new node z. Please choose the right code to insert node z between node x and node y.

a. p->next = h; h=p->next;
b. h->next=p; p->next=h->next;
c. p->next = h; h=p;
d. p->next=h->next; h->next=p;

Q4. Given a sequential stack S, push *o p q r s t u* into S one by one. If the pop sequence is: *p t s u r q o,* then the most numbers of elements inside the stack at the same time is:

a. 3
b. 4
c. 5
d. 2

Which of the following scenarios is best suited for a stack?

a. Packets forwarding in routers and switches
b. Managing function calls in recursion
c. CPU Scheduling
d. Serving requests on a shared resource (like a printer queue)

Q6. What does time complexity of an algorithm represent?

a. The actual time taken by the algorithm to execute
b. The number of instructions executed by the algorithm
c. The amount of memory used by the algorithm
d. The growth rate of the algorithm's running time with respect to the input size

Q7. Given the following function:

```
void funA(Node * head){
  while(head)
    head=head->next.
```

What this function will do to the linked list if it takes the head pointer of the linked list as input?

a. Insert new element into the linked list
b. Delete the whole linked list
c. Delete one element from the linked list
d. No Thing

Q8. Which of the following statements is true?

a. O(n log n) is always faster than $O(n^2)$ for large values of n
b. $O(n^2)$ is always faster than O(n log n) for large values of n
c. O(n log n) is always faster than O(n) for large values of n
d. O(n) is always faster than O(log n) for large values of n

Q9. Given the Dequeue sequence *Z Y X W V U*, which of the following was a valid Enqueue sequence:

a. U V W X Y Z
b. Z Y X W V U
c. U V W Z X Y
d. Z Y V U W X

Q10. Which of the following is NOT true about arrays in C?

a. Arrays can be multi-dimensional.
b. Arrays can have elements of different data types.
c. Arrays have a fixed size determined at compile time.
d. Elements in an array are stored sequentially in memory.

Q11. Consider a sequence of push and pop operations used to push the characters, *L M N O P Q R S T U* on a stack. The characters will be pushed in order; however, the pop operations can be interleaved with the push operations, and can occur any time there is at least one item on the stack. When an item is popped, it is printed to the terminal. Which of the following could NOT be the output from such a sequence of operations?

a. Q P O N R S T U M L
b. N O Q P R M L U T S
c. N O Q P R S T U L M
d. L M N O P Q R S T U

Q12. Given the following function:

```
function matrixSum(matrix) {
    let sum = 0;
    for(i = 0; i < row; i++) {
        for(j = 1; j < column; j=j*2)
            sum += matrix [i][j];
        }
        return sum;
    }
```

Assuming the input matrix of size n * m, the big O of function *matrixSum* is:

a. O(n * m)
b. O(m log (n))
c. O(n log (m))
d. $O(n^2)$

Q13. What is the time complexity for accessing an element *in* an array?

a. O(log n)
b. O(1)
c. O(n)
d. $O(n^2)$

Q14. Given the following lines of code:

```
char s1[20]="Welcome"; char s2[20]="HI"; char s3[20]="AIT206";
printf(" %s", strcat(s2, s1));
printf(", %s ", strcpy(s1, s3));
```

The output of this code is:

a. HIWelcome , AIT206
b. HI, AIT206WelcomeHI
c. WelcomeHI , AIT206
d. HIWelcome , WelcomeAIT206

Q15. What is the purpose of the "head" pointer in a linked list?

a. It points to the middle node of the list.
b. It points to the last node of the list.
c. It points to the first node of the list.
d. It points to a random node in the list.

Q16. In the optimal implementation of the stack using singly linked list, what happens when you push a new node onto a stack?

a. No Changes happens
b. The new node is placed at the middle of the linked list
c. The new node is placed at the head of the linked list
d. The new node is placed at the tail of the linked list

Q17. What is the time complexity for inserting a node at the end of a linked list?

a. $O(n^2)$
b. $O(1)$
c. $O(n)$
d. $O(\log n)$

Q18. Which of the following is true about a stack implemented using a linked list?

a. It cannot be implemented using a linked list.
b. It can dynamically grow and shrink in size.
c. It has a fixed size determined at compile time.
d. None of the above

Q19. What is the time complexity to delete an element from the middle of a single linked list, given a pointer *p* points to that element?

a. O(log n)
b. O(1)
c. $O(n^2)$
d. O(n)

Q20. Assume $f(N) = 10N^4 + 20\,N^{0.5} + N^3\log(N) + 5N^2 + 10$, then g(N) is might be:

a. N^2
b. N^4
c. $N^{0.5}$
d. $N^3\log(N)$