Gujarat Technological University

Master of Computer Applications

Semester-I

Subject Name: Fundamentals of Programming (FOP)

Subject Code: 2610001

---

1. Write a C program for the following:

1. Define a structure to store the details about a book, having members like bood_id, title of book, subject of the book, publisher, edition and price. 10

2. Input a number which works as the size of the array. Create a dynamic array of the book structure. 10

3. Define the following functions: 40

    1. getData (&bookVariable) ;
       // Function which takes input of all the members of the book structure variable passed a parameter

    2. showData (&bookVariable) ;
       // Function which show the details stored in the member variables of book structure variable passed as parameter

    3. totalCost = calculateTotalCost (booksArray, arraySize) ;
       // Function which takes array of books and the size of the book array as parameter and returns the total of price of all books

    4. storeToFile (filename, bookArray, arraySize) ;
       // Function which stores the books from the book array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

2. Write a C program for the following:

1. Define a structure named VALUES which has the members as firstSize, firstArray, secondSize, secondArray 10

2. Create a variable of the above mentioned structure with initial values as 0 for numeric variables and null for array variables 10

3. Define the following functions: 40

    1. fetchFromFile (filename, &structureVariable) ;
       // Function which reads values from file and store them in first array of the structure variable. Stores the values in array as per the first size variable

    2. removeDuplicates (&structureVariable) ;
       // Function which removes the duplicate values from the first array and store the unique values in the second array of the structure variable. The second size variable then holds the count of the values in the second array
       // if first array has elements: 3 5 2 1 3 5 4 5 1 7 and first size is 10
       // The second array elements should be: 3 5 2 1 4 7 0 0 0 0 and the second size should be 6

    3. showValues (&structureVariable) ;
       // Function which displays the values from the first array and second array in two separate lines on the screen

    4. storeToFile (&structureVariable, filename) ;
       // Function which stores the unique values from the structure variable to the file specified as the file name parameter

5. Write the main function to test above mentioned functions using a menu. 20

3. Write a C program for the following:

1. Define a structure that stores basic detail about the Person like the first name, middle name, last name, address locality and basic salary 10

2. Create a variable top store the size of an array and take input for the same from the user. Creates an array of person structure as per the size specified by user 10

3. Define the following functions: 40

    1. getData (&structureArray, arraySize) ;
       // Function which takes input of structure members for and store them in the array indices

    2. showData (&structureArray, arraySize) ;
       // Function which takes displays the values of structure members from the array indices

    3. count = getCount (&structureArray, arraySize, operationFlag) ;
       // Function that counts the total number of persons as per the operation flag value
       // i.e. operationFlag = 0 -> gives count of salary less than 10000
       // i.e. operationFlag = 1 -> gives count of salary more than 10000

    4. storeToFile (&structureArray, arraySize, fileName) ;
       // Function that stores the values of all persons from that array indices to the file name specified as the parameter name

4. Write the main function to test above mentioned functions using a menu. 20

4. Write a C program for the following:

1. Create a structure that has the members as arraySize, an array of dynamic size, minValue, maxValue and averageValue 10

2. Create a variable to store a number and input the value for the same from user. Store this value in the arraySize member of the structure and create the member array of same size 10

3. Define the following functions: 40

    1. fetchValues (filename, &structureVariable) ;
    // Function that fetches values from the specified file and store them in the array of the structure variable. The count of the numbers to be stored should match the size of the array

    2. calculateValues (&structureVariable, operationFlag) ;
    // Function that calculates the value from the array of the structure variable as per the operation flag specified.
    // e.g. if operation flag = 0 then calculate minimum
    // e.g. if operation flag = 1 then calculate maximum
    // e.g. if operation flag = 2 then calculate average
    // e.g. if operation flag = 3 then calculate all minimum, maximum and average

    3. showValues (&structureVariable) ;
    // Function that shows the values of the members of the structure variable on the screen

    4. storeToFile (&structureVariable, filename) ;
    // Function that stores the values of all members of the structure variable to the file specified as the parameter

4. Write the main function to test above mentioned functions using a menu. 20

5. Write a C program for the following:

A. Define a structure called node that represents a node of a linked list containing the data as an integer value. 10

2. Create the necessary pointers required to create a singly linked list and initialize them to appropriate values 10

3. Define the following functions: 40

   1. start = appendToList (start) ;
      // Function that create a new node and returns the start of the list

   2. traverseList (start) ;
      // Function that displays all the nodes of the list from the start node

   3. count = countNodes (start) ;
      // Function that returns the count of all the nodes available in the list

   4. storeToFile (start, filename) ;
      // Function that stores the values of each node to the file passed as parameter

4. Write the main function to test above mentioned functions using a menu. 20

6. Write a C program for the following:

1. Define two variables totalRows and totalCols and initialize them to the values passed as command line parameters 10

2. Define a dynamic two dimensional array which can work as matrix with number of rows and cols as the values in variables totalRows and totalCols respectively 10

3. Define the following functions: 40

    1. inputValues (matrix, rows, cols) ;
       // Function which takes input of values in the matrix of rows and cols size

    2. showValues (matrix, rows, cols) ;
       // Function that display the values from matrix in rows and cols format

    3. grandTotal = calculateSum (matrix, rows, cols, rowSumArray, colSumArray) ;
       // Function that calculates row sum from the matrix and stores in rowSumArray, calculates col sum and stores in colSumArray, and return the grand total

    4. storeToFile (matrix, rows, cols, filename) ;
       // Function that stores the values of the matrix to the file in appropriate format

4. Write the main function to test above mentioned functions using a menu. 20

7. Write a C program for the following:

1. Create a structure that has the members as valueToCheck, isArmstrong, isPalindrome 10

2. Create a variable of the above mentioned number and initialize its member valueToCheck to the value taken as input from the user 10

3. Define the following functions: 40

    1. returnStatus = isArmstrong (&structureVariable) ;
       // Function that checks whether the value in the structure is Armstrong or not and returns 1 if it is and 0 if it is not. Store the returned value in the appropriate member of the structure

    2. returnStatus = isPalindrome (&structureVariable) ;
       // Function that checks whether the value in the structure is Palindrome or not and returns 1 if it is and 0 if it is not. Store the returned value in the appropriate member of the structure

    3. returnStatus = checkValue (&structureVariable, operationFlag) ;
       // Function which checks whether the value is as required or not reusing the above mentioned two functions [isArmstrong and isPaindrome]
       // if operationFlag is 0 then checks for palindrome
       // if operationFlag is 1 then checks for armstrong

    4. storeToFile (&structureVariable, filename) ;
       // Function that stores the values of all members of the structure variable to the file specified as the parameter

4. Write the main function to test above mentioned functions using a menu. 20

8. Write a C program for the following:

1. Define a structure to store the details about a hostel, having members like the name of the hostel, the city name where the hostel is located, the yearly charge of the hostel and the number of students currently staying in that hostel. 05

2. Input a number which works as the size of the array. Create a dynamic array of the hostel structure. 05

3. Define the following functions:

    1. getData (&hostelVariable) ; 10
       // Function which takes input of all the members of the hostel structure variable passed a parameter

    2. showData (&hostelVariable) ; 10
       // Function which show the details of hostel structure variable

    3. totalCount = getCount (hostelArray, arraySize, countFlag, criteriaValue) ; 20
       // Function that counts the value from the array of the structure variable as per the count flag specified.
       // e.g. if count flag = 0 then count on the basis of yearly charge
       // e.g. if count flag = 1 then count on the basis of students staying
       // The criteriaValue is the value for which the count should be returned if it is more than the one stored in the structure variable
       // e.g. if count is 0 and criteriaValue is 1500 then the function will return the count of such hostels whose yearly charge is more than 1500 Rs.

    4. storeToFile (filename, hostelArray, arraySize) ; 10
       // Function which stores the hostels from the hostel array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

9. Write a C program for the following:

1. Define a structure to store the details about a movie, having members like the name of the movie, the release date of the movie, the director of the movie and the production house name of the movie. 05

2. Input a number which works as the size of the array. Create a dynamic array of the movie structure. 05

3. Define the following functions:

    1. getData (&movieVariable) ; 10
       // Function which takes input of all the members of the movie structure variable passed a parameter

    2. showData (&movieVariable) ; 10
       // Function which show the details of movie structure variable

    3. totalCount = getCount (movieArray, arraySize, releaseDate) ; 10
       // Function that returns the count of all such movies which have been released after the date specified as the releaseDate parameter.

    4. sortAndShow (movieArray, arraySize) ; 10
       // Function that sorts the movieArray on the basis of the movieName and then displays all the movies

    5. storeToFile (filename, movieArray, arraySize) ; 10
       // Function which stores the movies from the movie array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

10. Write a C program for the following:

1. Define a structure to store the details about a Game, having members like the name of the game, the venue name where the game was played, player name who won the game, the number of games won by the player. 05

2. Input a number which works as the size of the array. Create a dynamic array of the game structure. 05

3. Define the following functions:

    1. getData (&gameVariable) ; 10
       // Function which takes input of all the members of the game structure variable passed a parameter

    2. showData (&gameVariable) ; 10
       // Function which show the details of game structure variable

    3. playerName = getBestPlayer (gameArray, arraySize) ; 10
       // Function that returns the name of the player who has won the maximum number of games.

    4. totalVenues = getVenues (gameArray, arraySize, gameName, venuesArray) ; 10
       // Function that stores the names of all those venues in the venuesArray where the game was played whose name is passed as parameter

    5. storeToFile (filename, gameArray, arraySize) ; 10
       // Function which stores the games from the game array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

11. Write a C program for the following:

1. Define a structure to store the details about a Competition, having members like the name of the competition, the number of participants, the number of winners and the number of participating teams. 05

2. Input a number which works as the size of the array. Create a dynamic array of the competition structure. 05

3. Define the following functions:

    1. getData (&competitionVariable) ; 10
       // Function which takes input of all the members of the competition structure variable passed a parameter

    2. showData (&competitionVariable) ; 10
       // Function which shows details of competition structure variable

    3. numberOfTeams = getTotalTeams (competitionArray, arraySize) ; 10
       // Function that returns the total number of teams participated so far in all competitions held.

    4. compitionCount = showDiscrepancies (competitionArray, arraySize, compitionNames) ; 10
       // Function that returns the count of all such competitions where the number of winners is more than number of participants and stores the names of such competitions in competitionArray

    5. storeToFile (filename, competitionArray, arraySize) ; 10
       // Function which stores the competitions from the competition array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

12. Write a C program for the following:

1. Define a structure to store the details about Actors, having members like the name of the actor, number of movies acted in, number of awards won and age. 05

2. Input a number which works as the size of the array. Create a dynamic array of the actor structure. 05

3. Define the following functions:

    1. getData (&actorVariable) ; 10
       // Function which takes input of all the members of the actor structure variable passed a parameter

    2. showData (&actorVariable) ; 10
       // Function which shows details of actor structure variable

    3. count = showBestActors (actorArray, arraySize, actorNames) ; 10
       // Function that returns the count of all such actors who have won the maximum number of awards and store their names in the actorNames array.

    4. countValue = getCount (actorArray, arraySize, ageValue) ; 10
       // Function that returns the count of the actors who are above the age specified in the parameter

    5. storeToFile (filename, competitionArray, arraySize) ; 10
       // Function which stores the competitions from the competition array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

13. Write a C program for the following:

1. Define a structure to store the details about Investors, having members like the name of the investor, age of the investor, city where the investor stays and the total amount invested. 05

2. Input a number which works as the size of the array. Create a dynamic array of the investor's structure. 05

3. Define the following functions:

    1. getData (&investorVariable) ; 10
       // Function which takes input of all the members of the investor structure variable passed a parameter

    2. showData (&investorVariable) ; 10
       // Function which shows details of investor structure variable

    3. count = getInvestors (investorArray, arraySize, investedAmount, investorNames) ; 10
       // Function that returns the count of all such investors who have invested more than the amount passed as parameter and store their names in the investorNames.

    4. count = getDetails (investorArray, arraySize, cityName, investorDetails) ; 10
       // Function that returns the count of all those investors who belongs to the city specified in the parameter and store their details in the investorDetails array

    5. storeToFile (filename, investorArray, arraySize) ; 10
       // Function which stores the investors from the investor array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

14. Write a C program for the following:

1. Define a structure to store the details about Colleges, having members like the name of the college, the city where the college is located, the intake of the college and the number of students admitted by the college. 05

2. Input a number which works as the size of the array. Create a dynamic array of the college structure. 05

3. Define the following functions:

    1. getData (&collegeVariable) ; 10
       // Function which takes input of all the members of the college structure variable passed a parameter

    2. showData (&collegeVariable) ; 10
       // Function which shows details of college structure variable

    3. count = getColleges (collegeArray, arraySize, cityName, collegeNames) ; 10
       // Function that returns the count of all such colleges which are located in the city passed as parameter and store their names in the collegeNames array.

    4. count= getDetails (collegeArray, arraySize, collegeDetails) ; 10
       // Function that returns the count of all those colleges whose admitted students are more than the intake capacity of the college and store their details in the collegeDetails array

    5. storeToFile (filename, collegeArray, arraySize) ; 10
       // Function which stores the colleges from the college array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

15. Write a C program for the following:

1. Define a structure to store the details about Subjects, having members like the code, name, maximum marks, passing marks and credit. 05

2. Input a number which works as the size of the array. Create a dynamic array of the subject structure. 05

3. Define the following functions:

    1. getData (& subjectVariable) ; 10
       // Function which takes input of all the members of the subject structure variable passed a parameter

    2. showData (& subjectVariable) ; 10
       // Function which shows details of subject structure variable

    3. count = getSubjects (subjectArray, arraySize, subjectNames) ; 10
       // Function that returns the count of all such subjects which have the credit as 10% of the maximum marks and store their names in the subjectNames array.

    4. count = getDetails (subjectArray, arraySize, subjectCode, subjectDetails) ; 10
       // Function that returns the count of all that subject whose code matches the code that is passed in parameter and store their complete details in the last parameter.

    5. storeToFile (filename, subjectArray, arraySize) ; 10
       // Function which stores the subjects from the subject array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

16. Write a C program for the following:

1. Define a structure to store the details about Restaurants, having members like the name, city, number of cuisine offered and names of all cuisine offered. 05

2. Input a number which works as the size of the array. Create a dynamic array of the restaurant structure. 05

3. Define the following functions:

    1. getData (&restaurantVariable) ; 10
       // Function which takes input of all the members of the subject structure variable passed a parameter

    2. showData (&restaurantVariable) ; 10
       // Function which shows details of subject structure variable

    3. count = getSubjects (restaurantArray, arraySize, cuisineName, restaurantNames) ; 10
       // Function that returns the count of all such restaurants which offer the cuisine passed as parameter and store their names in the last parameter.

    4. countValue = getCount (restaurantArray, arraySize, cityName) ; 10
       // Function that returns the count of restaurants available in the city passed as parameter

    5. storeToFile (filename, restaurantArray, arraySize) ; 10
       // Function which stores the restaurants from the restaurant array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

17. Write a C program for the following:

1. Define a structure named PrimeNumbers which stores a file name, size of an array, an array of specified size to store numbers and a variable to store count 10

2. Create a variable of the structure. Input a number which works as the size of the array. Create a dynamic array inside the structure variable as per the user specified size 10

3. Define the following functions: 40

    1. fetchNumbers (&structureVariable) ;
       // Function which fetches data from the file name specified inside the structure variable and stores the numbers from the file into the array. The total number of values to be fetched equals the size specified in the structure variable

    2. returnStatus = isPrime (value) ;
       // Function which checks whether the value specified as parameter is prime or not and returns 1 if prime and 0 if not prime

    3. count = getPrimes (&structureVariable) ;
       // Function which fetches all numbers from the file specified in the structure variable and stores them in the array inside the structure. [Store only size number of values which are prime from the file in the array]. The function returns the count of total prime numbers stored. [Reuse the above mentioned function to check a number is prime or not]

    4. sum = totalOfPrimes (&structureVariable) ;
       // Function which returns the summation of all the prime numbers stored in the structure variable

4. Write the main function to test above mentioned functions using a menu. 20

18. Write a C program for the following:

1. Create a structure which has the data members as filename, totalCharacters, totalWords and totalLines 10

2. Create a variable of the above mentioned structure. The member variable filename should be initialized to the string passed as command line argument 10

3. Define the following functions: 40

    1. totalChars = countTotalCharacters (&structureVariable) ;
       // Function which counts the total characters from the file name stored in the structure variable and stored in the totalCharacters member of the structure variable and return the same count

    2. totalWords = countTotalWords (&structureVariable) ;
       // Function which counts the total Words from the file name stored in the structure variable and stored in the totalWords member of the structure variable and return the same count

    3. totalLines = countTotalLines (&structureVariable) ;
       // Function which counts the total lines from the file name stored in the structure variable and stored in the totalLines member of the structure variable and return the same count

    4. storeToFile (&structureVariable, filename) ;
       // Function which stores the details of the structure variable to the file specified as the file name parameter. The details should be stored as comma separated values, one record per line

5. Write the main function to test above mentioned functions using a menu. 20

19. Write a C program for the following:

1. Create a structure with the members as ordinaryString, compressedString, lengthOfOrdinaryString and lengthOfCompressedString 10

2. Create a variable of the structure and initialize both the member strings to the values passed as command line arguments 10

3. Define the following functions: 40

    1. compressTheString (&structureVariable) ;
       // Function that compresses the ordinaryString of the structure variable and store it to the compressedString member of the structure variable.
       // e.g aaabcccccdde will be compressed to a3bc5dde

    2. showStrings (&structureVariable) ;
       // Function that shows both the ordinary string and the compressed strings along with their respective lengths

    3. status = isPalindrome (&structureVariable) ;
       // Function which checks whether the two strings stored in the structure variable are palindrome. The function returns 1 if they are palindrome and 0 in case they are not palindrome

    4. storeToFile (&structureVariable, fileName) ;
       // Function which stores all the members of the structure variable in the file specified as the parameter

4. Write the main function to test above mentioned functions using a menu. 20

20. Write a C program for the following:

1. Define a structure Country that contains the name of country, its capital, it population and its national language 10

2. Create a dynamic array of the above mentioned structure. The size of the array should be taken as input from the user 10

3. Define the following functions: 40

    1. getData (&structureArray, arraySize) ;
       // Function that takes input of the members of the structure variable

    2. showData (&structureArray, arraySize) ;
       // Function that displays values of the members of the structure variable

    3. count = languageSpeakingCountry (&structureArray, arraySize, language) ;
       // Function that returns the count of all such countries that has national language as the language specified in the parameter

    4. storeToFile (&structureArray, arraySize, filename) ;
       // Function that stores the details of all countries to the file specified in the parameter name

5. Write the main function to test above mentioned functions using a menu. 20

21. Write a C program for the following:

1. Define a structure Student that contains the roll no of the student, name of student, the course he is enrolled in, the semester he studies in and the average score he has obtained 05

2. Create a dynamic array of the above mentioned structure as per the size of entered by the user 05

3. Define the following functions: 50

    1. getData (&structureArray, arraySize) ;
       // Function that takes input of the members of the structure variable

    2. showData (&structureArray, arraySize) ;
       // Function that displays values of the members of the structure variable

    3. count = getNames (&structureArray, arraySize, semesterNumber, restaurantNames) ;
       // Function that returns the count of those students who study in the semester specified as parameter and store their names in the last parameter

    4. count = getCount (&structureArray, arraySize, averageScore) ;
       // Function that returns the count of all such students that has average more than the average score specified in the parameter

    5. storeToFile (&structureArray, arraySize, filename) ;
       // Function that stores the details of all students to the file specified in the parameter name

5. Write the main function to test above mentioned functions using a menu. 20

22. Write a C program for the following:

1. Define a structure to store the details about Elections, having members like the election name, election held year, total candidates, winning candidate name. 05

2. Input a number which works as the size of the array. Create a dynamic array of the election structure. 05

3. Define the following functions:

    1. getData (& electionVariable) ; 10
       // Function which takes input of all the members of the election structure variable passed a parameter

    2. showData (&electionVariable) ; 10
       // Function which shows details of election structure variable

    3. count = getCandidates (electionArray, arraySize, year, candidateNames) ; 10
       // Function that returns the count of all those candidates who have won the elections held in the year passed as parameter and store their names in the last parameter.

    4. countValue = getCount (electionArray, arraySize, year) ; 10
       // Function that returns the total number of elections held in the year passed as parameter

    5. storeToFile (filename, electionArray, arraySize) ; 10
       // Function which stores the elections from the election array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

23. Write a C program for the following:

1. Define a structure to store the details about Zones, having members like the zone name, the number of institutes in the zone, the number of faculties in the zone, the number of students in the zone. 05

2. Input a number which works as the size of the array. Create a dynamic array of the zone structure. 05

3. Define the following functions:

    1. getData (& zoneVariable) ; 10
       // Function which takes input of all the members of the zone structure variable passed a parameter

    2. showData (& zoneVariable) ; 10
       // Function which shows details of zone structure variable

    3. count = getZone (zoneArray, arraySize, criteriaValue, zoneNames) ; 10
       // Function that returns the count of all those zones which has the number of institutes more than the criteriaValue passed as parameter and store their names in the last parameter.

    4. zoneNames = getCount (zone Array, arraySize) ; 10
       // Function that returns the name of all those zone which has the number of faculties four times more than the number of institutes in the zone

    5. storeToFile (filename, zone Array, arraySize) ; 10
       // Function which stores the zones from the zone array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

24. Write a C program for the following:

1. Define a structure to store the details about Jails, having members like the jail name, the city where the jail is located, the number of criminals in jail, the number of guards in the jail and the year the jail was established. 05

2. Input a number which works as the size of the array. Create a dynamic array of the jail structure. 05

3. Define the following functions:

    1. getData (& jailVariable) ; 10
       // Function which takes input of all the members of the jail structure variable passed a parameter

    2. showData (&jailVariable) ; 10
       // Function which shows details of jail structure variable

    3. countValue = getCount (jailArray, arraySize, city) ; 10
       // Function that returns the total number of criminals in all the jails of the city passed as parameter.

    4. count = getJailNames (jailArray, arraySize, jailNames) ; 10
       // Function that returns the count of all those jails which has the number of guards half to the number of criminals in the jail and store their names in the last parameter

    5. storeToFile (filename, jailArray, arraySize) ; 10
       // Function which stores the jails from the jail array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

25. Write a C program for the following:

1. Define a structure to store the details about Courts, having members like the court name, the number of cases in the court, the number of judges in the court, the names of all judges of the court and the city name where the court is located. 05

2. Input a number which works as the size of the array. Create a dynamic array of the court structure. 05

3. Define the following functions:

    1. getData (& courtVariable) ; 10
       // Function which takes input of all the members of the jail structure variable passed a parameter

    2. showData (& courtVariable) ; 10
       // Function which shows details of jail structure variable

    3. countValue = getCount (courtArray, arraySize, city) ; 10
       // Function that returns the total number of cases in all the courts of the city passed as parameter.

    4. count = getJudgeNames (courtArray, arraySize, courtName, judgeNames) ; 10
       // Function that returns the count of all those judges who belong to the court passed as parameter and store their names in the last parameter

    5. storeToFile (filename, courtArray, arraySize) ; 10
       // Function which stores the courts from the court array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

26. Write a C program for the following:

1. Define a structure to store the details about BloodDonors, having members like the donor name, donor blood group, donor city, numberOfTimesBloodDonated. 05

2. Input a number which works as the size of the array. Create a dynamic array of the BloodDonors structure. 05

3. Define the following functions:

    1. getData (& structureVariable) ; 10
       // Function which takes input of all the members of the structure variable passed a parameter

    2. showData (& structureVariable) ; 10
       // Function which shows details of structure variable

    3. count = getMaximum (structureArray, arraySize, names) ; 10
       // Function that returns the count of all those donors who have donated blood for the maximum time and store their names in the last parameter.

    4. count = getNames (structureArray, arraySize, bloodGroup, donors) ; 10
       // Function that returns the count of all those donors who have the blood group passed as parameter and also have donated blood for the maximum time and store their names in the last parameter

    5. storeToFile (filename, structureArray, arraySize) ; 10
       // Function which stores the structures from the structure array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

27. Write a C program for the following:

1. Define a structure to store the details about Transport, having members like the transport name, the number of routes, the number of vehicles per route, the number of places where transport available and the name of those places where transport is provided. 05

2. Input a number which works as the size of the array. Create a dynamic array of the transport structure. 05

3. Define the following functions:

    1. getData (& structureVariable) ; 10
       // Function which takes input of all the members of the structure variable passed a parameter

    2. showData (& structureVariable) ; 10
       // Function which shows details of structure variable

    3. count = getNames (structureArray, arraySize, placeName, names) ; 10
       // Function that returns the count of all those transports which provides transport to the place passed as parameter and store their names in the last parameter.

    4. countValue = getCount (structureArray, arraySize) ; 10
       // Function that returns the count of all those transports which has the maximum number of routes and maximum number of vehicles per route

    5. storeToFile (filename, structureArray, arraySize) ; 10
       // Function which stores the structures from the structure array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

28. Write a C program for the following:

1. Define a structure to store the details about Routes, having members like the route number, route name, the total number of drivers for route, driver names and total distance of the route. 05

2. Input a number which works as the size of the array. Create a dynamic array of the route structure. 05

3. Define the following functions:

    1. getData (& structureVariable) ; 10
       // Function which takes input of all the members of the structure variable passed a parameter

    2. showData (& structureVariable) ; 10
       // Function which shows details of structure variable

    3. count = getDriverNames (structureArray, arraySize, names) ; 10
       // Function that returns the count of all those drivers who drives on the longest route and store their names in the last parameter.

    4. routeDistance = getLongestRoute (structureArray, arraySize) ; 10
       // Function that returns the distance of the longest route

    5. storeToFile (filename, structureArray, arraySize) ; 10
       // Function which stores the structures from the structure array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

29. Write a C program for the following:

1. Define a structure to store the details about Cities, having members like the city name, the number of areas in the cities, the names of all areas in the city and the pin codes of all areas of the city. 05

2. Input a number which works as the size of the array. Create a dynamic array of the transport structure. 05

3. Define the following functions:

    1. getData (& structureVariable) ; 10
       // Function which takes input of all the members of the structure variable passed a parameter

    2. showData (& structureVariable) ; 10
       // Function which shows details of structure variable

    3. count = getAreaNames (structureArray, arraySize, pinCode, names) ; 10
       // Function that returns the count of all those areas whose pin code matches the value passed as parameter and store their names in the last parameter.

    4. maximumValue = getMaximum (structureArray, arraySize, cityName) ; 10
       // Function that returns the maximum pin code from the city passed as parameter

    5. storeToFile (filename, structureArray, arraySize) ; 10
       // Function which stores the structures from the structure array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

30. Write a C program for the following:

1. Define a structure to store the details about Tours, having members like the tour name, the starting place, the destination place, the price of the tour package, number of places covered in the tour. 05

2. Input a number which works as the size of the array. Create a dynamic array of the tour structure. 05

3. Define the following functions:

    1. getData (& structureVariable) ; 10
       // Function which takes input of all the members of the structure variable passed a parameter

    2. showData (& structureVariable) ; 10
       // Function which shows details of structure variable

    3. count = getTourNames (structureArray, arraySize, startPlace, endPlace, desiredCost, names) ; 10
       // Function that returns the count of all those tours which starts and ends at the places passed as parameters and also has the cost less than the cost passed as parameter and store their names in the last parameter.

    4. maximumValue = getCostliestTour (structureArray, arraySize, cityName) ; 10
       // Function that returns the maximum cost for the tour with the destination passed as parameter

    5. storeToFile (filename, structureArray, arraySize) ; 10
       // Function which stores the structures from the structure array to the file specified with the file name parameter to the function

4. Write the main function to test above mentioned functions using a menu. 20

31. Write a C program to perform following operation on student record.

A. Design a structure called Student having following data members. 05

1. Roll number
2. Name
3. Semester
4. Subject 1
5. Subject 2
6. Subject 3
7. Max marks
8. Subject1 obtained marks
9. Subject2 obtained marks
10. Subject3 obtained marks
11. Total
12. Percentage
13. Grade

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addStudent(&studentObject);
   // Add student details to the structure variable which is passed as an argument
   // Total and Grade should not be entered by user. It must be calculated by program
   // The marks of three subjects must be less than equal to Max marks
2. displayStudent(&studentObject);
   // Displays the student details from the structure variable passed as an argument
3. calTotalPer(&studentObject);
   // Calculate the total of the student from the structure variable passed as an argument and percentage
4. calGrade(&studentObject);
   // Calculate the grade of the student form the structure variable passed as an argument
   // Following is the grade table calculation.

Percentage	**Grade**	**Percentage**	**Grade**
100-90	A+	89-70	A
69-60	B+	59-50	B
49-40	C	39-0	F

C. Generate a report which display the following information related to the student. 20

(Display records on console)

1. Display the details of all the students
   (hint: use for loop and in that call displayStudent(&studentObject) function)
2. Display the details of failed or passed students

D. The details of all the students should be stored in a file called 10

“studentData.dat”.
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of student and the user defined functions listed above. 05

32. Write a C program to perform following operation on student record.

A. Design a structure called Student having following data members. 05

1. Roll number
2. Name
3. Semester
4. Subject 1
5. Subject 2
6. Subject 3
7. Max marks
8. Array of three subjects marks
9. Total
10. Percentage
11. Grade

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addStudent(&studentObject);
   // Add student details to the structure variable which is passed as an argument
   // Total and Grade should not be entered by user. It must be calculated by program
   // The marks of three subjects must be less than equal to Max marks
2. displayStudent(&studentObject);
   // Displays the student details from the structure variable passed as an argument
3. calTotalPer(&studentObject);
   // Calculate the total of the student from the structure variable passed as an argument and percentage
4. calGrade(&studentObject);
   // Calculate the grade of the student form the structure variable passed as an argument
   // Following is the grade table calculation.

Percentage	**Grade**	**Percentage**	**Grade**
100-90	A+	89-70	A
69-60	B+	59-50	B
49-40	C	39-0	F

C. Generate a report which display the following information related to the student. 20

(Display records on console)

1. Display the details of all the students
   (hint: use for loop and in that call displayStudent(&studentObject) function)
2. Display the details of those students who got grade A and C

D. The details of all the students should be stored in a file called "studentData.dat". 10

Use fwrite ( ) and fread ( ) function to store to file and read from file

respectively

E. Write an appropriate main function to test the structure of student and the user defined functions listed above. 05

33. Write a C program to perform following operation on student record.

A. Design a structure called Student having following data members. 05

1. First name
2. Last name
3. Graduate Degree
4. Passed year (YYYY)
5. Percentage of graduate degree
6. Post graduate degree
7. Passed year (YYYY)
8. Percentage of post graduate degree
9. Points

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addStudent(&studentObject); [10]
   // Add student details to the structure variable which is passed as an argument
   // Points should not added by user. It should be calculated by program
2. displayStudent(&studentObject); [10]
   // Displays the student details from the structure variable passed as an argument
3. calPoints(&studentObject); [20]
   // Calculate the points of the students from the structure variable passed as an argument. (User should add the points of graduate and post graduate degree)
   Points calculation given below

Graduate degree	**Points**	**Post graduate degree**	**Points**
B.Com	10	M.Com	20
B.B.A.	20	M.B.A.	50
B.C.A.	40	M.C.A.	60
B.Sc	20	M.Sc	40
B.Sc.IT	30	M.Sc.IT	50

C. Generate a report which display the following information related to the student. 20
(Display records on console)

1. Display the details of all the students
   (hint: use for loop and in that call displayStudent(&studentObject) function)
2. Display the details of those students who got more than 50 points

D. The details of all the students should be stored in a file called "studentData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of student and the user defined functions listed above. 05

34. Write a C program to perform following operation on student record.

A. Design a structure called Student having following data members. 05

1. First name
2. Last name
3. Graduate Degree
4. Passed year (YYYY)
5. Percentage of graduate degree
6. Post graduate degree
7. Passed year (YYYY)
8. Percentage of post graduate degree
9. Points

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addStudent(&studentObject); [10]
   // Add student details to the structure variable which is passed as an argument
   // Points should not added by user. It should be calculated by program
2. displayStudent(&studentObject); [10]
   // Displays the student details from the structure variable passed as an argument
3. calPoints(&studentObject); [20]
   // Calculate the points of the students from the structure variable passed as an argument. (User should add the points of graduate and post graduate degree)
   Points calculation given below

Graduate degree	**Points**	**Post graduate degree**	**Points**
B.Com	10	M.Com	20
B.B.A.	20	M.B.A.	50
B.C.A.	40	M.C.A.	60
B.Sc	20	M.Sc	40
B.Sc.IT	30	M.Sc.IT	50

C. Generate a report which display the following information related to the student. 20

(Display records on console)

1. Display the details of all the students
   (hint: use for loop and in that call displayStudent(&studentObject) function)
2. Display the details of those students whose post graduate degree is M.C.A. OR M.Sc.IT
   // Use string handling functions

D. The details of all the students should be stored in a file called "studentData.dat". 10

Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of student and the user defined functions listed above. 05

35. Write a C program to perform following operation on student record.

A. Design a structure called Student having following data members. 05

1. First name
2. Last name
3. Graduate Degree
4. Passed year (YYYY)
5. Percentage of graduate degree
6. Post graduate degree
7. Passed year (YYYY)
8. Percentage of post graduate degree
9. Points

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addStudent(&studentObject); [10]
   // Add student details to the structure variable which is passed as an argument
   // Points should not added by user. It should be calculated by program
2. displayStudent(&studentObject); [10]
   // Displays the student details from the structure variable passed as an argument
3. calPoints(&studentObject); [20]
   // Calculate the points of the students from the structure variable passed as an argument. (User should add the points of graduate and post graduate degree)
   Points calculation given below

Graduate degree	**Points**	**Post graduate degree**	**Points**
B.Com	10	M.Com	20
B.B.A.	20	M.B.A.	50
B.C.A.	40	M.C.A.	60
B.Sc	20	M.Sc	40
B.Sc.IT	30	M.Sc.IT	50

C. Generate a report which display the following information related to the student. 20

(Display records on console)

1. Display the details of all the students
   (hint: use for loop and in that call displayStudent(&studentObject) function)
2. Display the details of those students whose graduate degree is B.C.A. OR B.Sc.IT and Post graduate degree is NULL.
   // Use string handling functions

D. The details of all the students should be stored in a file called "studentData.dat". 10

Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of student and the user defined functions listed above. 05

36. Write a C program to perform following operation on employee record.

A. Design a structure called Employee having following data members. 05

1. First name
2. Last name
3. Company name
4. Designation
5. Check-in time
6. Check-out time
7. Salary

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addEmployee(&employeeObject); [10]
   // Add employee details to the structure variable which is passed as an argument
2. displayEmployee(&employeeObject); [10]
   // Displays the employee details from the structure variable passed as an argument
3. calTime(&employeeObject); [20]
   // Calculate the time of employee form the structure variable passed as an argument

C. Generate a report which display the following information related to the employee. 20
(Display records on console)

1. Display the details of all the employee
   (hint: use for loop and in that call displayEmployee(&employeeObject) function)
2. Display the details of those employee who worked for more than 10 hours

D. The details of all the employees should be stored in a file called "employeeData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of employee and the user defined functions listed above. 05

37 Write a C program to perform following operation on employee record.

A. Design a structure called Employee having following data members. 05

1. First name
2. Last name
3. Company name
4. Designation
5. Check-in time
6. Check-out time
7. Salary

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addEmployee(&employeeObject); [10]
   // Add employee details to the structure variable which is passed as an argument
2. displayEmployee(&EmployeeObject); [10]
   // Displays the employee details from the structure variable passed as an argument
3. calOverTime(&employeeObject); [20]
   // Calculate the time of employee form the structure variable passed as an argument, and those worked more than 10 hours add 1000 to their salary

C. Generate a report which display the following information related to the employee. 20
(Display records on console)

1. Display the details of all the employee
   (hint: use for loop and in that call displayEmployee(&employeeObject) function)
2. Display the details of those employee whose salary is more than 30000

D. The details of all the employees should be stored in a file called "employeeData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of employee and the user defined functions listed above. 05

38. Write a C program to perform following operation on employee record.

A. Design a structure called Employee having following data members. 05

1. First name
2. Last name
3. Company name
4. Designation
5. Basic pay
6. HRA
7. DA
8. PF
9. Net pay

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addEmployee(&employeeObject); [10]
   // Add employee details to the structure variable which is passed as an argument
   // Net pay should not be entered by user, it should be calculate on the basis of basic pay, hra, da and pf. And the value of hra, da and pf should be entered as percentage (E.g. HRA=20% of basic pay, DA=60% of basic pay, PF=25% of basic pay
2. displayEmployee(&employeeObject); [10]
   // Displays the employee details from the structure variable passed as an argument
3. calNetPay(&employeeObject); [20]
   // Calculate the net payment of the employee on the basis of basic pay and store it to Net pay variable

C. Generate a report which display the following information related to the employee. 20
(Display records on console)

1. Display the details of all the employee
   (hint: use for loop and in that call displayEmployee(&employeeObject) function)
2. Display the details of those employee whose Net pay is larger than 40000

D. The details of all the employees should be stored in a file called "employeeData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of employee and the user defined functions listed above. 05

39. Write a C program to perform following operation on employee record.

A. Design a structure called Employee having following data members. 05

1. First name
2. Last name
3. Company name
4. Designation
5. Basic pay
6. HRA
7. DA
8. PF
9. Net pay
10. No. of leaves

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addEmployee(&employeeObject); [10]
   // Add employee details to the structure variable which is passed as an argument
   // Net pay should not be entered by user, it should be calculate on the basis of basic pay, hra, da and pf. And the value of hra, da and pf should be entered as percentage (E.g. HRA=20% of basic pay, DA=60% of basic pay, PF=25% of basic pay
2. displayEmployee(&employeeObject); [10]
   // Displays the employee details from the structure variable passed as an argument
3. calNetPay(&employeeObject); [20]
   // Calculate the net payment of the employee on the basis of basic pay and store it to Net pay variable

C. Generate a report which display the following information related to the employee. 20
(Display records on console)

1. Display the details of all the employee
   (hint: use for loop and in that call displayEmployee(&employeeObject) function)
2. Display the details of those employee whose Net pay is larger than 40000 and the values of No. of leaves is greater than equal to 3.

D. The details of all the employees should be stored in a file called "employeeData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of employee and the user defined functions listed above. 05

40. Write a C program to perform following operation on employee record.

A. Design a structure called Employee having following data members. 05

1. First name
2. Last name
3. Company name
4. Designation
5. Basic pay
6. HRA
7. DA
8. PF
9. Net pay
10. No. of leaves

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addEmployee(&employeeObject); [10]
   // Add employee details to the structure variable which is passed as an argument
   // Net pay should not be entered by user, it should be calculate on the basis of basic pay, hra, da and pf. And the value of hra, da and pf should be entered as percentage (E.g. HRA=20% of basic pay, DA=60% of basic pay, PF=25% of basic pay
2. displayEmployee(&EmployeeObject); [10]
   // Displays the employee details from the structure variable passed as an argument
3. calNetPay(&employeeObject); [20]
   // Calculate the net payment of the employee on the basis of basic pay and store it to Net pay variable. If No. of leaves is more than 3 than subtract 500 rs for each day.
   (E.g. if no of leaves is 4 subtract 500 rs. If 5 subtract 1000 rs. If 6 subtract 1500 etc.)

C. Generate a report which display the following information related to the employee. 20
(Display records on console)

1. Display the details of all the employee
   (hint: use for loop and in that call displayEmployee(&employeeObject) function)
2. Display the details of those employee whose Net pay is larger than 40000 and the values of No. of leaves is greater than 3.

D. The details of all the employees should be stored in a file called "employeeData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of employee and the user defined functions listed above. 05

41. Write a C program to perform following operation on employee record.

A. Design a structure called Employee having following data members. 05

1. First name
2. Last name
3. Company name
4. Designation
5. Basic pay
6. HRA
7. DA
8. PF
9. Net pay
10. No. of leaves

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addEmployee(&employeeObject); [10]
   // Add employee details to the structure variable which is passed as an argument
   // Net pay should not be entered by user, it should be calculate on the basis of basic pay, hra, da and pf. And the value of hra, da and pf should be entered as percentage (E.g. HRA=20% of basic pay, DA=60% of basic pay, PF=25% of basic pay
2. displayEmployee(&employeeObject); [10]
   // Displays the employee details from the structure variable passed as an argument
3. calNetPay(&employeeObject); [20]
   // Calculate the net payment of the employee on the basis of basic pay and store it to Net pay variable. If No. of leaves is more than 3 than subtract 500 rs for each day.
   (E.g. if no of leaves is 4 subtract 500 rs. If 5 subtract 1000 rs. If 6 subtract 1500 etc.)

C. Generate a report which display the following information related to the employee. 20

1. Delete the records of employee whose salary is less than 10000 from file.
   // Hint: Use two files.

D. The details of all the employees should be stored in a file called "employeeData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of employee and the user defined functions listed above. 05

42. Write a C program to perform following operation on faculty record.

A. Design a structure called Faculty having following data members. 05

1. First name
2. Last name
3. College name
4. Designation
5. Stream
6. Basic pay
7. HRA
8. DA
9. PF
10. Net pay
11. No. of leaves

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addFaculty(&facultyObject); [10]
   // Add faculty details to the structure variable which is passed as an argument
   // Net pay should not be entered by user, it should be calculate on the basis of basic pay, hra, da and pf. And the value of hra, da and pf should be entered as percentage (E.g. HRA=20% of basic pay, DA=60% of basic pay, PF=25% of basic pay
2. displayFaculty(&facultyObject); [10]
   // Displays the faculty details from the structure variable passed as an argument
3. findStream(&facultyObject); [20]
   // Find the name of those faculties who teach in MCA stream (Use string handling function)

C. Generate a report which display the following information related to the faculties. 20

1. Delete the records of faculty whose stream is BCom.
   // Hint: Use two files.

D. The details of all the faculties should be stored in a file called "faclultyData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of faculty and the user defined functions listed above. 05

43. Write a C program to perform following operation on faculty record.

A. Design a structure called Faculty having following data members. 05

1. First name
2. Last name
3. College name
4. Designation
5. Stream
6. Basic pay
7. Subject taught1
8. Subject taught2
9. Subject taught3

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addFaculty(&facultyObject); [10]
   // Add faculty details to the structure variable which is passed as an argument
   // If faculty take 1 subject only then subject taught2 and subject taught3 should have NULL values. And if faculty take 2 subjects then subject taught3 should be NULL
2. displayFaculty(&facultyObject); [10]
   // Displays the faculty details from the structure variable passed as an argument
3. findStream(&facultyObject); [20]
   // Find the name of those faculties who teach in MCA stream (Use string handling function)

C. Generate a report which display the following information related to the faculties. 20
(Display records on console)

1. Display the details of all the faculties
   (hint: use for loop and in that call displayFaculty(&facultyObject) function)
2. Display the details of only those faculties who taught only 2 subjects.

D. The details of all the faculties should be stored in a file called "faclultyData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of faculty and the user defined functions listed above. 05

44. Write a C program to perform following operation on item record.

A. Design a structure called Item having following data members. 05

1. Item name
2. Item category
3. Item type
4. Manufacture price
5. Sell price
6. Order date (DD)
7. Order month (MM)
8. Order year (YYYY)

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addItem(&itemObject); [10]
   // Add item details to the structure variable which is passed as an argument
2. displayItem(&itemObject); [10]
   // Displays the item details from the structure variable passed as an argument
3. findDifference(&itemObject); [20]
   // Find the name of item whose manufacture's price and selling's price difference is more than 10%

C. Generate a report which display the following information related to the items. 20

(Display records on console)

1. Display the details of all the items
   (hint: use for loop and in that call displayItem(&itemObject) function)
2. Display the details of only those items which belongs to cloth category (Use string handling function)

D. The details of all the items should be stored in a file called "itemData.dat". 10

Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of item and the user defined functions listed above. 05

45. Write a C program to perform following operation on item record.

A. Design a structure called Item having following data members. 05

1. Item name
2. Item category
3. Item type
4. Manufacture price
5. Sell price
6. Order date (DD)
7. Order month (MM)
8. Order year (YYYY)

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addItem(&itemObject); [10]
   // Add item details to the structure variable which is passed as an argument
2. displayItem(&itemObject); [10]
   // Displays the item details from the structure variable passed as an argument
3. findDifference(&itemObject); [20]
   // Find the name of item whose manufacture's price and selling's price difference is more than 10%

C. Generate a report which display the following information related to the items. 20

(Display records on console)

1. Display the details of all the items
   (hint: use for loop and in that call displayItem(&itemObject) function)
   And display the date in DD/MM/YYYY format using string handling function.

D. The details of all the items should be stored in a file called "itemData.dat". 10

Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of item and the user defined functions listed above. 05

46. Write a C program to perform following operation on item record.

A. Design a structure called Item having following data members. 05

1. Item name
2. Item category
3. Item type
4. Manufacture price
5. Sell price
6. Order date (DD)
7. Order month (MM)
8. Order year (YYYY)

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addItem(&itemObject); [10]
   // Add item details to the structure variable which is passed as an argument
2. displayItem(&itemObject); [10]
   // Displays the item details from the structure variable passed as an argument
3. findDifference(&itemObject); [20]
   // Find the name of item whose manufacture's price and selling's price difference is more than 10%

C. Generate a report which display the following information related to the items. 20

(Display records on console)

1. Display the details of all the items
   (hint: use for loop and in that call displayItem(&itemObject) function)
2. Display the details of those items which are sold in current year.

D. The details of all the items should be stored in a file called "itemData.dat". 10

Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of item and the user defined functions listed above. 05

47. Write a C program to perform following operation on item record.

A. Design a structure called Item having following data members. 05

1. Item name
2. Item category
3. Item type
4. Manufacture price
5. Sell price
6. Order date (DD)
7. Order month (MM)
8. Order year (YYYY)

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addItem(&itemObject); [10]
   // Add item details to the structure variable which is passed as an argument
2. displayItem(&itemObject); [10]
   // Displays the item details from the structure variable passed as an argument
3. findDifference(&itemObject); [20]
   // Find the name of item whose manufacture's price and selling's price difference is more than 10%

C. Generate a report which display the following information related to the items. 20

1. Store the data of items in the file which are sold, having 10% profit margin.

D. The details of all the items should be stored in a file called "itemData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of item and the user defined functions listed above. 05

48. Write a C program to perform following operation on link list.

A. Design a structure called Node having following data members. 05

1. Data value
2. Address of next node

B. Design appropriate user defined functions to accomplish following listed tasks 60

1. addNode(&nodeObject); [10]
   // Add node details to the structure variable which is passed as an argument
2. displayNode(&nodeObject); [10]
   // Displays the node details from the structure variable passed as an argument
3. findDifference(&nodeObject); [20]
   // Find the two consecutives nodes whose difference is more than 10
4. findMaxMin(&nodeObject); [20]
   // Find the index of the node having maximum value.

C. Write an appropriate main function to test the structure of node and the user defined functions listed above. 05

49. Write a C program to perform following operation on link list.

A. Design a structure called Node having following data members. 05

1. Data value
2. Address of next node

B. Design appropriate user defined functions to accomplish following listed tasks 60

1. addNode(&nodeObject); [10]
   // Add node details to the structure variable which is passed as an argument
2. displayNode(&nodeObject); [10]
   // Displays the node details from the structure variable passed as an argument
3. search(&nodeObject,serachElement); [10]
   // Find the index of the node which is passed along with argument
4. sort(&nodeObject); [30]
   // Sort the list whose structure variable is passed as an argument

C. Write an appropriate main function to test the structure of node and the user defined functions listed above. 05

50. Write a C program to perform following operation on book record.

A. Design a structure called Book having following data members. 05

1. Book name
2. Book subject
3. Author1
4. Author2
5. Production price
6. Selling price
7. Publication
8. Publication year

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addBook(&bookObject); [10]
   // Add book details to the structure variable which is passed as an argument
2. displayBook(&bookObject); [10]
   // Displays the book details from the structure variable passed as an argument
3. findBook(&bookObject); [20]
   // Find the name of book whose production's price and selling's price difference is more than 10%

C. Generate a report which display the following information related to the books. 20

1. Display the details of all the books
   (hint: use for loop and in that call displayBook(&bookObject) function)
2. Display the name of books whose publications is PHI (Use string handling function)

D. The details of all the books should be stored in a file called "bookData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of book and the user defined functions listed above. 05

51. Write a C program to perform following operation on book record.

A. Design a structure called Book having following data members. 05

1. Book name
2. Book subject
3. Author1
4. Author2
5. Production price
6. Selling price
7. Publication
8. Publication year

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addBook(&bookObject); [10]
   // Add book details to the structure variable which is passed as an argument
2. displayBook(&bookObject); [10]
   // Displays the book details from the structure variable passed as an argument
3. findPublication(&bookObject,publication); [20]
   // Find the name of books whose publication is passed through argument to a function

C. Generate a report which display the following information related to the books. 20

1. Display the details of all the books
   (hint: use for loop and in that call displayBook(&bookObject) function)
2. Count the number of books published in the year 2012 and display the total count on console

D. The details of all the books should be stored in a file called "bookData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of book and the user defined functions listed above. 05

52. Write a C program to perform following operation on hotel record.

A. Design a structure called Hotel having following data members. 05

1. Hotel name
2. Type of hotel (E.g. 5 star, 4 star etc.)
3. Address
4. Area
5. Number of rooms
6. Number of allocated rooms

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addHotel(&hotelObject); [10]
   // Add hotel details to the structure variable which is passed as an argument
2. displayHotel(&hotelObject); [10]
   // Displays the hotel details from the structure variable passed as an argument
3. findHotel(&hotelObject); [20]
   // Find the name of hotels belongs to Ahmadabad area (Use string handling functions)

C. Generate a report which display the following information related to the hotel. 20

1. Display the details of all the hotels
   (hint: use for loop and in that call displayHotel(&hotelObject) function)
2. Count the number of 5 star hotels belongs to particular area.

D. The details of all the hotels should be stored in a file called "hotelData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of hotel and the user defined functions listed above. 05

53. Write a C program to perform following operation on hotel record.

A. Design a structure called Hotel having following data members. 05

1. Hotel name
2. Type of hotel (E.g. 5 star, 4 star etc.)
3. Address
4. Area
5. Number of rooms
6. Number of allocated rooms

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addHotel(&hotelObject); [10]
   // Add hotel details to the structure variable which is passed as an argument
2. displayHotel(&hotelObject); [10]
   // Displays the hotel details from the structure variable passed as an argument
3. findHotelRoom(&hotelObject); [20]
   // Find the name of hotels where 10% of rooms are not allocated

C. Generate a report which display the following information related to the hotel. 20

1. Store the records of 5 star hotel in 5star.dat file, 4 star hotel in 4star.dat file, 3 star hotel in 3star.dat etc. (Use string handling functions)

D. The details of all the hotels should be stored in a file called "hotelData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of hotel and the user defined functions listed above. 05

54. Write a C program to perform following operation on hotel record.

A. Design a structure called Hotel having following data members. 05

1. Hotel name
2. Type of hotel (E.g. 5 star, 4 star etc.)
3. Address (Only area name)
4. Number of deluxe rooms
5. Number of super-deluxe rooms
6. Deluxe rooms (Array variable= 1 for allocated, 0 for not allocated)
7. Super-deluxe rooms (Array variable = 1 for allocated, 0 for not allocated)
8. Manager

B. Design appropriate user defined functions to accomplish following listed tasks 40

4. addHotel(&hotelObject); [10]
   // Add hotel details to the structure variable which is passed as an argument
5. displayHotel(&hotelObject); [10]
   // Displays the hotel details from the structure variable passed as an argument
6. findHotelRoom(&hotelObject); [20]
   // Find the name of hotels having highest numbers of deluxe rooms

C. Generate a report which display the following information related to the hotel. 20

1. Display the name of hotels where all the deluxe rooms or all the super-deluxe rooms are occupied

D. The details of all the hotels should be stored in a file called "hotelData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of hotel and the user defined functions listed above. 05

55. Write a C program to perform following operation on hotel record.

A. Design a structure called Hotel having following data members. 05

1. Hotel name
2. Type of hotel (E.g. 5 star, 4 star etc.)
3. Address (Only area name)
4. Number of deluxe rooms
5. Number of super-deluxe rooms
6. Deluxe rooms (Array variable= 1 for allocated, 0 for not allocated)
7. Super-deluxe rooms (Array variable = 1 for allocated, 0 for not allocated)
8. Manager

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addHotel(&hotelObject); [10]
   // Add hotel details to the structure variable which is passed as an argument
2. displayHotel(&hotelObject); [10]
   // Displays the hotel details from the structure variable passed as an argument
3. findHotelRoom(&hotelObject); [20]
   // Find the total number of empty deluxe rooms and super deluxe rooms of particular hotel.

C. Generate a report which display the following information related to the hotel. 20

1. Display the name of hotels of ahmedabad where 5 deluxe rooms and 3 super-deluxe rooms are empty

D. The details of all the hotels should be stored in a file called "hotelData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of hotel and the user defined functions listed above. 05

56. Write a C program to perform following operation on hotel record.

A. Design a structure called Hotel having following data members. 05

1. Hotel name
2. Type of hotel (E.g. 5 star, 4 star etc.)
3. Address (Only area name)
4. Number of deluxe rooms
5. Number of super-deluxe rooms
6. Deluxe rooms (Array variable= 1 for allocated, 0 for not allocated)
7. Super-deluxe rooms (Array variable = 1 for allocated, 0 for not allocated)
8. Manager

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addHotel(&hotelObject); [10]
   // Add hotel details to the structure variable which is passed as an argument
2. displayHotel(&hotelObject); [10]
   // Displays the hotel details from the structure variable passed as an argument
3. findHotel (&hotelObject); [20]
   // Find the hotel names having identical manager name

C. Generate a report which display the following information related to the hotel. 20

1. Display the name of hotels of Bombay where M deluxe rooms and N super-deluxe rooms are empty
   // The value of M and N should be taken from user and display appropriate hotel details

D. The details of all the hotels should be stored in a file called "hotelData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of hotel and the user defined functions listed above. 05

57. Write a C program to perform following operation on bike record.

A. Design a structure called Item having following data members. 05

1. Bike number (Four digits numbers only)
2. Company (E.g. Bajaj, TVS, Hero Honda etc.)
3. Model (E.g. Pulsor, Victor, Splendor etc.)
4. CC (How many CC bike has)
5. Average (Average of bike)
6. Price
7. Color
8. Number of free services

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addBike(&bikeObject); [10]
   // Add bike details to the structure variable which is passed as an argument
2. displayBike(&bikeObject); [10]
   // Displays the bike details from the structure variable passed as an argument
3. findBike(&bikeObject); [20]
   // Find the bike having CC greater than 150 or average less than 40

C. Generate a report which display the following information related to the bike. 20

1. Display the details of all the bike
   (hint: use for loop and in that call displayBike(&bikeObject) function)
2. Display the details of bike having price above 50000 rs and of non-black color

D. The details of all the bikes should be stored in a file called "bikeData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of bike and the user defined functions listed above. 05

58. Write a C program to perform following operation on bike record.

A. Design a structure called Item having following data members. 05

1. Bike number (Four digits numbers only)
2. Company (E.g. Bajaj, TVS, Hero Honda etc.)
3. Model (E.g. Pulsor, Victor, Splendor etc.)
4. CC (How many CC bike has)
5. Average (Average of bike)
6. Price
7. Color
8. Number of free services

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addBike(&bikeObject); [10]
   // Add bike details to the structure variable which is passed as an argument
2. displayBike(&bikeObject); [10]
   // Displays the bike details from the structure variable passed as an argument
3. findBike(&bikeObject); [20]
   // Find the number of all black pulsor

C. Generate a report which display the following information related to the bike. 20

1. Display the details of all the bike in descending order on the basis of price

D. The details of all the bikes should be stored in a file called "bikeData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of bike and the user defined functions listed above. 05

59. Write a C program to perform following operation on car record.

A. Design a structure called Car having following data members. 05

1. Car number (Four digits numbers only)
2. Company (E.g. Tata, Honda etc.)
3. Model (E.g. Santro, City etc.)
4. Number of seats
5. Average (Average of bike)
6. Petrol/Diesel (If 1 then it is petrol car, and if it is 2 then it is diesel car)
7. Price
8. Color
9. Insurance car

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addCar(&carObject); [10]
   // Add car details to the structure variable which is passed as an argument
2. displayCar(&carObject); [10]
   // Displays the car details from the structure variable passed as an argument
3. findCar(&carObject); [20]
   // Find all the identical model name of white color

C. Generate a report which display the following information related to the car. 20

1. Display the details of all the cars
   (hint: use for loop and in that call displayCar(&carObject) function)
2. Display the name of company having petrol car

D. The details of all the hotels should be stored in a file called "carData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of car and the user defined functions listed above. 05

60. Write a C program to perform following operation on car record.

A. Design a structure called Car having following data members. 05

1. Car number (Four digits numbers only)
2. Company (E.g. Tata, Honda etc.)
3. Model (E.g. Santro, City etc.)
4. Number of seats
5. Average (Average of bike)
6. Petrol/Diesel (If 1 then it is petrol car, and if it is 2 then it is diesel car)
7. Price
8. Color
9. Insurance car

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addCar(&carObject); [10]
   // Add car details to the structure variable which is passed as an argument
2. displayCar(&carObject); [10]
   // Displays the car details from the structure variable passed as an argument
3. findCar(&carObject); [20]
   // Find all the car company name having insurance in HDFC bank (Use string handling function)

C. Generate a report which display the following information related to the car. 20

1. Display the details of car in ascending order on the basis of average

D. The details of all the hotels should be stored in a file called "carData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of car and the user defined functions listed above. 05

61. Write a C program to perform following operation on car record.

A. Design a structure called Car having following data members. 05

1. Car number (Four digits numbers only)
2. Company (E.g. Tata, Honda etc.)
3. Model (E.g. Santro, City etc.)
4. Number of seats
5. Average (Average of bike)
6. Petrol/Diesel (If 1 then it is petrol car, and if it is 2 then it is diesel car)
7. Price
8. Color
9. Insurance car

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addCar(&carObject); [10]
   // Add car details to the structure variable which is passed as an argument
2. displayCar(&carObject); [10]
   // Displays the car details from the structure variable passed as an argument
3. findCar(&carObject); [20]
   // Find the name of those cars, which do not have insurance

C. Generate a report which display the following information related to the car. 20

1. Display the details of car having more than 4 seats and run on diesel

D. The details of all the hotels should be stored in a file called "carData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of car and the user defined functions listed above. 05

62. Write a C program to perform following operation on car record.

A. Design a structure called Car having following data members. 05

1. Car number (Four digits numbers only)
2. Company (E.g. Tata, Honda etc.)
3. Model (E.g. Santro, City etc.)
4. Number of seats
5. Average (Average of bike)
6. Petrol/Diesel (If 1 then it is petrol car, and if it is 2 then it is diesel car)
7. Price
8. Color
9. Insurance car

B. Design appropriate user defined functions to accomplish following listed tasks 60

1. addCar(&carObject); [10]
   // Add car details to the structure variable which is passed as an argument
2. displayCar(&carObject); [10]
   // Displays the car details from the structure variable passed as an argument
3. findCar(&carObject); [40]
   // Display the details of cars whose summation is either 3 or 9.
   (E.g. 1234=1+2+3+4=10=1+0=1 Do not display the car)
   (E.g. 777=7+7+7=21=2+1=3 Display the car)

C. The details of all the hotels should be stored in a file called "carData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

D. Write an appropriate main function to test the structure of car and the user defined functions listed above. 05

63. Write a C program to perform following operation on bank record.

A. Design a structure called Bank having following data members. 05

1. Customer id (5 digits only)
2. First name
3. Last name
4. Bank number
5. Branch name
6. Manager
7. Account number
8. Balance

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addCustomer(&customerObject); [10]
   // Add customer details to the structure variable which is passed as an argument
2. displayCustomer(&customerObject); [10]
   // Displays the customer details from the structure variable passed as an argument
3. findCustomer(&customerObject); [20]
   // Find the name of those customers having account at Maninagar branch irrespective of bank name

C. Generate a report which display the following information related to the customer. 20

1. Display the details of all the customers
   (hint: use for loop and in that call displayCustomer(&customerObject) function)
2. Display the details of bank having more than five customer

D. The details of all the hotels should be stored in a file called "customerData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of customer and the user defined functions listed above. 05

64. Write a C program to perform following operation on bank record.

A. Design a structure called Bank having following data members. 05

1. Customer id (5 digits only)
2. First name
3. Last name
4. Bank number
5. Branch name
6. Manager
7. Account number
8. Balance

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addCustomer(&customerObject); [10]
   // Add customer details to the structure variable which is passed as an argument
2. displayCustomer(&customerObject); [10]
   // Displays the customer details from the structure variable passed as an argument
3. findBank(&customerObject); [20]
   // Find the name of bank having total balance of all customer greater than 500000

C. Generate a report which display the following information related to the customer. 20

1. Display the details of all the customers in ascending order on the basis of account number

D. The details of all the hotels should be stored in a file called "customerData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of customer and the user defined functions listed above. 05

65. Write a C program to perform following operation on bank record.

A. Design a structure called Bank having following data members. 05

1. Customer id (5 digits only)
2. First name
3. Last name
4. Bank number
5. Branch name
6. Manager
7. Account number
8. Balance

B. Design appropriate user defined functions to accomplish following listed tasks 40

1. addCustomer(&customerObject); [10]
   // Add customer details to the structure variable which is passed as an argument
2. displayCustomer(&customerObject); [10]
   // Displays the customer details from the structure variable passed as an argument
3. deposit(&customerObject); [20]
   // Deposit the particular amount to user's account. Make sure user can deposit at most 50000 rs. at a time
4. withdraw(&customerObject);
   // Withdraw the particular amount from user's account. Make sure user must have minimum balance of rs. 1000 in account

D. The details of all the hotels should be stored in a file called "customerData.dat". 10
Use fwrite ( ) and fread ( ) function to store to file and read from file respectively

E. Write an appropriate main function to test the structure of customer and the user defined functions listed above. 05