**BITS Pilani, Dubai Campus**
Dubai International Academic City, Dubai
**I Semester 2023-24**

**CS F222 Discrete Structures for Computer Science**
**Assignment**

**Due date: 12-Dec-2023**

---

**Instructions:**

- Select the problem assigned to you as follows:

  *The Problem assigned to you = [Integer reminder of Last 3 numerals of your IdNo divided by 120] + 1.*

- You can use any programming language of your choice to solve the question.
- You need to submit a two page word document of your assignment (1.15 line spacing, 11 point Arial font, all margins set to 0.7 inch, one blank line between paragraphs) stating how your program, solution can be used to solve a real life problem. Also you should make a short video (duration not more than 2 minutes) explaining your problem and upload the file.
- The document must contain your ID No., Name, Section No., Your assignment question number and question statement, where the solution can be used.
- Also you need to upload the following documents in the LMS.
  (i) Source code file. File name should be aqxx-your ID No.EXT (aq indicates assignment question, xx is the question number strictly in two digits. EXT is the extension as per the programming language.
  (ii) Input file. File name should be aqxxip-your ID No
  (iii) Output file. File name should be aqxxop-your ID No

---

In problems 1-6, assume that a set $X$ of $n$ elements is represented as an array $A$ of size at least $n+1$. The elements of $X$ are listed consecutively in $A$ starting in the first position and terminating with 0. Assume further that no set contains 0.

1. Write a program to represent the sets $X \cup Y, X \cap Y, X - Y$ and $X \Delta Y$, given the arrays representing $X$ and $Y$. $\Delta$ denotes the symmetric difference.

2. Write a program to determine whether $X \subseteq Y$, given arrays representing $X$ and $Y$.

3. Write a program to determine whether $X = Y$, given arrays representing $X$ and $Y$.

4. Assuming the universe represented as an array, write a program to represent the set $\overline{X}$, given the array representing $X$.

5. Given an element $x$ and the array $A$ that represents $X$, write a program that determines whether $x \in X$.

6. Given the array representing $X$, write a program that lists all subsets of $X$.

   In problems 7-10, assume that a sequence from $\{1, 2, \cdots, n\}$ to the real numbers is represented as an array $A$, indexed from 1 to $n$.

7. Write a program that tests whether $A$ is one-to-one.

8. Write a program that tests whether $A$ is onto a given set.

9. Write a program that tests whether $A$ is increasing.

10. Write a program that tests whether $A$ is decreasing.

11. Write a program to determine whether one sequence is a subsequence of another sequence.

12. Write a program that finds the domain of a relation.

13. Write a program that finds the range of a relation.

14. Write a program to determine whether a relation is reflexive.

15. Write a program to determine whether a relation is antisymmetric.

16. Write a program to determine whether a relation is transitive.

17. Write a program that finds the inverse of a relation.

18. Write a program that finds the composition $R \circ S$ of the relations $R$ and $S$.

19. Write a program that checks whether a relation $R$ is an equivalence relation. If $R$ is an equivalence relation, the program outputs the equivalence classes of $R$.

20. Write a program to determine whether a relation is a function from a set $X$ to a set $Y$.

21. Write a program that generates all $r$-combinations of the elements $\{1, \cdots, n\}$.

22. Write a program that generates all permutations of the elements $\{1, \cdots, n\}$.

23. Write a program that generates all $r$-permutations of the elements $\{1, \cdots, n\}$.

24. Write a program that lists all permutations of $ABCDEF$ in which $A$ appears before $D$.

25. Write a program that lists all permutations of $ABCDEF$ in which $C$ and $E$ are side by side in either order.

26. Write a program that prints the amount accumulated yearly if a person invests $n$ rupees at $p$ percent compounded annually.

27. Write a program that prints the amount accumulated yearly if a person invests $n$ rupees at $p$ percent annual interest compounded $m$ times yearly.

28. Write a program that solves the three peg Tower of Hanoi puzzle.

29. Write a program that solves the four peg Tower of Hanoi puzzle in fewer moves that does the solution to the three peg puzzle.

30. Write a program that prints all $n$-bit strings that do not contain the pattern 010.

31. Given a partial order $R$ on a finite set $S$, determine a total order on $S$ that contains $R$. Assume that the elements of the sets $S$ and $R$ are listed.

32. Given sets $X = \{x_1, x_2, \cdots, x_m,\}$ and $Y = \{y_1, y_2, \cdots, y_n,\}$. List all the functions with domain $X$ and codomain $Y$.

33. Given sets $X = \{x_1, x_2, \cdots, x_m,\}$ and $Y = \{y_1, y_2, \cdots, y_n,\}$. List all the one-to-one functions with domain $X$ and codomain $Y$.

34. Given sets $X = \{x_1, x_2, \cdots, x_m,\}$ and $Y = \{y_1, y_2, \cdots, y_n,\}$. List all the onto functions with domain $X$ and codomain $Y$.

35. Given positive integers $k$ and $n$, list all the nonnegative integer solutions to the equation $x_1 + x_2 + \cdots + x_k = n$.

36. Given a positive integer $n$, list all the derangements of the integers $1, 2, \cdots, n$.

37. Given a positive integer $n$, list all the permutations of $1, 2, \cdots, n$ in lexicographic order.

38. Given a positive integers $r$ and $n$, list all the $r$-element subsets of $\{1, 2, \cdots, n\}$ in lexicographic order.

39. Given a positive integers $r$ and $n$, list all the $r$-permutations of $1, 2, \cdots, n$ in lexicographic order.

40. Given a positive integers $r$ and $n$, print all possible ordered list of $r$ items selected from the set $\{1, 2, \cdots, n\}$ if repetition of items is allowed.

41. Given a positive integers $r$ and $n$, print all possible unordered list of $r$ items selected from the set $\{1, 2, \cdots, n\}$ if repetition of items is allowed.

42. Given a recurrence relation $s_n = a_1 s_{n-1} + a_2 s_{n-2} + \cdots a_k s_{n-k}$ and initial values $s_0, s_1, \cdots, s_{k-1}$, compute a specified term of the sequence defined by these conditions.

43. Given a positive integer $n$, list the moves necessary to win the Towers of Hanoi game with $n$ disks in the fewest possible moves.

44. Given $f(x) = a_0 + a_1 x + \cdots a_n x^n$ and $g(x) = b_0 + b_1 x + \cdots b_n x^n$ and a nonnegative integer $k$, compute the coefficients of $x^k$ in the polynomial $f(x)g(x)$.

45. Consider the sequence recursively defined by $g(0) = 1, g(1) = -1, g(n) = 3g(n-1) - 2g(n-2)$. Write a program to print the first 20 terms of the sequence.

46. Consider the sequence recursively defined by $g(0) = 1, g(1) = -1, g(n) = 3g(n-1) - 2g(n-2)$. Write a program to print the first $n$ terms of the sequence. The user should be able to supply the value of $n$ at runtime.

47. Let $R$ and $S$ be relations represented by matrices $M_R$ and $M_S$ respectively. Write a program to produce matrix of $R \cup S, R \cap S, R \circ S$.

48. Let $R$ be a relation represented by the matrix $M_R$. Write a program to produce the matrix of $R^{-1}, \overline{R}$

49. Write a program that writes a given permutation as a product of disjoint cycles.

50. Write a program that writes a given permutation as a product of transpositions.

51. Write a program that determines whether a given permutation is even or odd.

52. Write a program that determines if a relation $R$ represented by its matrix is a partial order.

53. Let $H = \{[0], [2]\}$. Write a program that computes the left cosets of $H$ in $Z_6$.

54. Let $H = \{[0], [2], [4], [6]\}$. Write a program that computes the right cosets of $H$ in $Z_8$.

55. Write a program that given a finite operation table will determine if the operation satisfies the associative property.

56. Write a program that given a finite group $G$ and a subgroup $H$ determines if $H$ is a normal subgroup of $G$.

57. Given a real number $a$ and a nonnegative integer $n$, write a program to find $a^n$ using recursion.

58. Given a real number $a$ and a nonnegative integer $n$, write a program to find $a^{2^n}$ using recursion.

59. Given a positive integer $n$, write a program list all the bit sequences of length $n$ that do not have a pair of consecutive 0s.

60. Given a recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ where $c_1$ and $c_2$ are real numbers, initial conditions $a_0 = C_0$ and $a_1 = C_1$, and a positive integer $k$, write a program to find $a_k$ using iteration.

61. Write a program that accepts as input any of the following:

    - a listing of edges of a graph given as pairs of positive integers
    - the adjacency matrix
    - the incidence matrix

    and outputs the other two.

62. Write a program to check whether the given graph is a bipartite graph. If it is a bipartite graph, the program should list the disjoint sets of vertices.

63. Write a program to check whether the given graph is a multigraph.

64. Write a program to check whether the given graph is a digraph.

65. Write a program to check whether the given graph is a complete bipartite graph. If yes, the program should list the disjoint sets of vertices.

66. Write a program that generates an $n \times n$ adjacency matrix and print the adjacency matrix, the number of edges, the number of loops.

67. Write a program which prints the degree of each vertex in an undirected graph.

68. Write a program which prints the degree of each vertex in a directed graph.

69. Write a program that accepts as input the edges of a graph and then draw the graph using a computer graphics display.

70. Write a program that lists all simple paths between two given vertices.

71. Write a program that determines whether a path is a simple path, a circuit or a cycle.

72. Write a program that tests whether a proposed isomorphism is an isomorphism.

73. Write a program that determines whether a graph is connected.

74. Write a program that tests whether the given graph which is represented by adjacency matrix is a tree.

75. Write a program that finds the components of a graph.

76. Given the adjacency matrix of a graph, write a program to find its adjacency list.

77. Given the adjacency list of a graph, write a program to find its adjacency matrix.

78. Given the adjacency matrix of a graph with vertices $v_1, v_2, \cdots, v_n$, write a program to find the number of paths of length $m$ from $v_i$ to $v_j$.

79. Write a program to determine whether a given graph is a complete graph.

80. Given a multigraph, write a program to determine whether there are any loops or parallel edges.

81. Given a graph and a positive integer $n$, write a program to determine whether the graph has a cycle of length $n$.

82. A graph is represented using adjacency list. Write a program which checks whether the given graph is bipartite. If yes, the program should list the disjoint sets of vertices.

83. Consider a graph which is represented using adjacency list. Write a program to check whether the given graph is a multigraph or not. If it a multigraph print all those pair of vertices having parallel edges.

84. Write a program which prints degree of each vertex of a given undirected graph, when the graph is represented by using adjacency list.

85. Write a program which prints the indegree and outdegree of each vertex of a given directed graph, when the graph is represented by using adjacency list.

86. Write a program that tests whether the given graph, which is represented by adjacency list is a tree.

87. Given a graph, write a program using the breadth-first algorithm to label its vertices.

88. Write a program that, given the adjacency matrix of a tree and a vertex $v$, draws the tree rooted at $v$ using a computer graphics display.

89. Write a program that, given a tree $T$, computes the eccentricity of each vertex in $T$ and finds the center(s) of $T$.

90. Write a program that, given a rooted tree and a vertex $v$, finds the parent of $v$, finds the ancestors of $v$, finds the children of $v$, finds the descendants of $v$, finds the siblings of $v$ and determines whether $v$ is a terminal vertex.

91. Write a program that finds a spanning tree in a graph.

92. Implement Prim's Algorithm as a program.

93. Implement Kruskal's Algorithm as a program.

94. Given a binary tree, write a program to give a preorder listing of vertices.

95. Given a binary tree, write a program to give a postorder listing of vertices.

96. Given a binary tree, write a program to give a inorder listing of vertices.

97. Given a rooted tree, write a program to find the internal vertices, the terminal vertices and the root.

98. Given a graph, write a program using the breadth-first search algorithm to find a spanning tree, if one exists.

99. Given a graph, write a program using the breadth-first search algorithm to determine whether it is connected.

100. Given a graph, write a program using the depth-first search algorithm to find a spanning tree, if one exists.

101. Given a graph, write a program using the depth-first search algorithm to determine whether it is connected.

102. Write a program which displays a sub-tree at a given vertex of a given rooted tree, when the graph is represented using adjacency matrix.

103. Write a program which displays a sub-tree at a given vertex of a given rooted tree, when the graph is represented using adjacency list.

104. Write a program which computes the height of a tree, when the tree is represented using adjacency matrix.

105. Write a program which computes the height of a tree, when the tree is represented using adjacency list.

106. Write a program which checks whether the given tree is a balanced tree, when the tree is represented using adjacency matrix.

107. Write a program which checks whether the given tree is a balanced tree, when the tree is represented using adjacency list.

108. Write a program which displays all the pendent vertices at a given level, when the tree is represented using adjacency matrix.

109. Write a program which displays all the pendent vertices at a given level, when the tree is represented using adjacency list.

110. Write a program which displays all the internal vertices at a given level, when the tree is represented using adjacency matrix.

111. Write a program which displays all the internal vertices at a given level, when the tree is represented using adjacency list.

112. Write a program which checks whether $T_2$ is a sub-tree of $T_1$, when both the trees are represented using adjacency matrix.

113. Write a program which checks whether $T_2$ is a sub-tree of $T_1$, when both the trees are represented using adjacency list.

114. Write a program which computes the number of regions in a planar graph, when the planar graph is represented using adjacency list.

115. Write a program which computes the number of regions in a planar graph, when the planar graph is represented using adjacency matrix.

116. Write a program that determines whether a graph contains an Euler cycle.

117. Write a program that finds an Euler cycle in a connected graph in which all vertices have even degree.

118. Write a program that determines whether a proposed cycle is a Hamiltonian cycle.

119. Write a program that determines whether a proposed path is a Hamiltonian path.

120. Write a program to determine whether a graph is planar.