8. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:

    i. INC
    ii. SPA
    iii. SNA
    iv. SZE

9. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:

    i. CIR
    ii. CIL

10. Write an assembly program that reads in integers and adds them together; until a negative non-zero number is read in. Then it outputs the sum (not including the last number).
11. Write an assembly program that reads in integers and adds them together; until zero is read in. Then it outputs the sum.

DSC 03: Mathematics for computing

Course Objective

This course introduces the students to the fundamental concepts and topics of linear algebra and vector calculus, whose knowledge is important in other computer science courses. The course aims to build the foundation for some of the core courses in later semesters.

Course Learning Outcomes

After successful completion of this course, the student will be able to:

1. Perform operations on matrices and sparse matrices
2. Compute the determinant, rank and eigenvalues of a matrix

3. Perform diagonalization
4. Perform operations on vectors, the dot product and cross product
5. Represent vectors geometrically and calculate the gradient, divergence, curl
6. Apply linear algebra and vector calculus to solve problems in sub-disciplines of computer science.

Syllabus

Unit 1 Introduction to Matrix Algebra: Echelon form of a Matrix, Rank of a Matrix, Determinant and Inverse of a matrix, Solution of System of Homogeneous & Non-Homogeneous Equations: Gauss elimination and Solution of System of Homogeneous Equations: Gauss Jordan Method.

Unit 2 Vector Space and Linear Transformation: Vector Space, Sub-spaces, Linear Combinations, Linear Span, Convex Sets, Linear Independence/Dependence, Basis & Dimension, Linear transformation on finite dimensional vector spaces, Inner Product Space, Schwarz Inequality, Orthonormal Basis, Gram-Schmidt Orthogonalization Process.

Unit 3 EigenValue and EigenVector: Characteristic Polynomial, Cayley Hamilton Theorem, Eigen Value and Eigen Vector of a matrix, Eigenspaces, Diagonalization, Positive Definite Matrices, Applications to Markov Matrices

Unit 4 Vector Calculus: Vector Algebra,Laws of Vector Algebra, Dot Product, Cross Product,Vector and Scalar Fields, Ordinary Derivative of Vectors, Space Curves, Partial Derivatives, Del Operator, Gradient of a Scalar Field, Directional Derivative, Gradient of Matrices, Divergence of a Vector Field, Laplacian Operator, Curl of a Vector Field.

References

1.Strang Gilbert. *Introduction to Linear Algebra*, 5th Edition, Wellesley-Cambridge Press, 2021.

2.Kreyszig Erwin. *Advanced Engineering Mathematics*, 10th Edition, Wiley, 2015.

3. Strang Gilbert. *Linear Algebra and Learning from Data*, 1st Edition, Wellesley-CambridgePress,2019.

4.Jain R. K., Iyengar S.R. K. *Advanced Engineering Mathematics*, 5^{th} Edition, Narosa, 2016.

Additional References

(i) Deisenroth, Marc Peter, Faisal A. Aldo and Ong ChengSoon. *Mathematics for Machine Learning*, 1^{st} Edition, Cambridge University Press, 2020.

(ii) Lipschutz Seymour and Lipson Marc. *Schaum's Outline of Linear Algebra*, 6^{th} Edition, McGraw Hill, 2017.

Suggested Practical List

1. Create and transform vectors and matrices (the transpose vector (matrix) conjugate transpose of a vector (matrix))

2. Generate the matrix into echelon form and find its rank.

3. Find cofactors, determinant, adjoint and inverse of a matrix.

4. Solve a system of Homogeneous and non-homogeneous equations using Gauss elimination method.

1. Solve a system of Homogeneous equations using the Gauss Jordan method.

6. Generate basis of column space, null space, row space and left null space of a matrix space.

7.Check the linear dependence of vectors. Generate a linear combination of given vectors of Rn/ matrices of the same size and find the transition matrix of given matrix space.

8.Find the orthonormal basis of a given vector space using the Gram-Schmidt orthogonalization process.

9.Check the diagonalizable property of matrices and find the corresponding eigenvalue and

verify the Cayley- Hamilton theorem.

10.Application of Linear algebra: Coding and decoding of messages using nonsingular matrices.

eg code “Linear Algebra is fun” and then decode it.

11.Compute Gradient of a scalar field.

12.Compute Divergence of a vector field.

13.Compute Curl of a vector field.

DSC 04: Object Oriented Programming with C++

Course Objective

This course is designed to introduce programming concepts using C++ to students. The course aims to develop structured as well as object-oriented programming skills using C++ programming language. The course also aims to achieve competence amongst its students to develop correct and efficient C++ programs to solve problems spanning multiple domains.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Write simple programs using built-in data types of C++.
2. Implement arrays and user defined functions in C++.
3. Write programs using dynamic memory allocation, handling external files, interrupts and exceptions.
4. Solve problems spanning multiple domains using suitable programming constructs in C++.
5. Solve problems spanning multiple domains using the concepts of object oriented programming in C++.

Syllabus

Unit 1 Introduction to C++: Overview of Procedural and Object-Oriented Programming, Using main() function, Header Files, Compiling and Executing Simple Programs in C++.

Unit 2 Programming Fundamentals: Data types, Variables, Operators, Expressions, Arrays, Keywords, Decision making constructs, Iteration, Type Casting, Input-output statements, Functions, Command Line Arguments/Parameters

Unit 3 Object Oriented Programming: Concepts of Abstraction, Encapsulation. Creating Classes and objects, Modifiers and Access Control, Constructors, Destructors, Implementation of Inheritance and Polymorphism, Template functions and classes

Unit 4 Pointers and References: Static and dynamic memory allocation, Pointer and Reference Variables, Implementing Runtime polymorphism using pointers and references