BIRLA INSTITUTE OF TECHNOLOGY, MESRA, RANCHI
(MID SEMESTER EXAMINATION)

CLASS: BTECH/IMSC **SEMESTER : III/V**
BRANCH: CSE & AIML/PHYSICS **SESSION : MO/2025**

SUBJECT: CS24201 DATA STRUCTURE AND ALGORITHMS

TIME: 02 Hours **FULL MARKS: 25**

INSTRUCTIONS:
1. The question paper contains 5 questions each of 5 marks and total 25 marks.
2. Attempt all questions.
3. The missing data, if any, may be assumed suitably.
4. For answering algorithm/pseudocode, writing language specific syntax will be penalized.
5. While writing algorithm/pseudocode, you can only use the data structures covered so far, i.e., array, linked list, and stack. Use of any other data structure or undefined built-in functions, e.g., length(), maxsize(), sort(), etc., will be penalized.

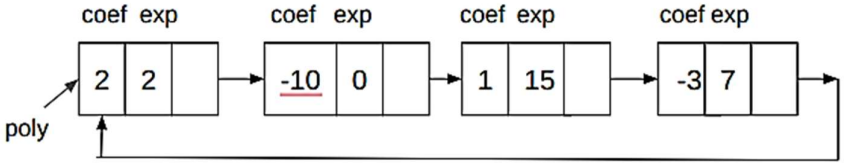
READ THE ABOVE INSTRUCTIONS CAREFULLY

			CO	BL
Q.1(a)	Given the following program and recurrence relation, check if the relation correctly represents the total number of instructions in terms of T(n), where n is the input size. If the relation is incorrect, write the correct relation and solve the recurrence. foo(n): 1. if n<= 1 then 2. return 3. for i from 1 to 3 do 4. foo(n-1) $T(n) = \begin{cases} 1 & \text{if } n \leq 1, \\ T(3n-1)+c & \text{if } n > 1 \end{cases}$	[2]	1	2,3
Q.1(b)	Find the relation between f(n) and k(n) when the following are given. Justify your answer. $f(n) = \Theta(g(n))$ $g(n) = O(h(n))$ $h(n) = \Theta(k(n))$	[3]	1	2, 4
Q.2(a)	Consider the given sparse matrix. Represent the compressed form of the matrix in triplet format. Calculate the memory space saved when compared to storing the full matrix in a 2-D array. Consider each element occupies 4 bytes. $\begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \end{bmatrix}$	[2]	2	2, 5
Q.2(b)	A nxn matrix A[][] is defined to be of bandwidth 1, if it has non zero elements in the main diagonal and the 2 diagonals below and above it, while the remaining elements are zero. A matrix of size 4x4 and bandwidth 1 is given below for illustration. The elements in the band of the nxn matrix are to be saved in a 1-dimensional arrayin row major form, say M[1], .. M[r], where M[1] = A_{11}, M[2] = A_{12}, M[3] = A_{21}, and so on. $\begin{bmatrix} A_{11} & A_{12} & 0 & 0 \\ A_{21} & A_{22} & A_{23} & 0 \\ 0 & A_{32} & A_{33} & A_{34} \\ 0 & 0 & A_{43} & A_{44} \end{bmatrix}$ i) Find the maximum number of nonzero elements in A as a function of n. ii) Given an element A_{ij}, $1 \leq i, j \leq n$, find the index in M[] where it will be placed. Justify your answers.	[3]	2	2,3, 5

PTO

Q.3(a) Write an algorithm/pseudocode for reversing a singly linked list without using any auxiliary space, e.g., an additional list, array, or other data structures. [2] 2 2, 6

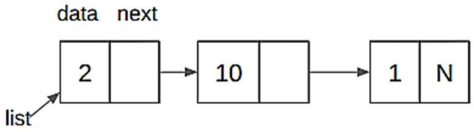
Q.3(b) It has been decided to use a circular linked list representation for a sparse polynomial p(x). For $p(x) = 2x^2 - 10 + x^{15} - 3x^7$, of degree 15 and 4 nonzero terms, the representation is shown below. [3] 2 4,5



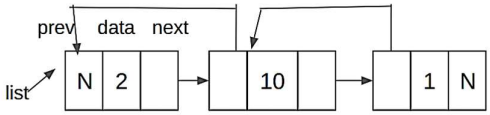
Given a polynomial named, poly, write pseudo code to find its degree and its number of terms. If poly is NULL, its degree and terms are defined as -1 and 0 respectively.

Q.4(a) Given a single linked list L as shown below, write a program to convert it to a doubly linked list. [2] 2 2,3,6

If the given list is as follows:



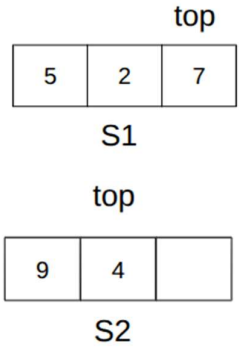
then following is the expected output:



Q.4(b) A theme park maintains a circular list of visitors waiting for a ride. The management wants to divide the visitors into two groups so that both groups have equal or nearly equal numbers of visitors. This allows the visitors to board two separate rides simultaneously. Write an algorithm to split the circular linked list of visitors into two halves in a single traversal, while maintaining the circular structure for each group. [3] 2 1,4,6

Q.5(a) Consider the following pseudo code involving 2 stacks given below, where the stack operations have their usual meaning. [2] 3 1,2,3

```
Stack S1, S2;
while ( isempty(S2) is false)
     {S1.push(S2.pop())}
// Point p1
while ( ( isempty(S1) is false)
     {S2.push(S1.pop()); S1.pop(); }
// Point p2
```



If the contents of S1 and S2 at the start were as shown in the right side, show the contents of the 2 stacks at the marked points p1 and p2.

Q.5(b) Convert the following infix expression to prefix using the reverse stack approach: **((A + B) * C) ^ (D - E + F) / G** [3] 3 1,3,5

In this approach, the given infix expression is first reversed, then converted into postfix by scanning each symbol from left to right (applying precedence and

associativity rules appropriately), and finally the obtained postfix expression is reversed again to get the prefix form.

For each symbol scanned in the expression, show the contents of the stack and the resulting output. For example, while converting the expression *A op1 B op2 C* following columns should be maintained: Symbol, Stack Content, and Output (where the rightmost entry in the Stack column indicates the current top).

Symbol	Stack Content	Output
A	**empty**	**A**
op1	**op1**	**A**
B	**op1**	**AB**
op2	**op1 op2**	**AB**

:::::18/09/2025 :::::E