1. 5% In a lower triangular matrix, $A$, with $n$ rows, the maximum number of non-zero terms in row $i$ is $i$. Obtain an addressing formula for elements $a_{ij}$ in the lower triangle if this lower triangle is stored by rows in an array $B(1:n(n+1)/2)$ with $A(1,1)$ being stored in $B(1)$.

2. 5% In a tridiagonal matrix, all elements other than those on the major diagonal and on the diagonals immediately above and below this one are zero. If the elements in the band formed by these three diagonals are represented rowwise in an array, $B$, with $A(1,1)$ being stored at $B(1)$, obtain an algorithm to determine the value of $A(i,j)$, $1 \leq i,j \leq n$ from the array $B$.

3. 20% Assume there are $n$ data numbered from 1 to $n$ to be put in the stack sequentially and removed at any time. For example, if $n = 3$, we could move 1 in, move 2 in, move 3 in and then take the data out producing the new order 3, 2, 1. For $n = 3$ and 4 what are the possible permutations of the data that can be obtained? Are any permutations not possible?

4. 10% What is the postfix form of the infix form $A / B + C * D + E - A * C$? Assume the priorities of $+, -$ are greater than those of $*, /$ and the associations of all operators are right association. Then again derive the postfix form of the above infix form.

5. 10% A node in a doubly linked list has at least 3 fields, say DATA, LLINK (left link) and RLINK (right link). Write an algorithm to insert node $P$ to the right of node $X$. Also write an algorithm to delete node $X$ from a doubly linked list $L$.

6. 20% State the algorithm of QUICKSORT to sort $n$ data into ascending order. Derive the best case, worst case time complexities of it.

7. 10% State the meaning of the Huffman codes. Construct the Huffman codes for the following messages whose access frequencies are 2, 3, 5, 8, 13, 15 and 18.

8. 20% A hashing scheme performs an identifier transformation through the use of a hash function $f$. It is desirable to choose a function $f$ which is easily computed and also minimizes the number of collisions. Since the size of the identifier space, $T$ is usually several orders of magnitude larger than the number of buckets $k$ (assume each bucket can only hold an identifier), overflows will occur. State as many mechanisms as possible to handle the overflows of hashing schemes.