Course Code	**:**	**MCS-021**
Course Title	**:**	**Data and File Structures**
Assignment Number	**:**	**BCA(III)/021/Assignment/2026-27**
Maximum Marks	**:**	**100**
Weightage	**:**	**30%**
Last Dates for Submission	**:**	**31st October, 2026 (For July 2026 Session)**
	:	**30th April, 2027 (For January 2027 Session)**

This assignment has 13 questions with a total of 80 Marks; answer all questions. The remaining 20 marks are for viva voce. Please go through the guidelines regarding assignments given in the Programme Guide for the format of presentation.

Q1: **(6 Marks)**

Write a program in C to accept two polynomials as input and print the resultant polynomial due to the multiplication of the input polynomials.

Q2: **(6 Marks)**

Write a program in 'C' to create a doubly linked list to store integers and perform the following operations on it:

(i) Find the sum of all even integers and odd integers separately and print them.
(ii) Traverse the linked list and store all the even integers in the beginning nodes and all odd integers in the remaining nodes in a separate linked list.

Q3: **(6 Marks)**

Write a program in 'C' to input two matrices and perform the following operations on them.

(i) Add both the matrices and display the resultant matrix.
(ii) Find the sum of the main diagonal elements of the resultant matrix, computed in (i)

Q4: **(6 Marks)**

Use the appropriate data structures and write a program in C to evaluate the following expressions:

Prefix expression: +, -, *, 3, 2, /, 8, 4, 1
Postfix expression: 10, 5, +, 60, 6, /, *, 8, -

Q5: **(6 Marks)**

Draw the binary tree for which the traversal sequences are given as follows:

(i) Post order: J K I H D F G E C B A
In order: H J I K A D C F E G B

(ii) Pre-order: A B D E H I C F G J K
In order: D B H E I A F C J G K

Q6: **(6 Marks)**

Define an AVL tree. Create an AVL tree for the following list of data if the data are inserted in the order in an empty AVL tree.

3, 10, 15, 5, 25, 12, 2, 30, 22, 24, 29, 8, 40, 14, 11

Further delete 3, 22, 8, and 11 from the above AVL tree.

Q7: **(6 Marks)**

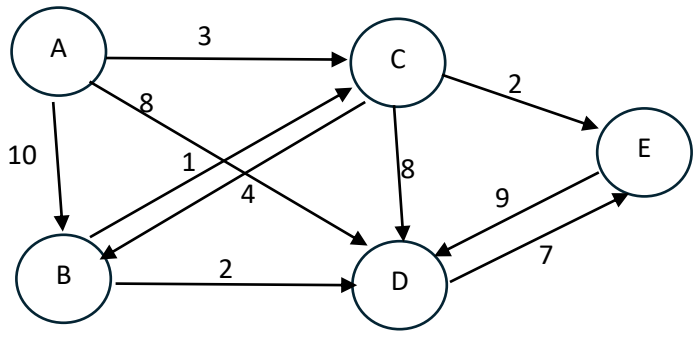
Define a B-tree and its properties. Create a B-tree of order-5, if the data items are inserted into an empty B-tree in the following sequence:

10, 2, 15, 25, 60, 45, 35, 80, 20, 8, 4, 55, 50, 68, 62, 40

Further, delete items 2, 4, 8, and 10 from the B-tree.

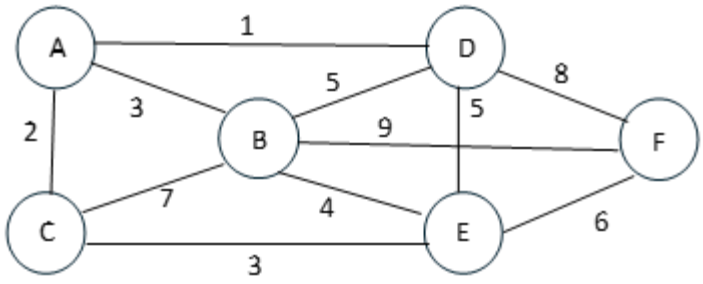
Q8: **(7 Marks)**

Apply Bellman-Ford's algorithm to find the shortest path from the vertex 'S' to all other vertices for the following graph:



Q9: **(7 Marks)**

What is a minimum spanning tree (MST)? Apply Prim's algorithm to find the MST for the following graph.



Q10: **(6 Marks)**

Apply Insertion and Bubble sorting algorithms to sort the following list of items. Show all the intermediate steps. Also, analyze the number of swap and comparison operations performed in both methods.

2, 12, 14, 16, 25, 30, 45, 555, 42, 50, 9, 6, 1

Q11: **(6 Marks)**

What is a heap tree? Create a max heap tree for the following list of items inserted in the order. Also, explain the heap sort with the help of the heap tree.

10, 20, 5, 25, 30, 18, 3, 70, 55, 45, 12, 24

Q12: **(6 Marks)**

What is a Red-Black tree? Explain insertion and deletion operations in a Red-Black tree with the help of a suitable example.

Q13: **(6 Marks)**

Explain Sequential, Direct, and Indexed File Organization.

Q1: (6 Marks) Write a program in C to accept two polynomials as input and print the resultant polynomial due to the multiplication of the input polynomials.

Answer

A polynomial can be stored in an array in which the index denotes the exponent and the value at that index is the coefficient. If the coefficient of x^i in the first polynomial is a[i] and that of x^j in the second is b[j], then their contribution to the product is a[i] x b[j] at exponent i + j. Therefore the product coefficient array is obtained by two nested loops.

C program

```
#include <stdio.h>
#define MAX 20

void printPoly(int p[], int d) {
    int first = 1;
    for (int i = d; i >= 0; --i) {
        if (p[i] == 0) continue;
        if (!first && p[i] > 0) printf(" + ");
        if (p[i] < 0) printf(first ? "-" : " - ");
        int c = p[i] < 0 ? -p[i] : p[i];
        if (i == 0 || c != 1) printf("%d", c);
        if (i >= 1) printf("x");
        if (i >= 2) printf("^%d", i);
        first = 0;
    }
    if (first) printf("0");
    printf("\n");
}

int main(void) {
    int d1, d2, a[MAX] = {0}, b[MAX] = {0}, c[2*MAX] = {0};
    printf("Degree of first polynomial: ");
    scanf("%d", &d1);
    printf("Enter coefficients a0..a%d: ", d1);
    for (int i=0; i<=d1; ++i) scanf("%d", &a[i]);

    printf("Degree of second polynomial: ");
    scanf("%d", &d2);
    printf("Enter coefficients b0..b%d: ", d2);
    for (int i=0; i<=d2; ++i) scanf("%d", &b[i]);

    for (int i=0; i<=d1; ++i)
        for (int j=0; j<=d2; ++j)
            c[i+j] += a[i] * b[j];

    printf("Product polynomial: ");
    printPoly(c, d1+d2);
    return 0;
}
```