M3-R4: PROGRAMMING AND PROBLEM SOLVING THROUGH 'C' LANGUAGE

Objective of the Course

The objectives of this course are to make the student understand programming language, programming, concepts of Loops, reading a set of Data, stepwise refinement, Functions, Control structure, Arrays. After completion of this course the student is expected to analyze the real life problem and write a program in 'C' language to solve the problem. The main emphasis of the course will be on problem solving aspect i.e. developing proper algorithms.

After completion of the course the student will be able to

- Develop efficient algorithms for solving a problem.
- Use the various constructs of a programming language viz. conditional, iteration and recursion.
- Implement the algorithms in "C" language.
- Use simple data structures like arrays, stacks and linked list in solving problems.
- Handling File in "C".

Outline of Course

S. No.	Topic	Minimum number of hours
1.	Introduction to Programming	04
2.	Algorithms for Problem Solving	10
3.	Introduction to 'C' Language	04
4.	Conditional Statements and Loops	07
5.	Arrays	06
6.	Functions	06
7.	Storage Classes	03
8.	Structures and Unions	06
9.	Pointers	06
10.	Self Referential Structures and Linked Lists	04
11.	File Processing	04

Lectures	**=**	**60**
Practical/tutorials	**=**	**60**
Total	**=**	**120**

Detailed Syllabus

1. Introduction to Programming **04 Hrs.**

The Basic Model of Computation, Algorithms, Flow-charts, Programming Languages, Compilation, Linking and Loading, Testing and Debugging, Documentation

2. Algorithms for Problem Solving **10 Hrs.**

Exchanging values of two variables, summation of a set of numbers, Decimal Base to Binary Base conversion, Reversing digits of an integer, GCD (Greatest Common Division) of

two numbers, Test whether a number is prime, Organize numbers in ascending order, Find square root of a number, factorial computation, Fibonacci sequence, Evaluate 'sin x' as sum of a series, Reverse order of elements of an array, Find largest number in an array, Print elements of upper triangular matrix, multiplication of two matrices, Evaluate a Polynomial

3. Introduction to 'C' Language **04 Hrs.**

Character set, Variables and Identifiers, Built-in Data Types, Variable Definition, Arithmetic operators and Expressions, Constants and Literals, Simple assignment statement, Basic input/output statement, Simple 'C' programs.

4. Conditional Statements and Loops **07 Hrs.**

Decision making within a program, Conditions, Relational Operators, Logical Connectives, if statement, if-else statement, Loops: while loop, do while, for loop, Nested loops, Infinite loops, Switch statement, structured Programming .

5. Arrays **06 Hrs.**

One dimensional arrays: Array manipulation; Searching, Insertion, Deletion of an element from an array; Finding the largest/smallest element in an array; Two dimensional arrays, Addition/Multiplication of two matrices, Transpose of a square matrix; Null terminated strings as array of characters, Standard library string functions

6. Functions **06 Hrs.**

Top-down approach of problem solving, Modular programming and functions, Standard Library of C functions, Prototype of a function: Formal parameter list, Return Type, Function call, Block structure, Passing arguments to a Function: call by reference, call by value, Recursive Functions, arrays as function arguments.

7. Storage Classes **03 Hrs.**

Scope and extent, Storage Classes in a single source file: auto, extern and static, register, Storage Classes in a multiple source files: extern and static

8. Structures and Unions **06 Hrs.**

Structure variables, initialization, structure assignment, nested structure, structures and functions, structures and arrays: arrays of structures, structures containing arrays, unions

9. Pointers **06 Hrs.**

Address operators, pointer type declaration, pointer assignment, pointer initialization, pointer arithmetic, functions and pointers, Arrays and Pointers, pointer arrays, pointers and structures, dynamic memory allocation.

10. Self Referential Structures and Linked Lists **04 Hrs.**

Creation of a singly connected linked list, Traversing a linked list, Insertion into a linked list, Deletion from a linked list

11. File Processing **04 Hrs.**

Concept of Files, File opening in various modes and closing of a file, Reading from a file, Writing onto a file

RECOMMENDED BOOKS

MAIN READING

1. Byron S Gottfried “Programming with C” Second edition, Tata McGrawhill, 2007 (Paper back)
2. R.G. Dromey, “How to solve it by Computer”, Pearson Education, 2008.
3. Kanetkar Y, “Let us C”, BPB Publications, 2007.
4. Hanly J R & Koffman E.B, “Problem Solving and Programm design in C”, Pearson Education, 2009.

SUPPLEMENTARY READING

1. E. Balagurusamy, “Programming with ANSI-C”, Fourth Edition,2008, Tata McGraw Hill.
2. Venugopal K. R and Prasad S. R, “Mastering ‘C’”, Third Edition, 2008, Tata McGraw Hill.
3. B.W. Kernighan & D. M. Ritchie, “The C Programming Language”, Second Edition, 2001, Pearson Education
4. ISRD Group, “Programming and Problem Solving Using C”, Tata McGraw Hill,2008.
5. Pradip Dey , Manas Ghosh, “Programming in C”, Oxford University Press, 2007.

M3-R4: PROGRAMMING AND PROBLEM SOLVING THROUGH 'C' LANGUAGE

Model Question Paper

NOTE:

1. There are **TWO PARTS** in this Module/Paper. **PART ONE** contains **FOUR** questions and **PART TWO** contains **FIVE** questions.

2. **PART ONE** is to be answered in the **TEAR-OFF ANSWER SHEET** only, attached to the question paper, as per the instructions contained therein. **PART ONE** is **NOT** to be answered in the answer book.

3. Maximum time allotted for **PART ONE** is **ONE HOUR**. Answer book for **PART TWO** will be supplied at the table when the answer sheet for **PART ONE** is returned. However, candidates, who complete **PART ONE** earlier than one hour, can collect the answer book for **PART TWO** immediately after handing over the answer sheet for **PART ONE**.

TOTAL TIME: **3 HOURS** **TOTAL MARKS: 100**
(PART ONE - 40; PART TWO - 60)

PART ONE
(Answer ALL Questions; each question carries ONE mark)

1. Each question below gives a multiple choices of answers. Choose the most appropriate one.

1.1 The programming Language C happens to be

a) An Assembly Level Language.
b) A High Level Language with some Assembly Level Language Features.
c) A Programming Language used only to write System Software.
d) A Programming Language used for developing Application Packages only.

1.2 The C declaration int I_a; implies

a) The variable I_a is a signed Binary Integer .
b) The variable I_a is an Unsigned Decimal Integer.
c) The variable I_a is an signed Hexadecimal Integer.
d) The variable I_a is a signed Integer that can be expressed in any Base.

1.3 The C statement printf ("The Value =%x",62); will print

a) The Value= 62
b) The Value = O62
c) The Value= OX 3C
d) The Value= 3C

1.4 In the following C declaration

```
float F_C = 12.5;
void  VF_A (int);
```

```
int  main();    { /* begin main */
            float F_B;  F_C = 13.5;
        ……
          return (0); }/* end main */
```

a) The Variable F_C is GLOBAL to both the functions main () as well as VF_A.
b) The Variable F_C is LOCAL to the function main();
c) The Variable F_C is LOCAL to the function VF_A.
d) The Variable F_C is EXTERNAL.

1.5 Consider the following C Program .

```
# define  S 10+2

#include <stdio.h>

 int main()
                    { /* begin main */
                         int  Result  = S + S ;

                        printf ("\n\n Result = %d\n\n", Result ); /* Output Line #2 */

                         return (0);
                     } /* end main*/
```

The Output generated by the above C Program will be

a) Result = 10
b) Result = 12
c) Result = 24
d) Result = 20

1.6 What will be the Output generated by the following C Program ?

```
#include <stdio.h>
 int main()
                    { /* begin main */

                        int I_C ; float F_D , F_E;

                        I_C = 5/2 ; F_D = 5/2 ; F_E = 5/2.0;

                        printf ("\n I_C = %d  F_D =%f  F_E = %f \n\n", I_C,F_D,F_E);

                         return (0);

                     } /* end main*/
```

a) I_C= 1 F D = 2.0 F_E = 2.5
b) I_C= 2 F D = 2.0 F_E = 2.5
c) I_C= 2 F D = 2.5 F_E = 2.0

d) I_C= 2 F D = 2.5 F_E = 2.5

1.7 In C Functions the actual expressions / parameters are passed on to Formal parameters using the method of :

a) Call by reference.
b) Call by Value Result.
c) Call by Value.
d) Call by Name.

1.8 Consider the following C program segment :

```
typedef  struct  Point
                { float  F_x;
float  F_y;
}Point_T;
typedef  struct  Circle
{ float  F_Radius;
  Point_T  R_Center;
}  Circle_T;
int  main();
{  //  begin  main
Point_T  R_Point;  Circle_T  R_Circle;
/*  Circle  Manipulation  Statements  */
return(0);
}  //  end  main
```

To manipulate a circle which of the following set of assignment statements will have to be used ?

a) R_Circle.F_Radius = 10.2; R_Circle.R_Center.F_x = 2.0 ; R_Center.F_y=3.0;
b) R_Circle.F_Radius = 10; R_Circle..F_x = 2.0 ; R_Circle.F_y=3.0;
c) R_Circle.F_Radius = 10.2; R_Circle.R_Center.F_x = 2.0 ;
d) R_Circle.R_Center.F_y=3.0;
e) R_Circle.F_Radius = 10.2; R_Circle.F_x = 2.0 ; R_Circle.F_y=3.0;

1.9 In the following C Declaration

```
#define CUI_Size 10
  typedef int AI_1D_01_T [CUI_Size];

int main()
    {/* begin main */
        AI_1D_01_T  AI_1D_A;
```

The variable AI_1D_A represents

a) An array of Integers of any size.
b) An array of Integers having minimum 10 integers.
c) An array of Integers having Maximum 10 Integers.
d) None of the above.

1.10 Consider the following C Code

```
#include <stdio.h>
#include <stdlib.h>
int main ()
      {/*begin main */
            int I_X=6; int *PI_Y;
PI_Y = (int*) malloc (sizeof (int));
*PI_Y = I_X;
printf(" *PI_Y =%d",*PI_Y);
*PI_Y = 7;
printf (" I_X = %d",I_X);
return(0);
} // end main
```

Which, among the following will it produce as output ?

a) *PI_Y = 7 I_X = 6
b) *PI_Y = 6 I_X = 7
c) *PI_Y = 7 I_X = 6
d) *PI_Y = 6 I_X = 6

2. Each statement below is either TRUE or FALSE. Identify and mark them accordingly in the answer book

2.1 In C %x format can be used for Inputting signed Octal Integers (FALSE).
2.2 A Pointer variable content will be the Address of the variable it points to. (TRUE).
2.3 In C , a SINGLE scanf () can be used to read in the values of any number of pre-declared variables (TRUE).
2.4 Arrays in C are always stored in Column Major fashion (FALSE).
2.5 ! operator is a BINARY Operator in C. (FALSE).
2.6 Recursive functions provide an elegant way of representing recurrences (TRUE).
2.7 Array represents a homogeneous Data Structure (TRUE).
2.8 A structure cannot be a member of an Union in C (FALSE).
2.9 In C *p++ increments the content of the location pointed to by p (TRUE).
2.10 A C Function can return a whole structure as it's value (TRUE).

3. Match words and phrases in column X with the nearest in meaning in column Y.

	X		**Y**
3.1	Premature exit from within a C Loop	a)	1 Byte.
3.2	Character variable will have a size of	b)	Indentation is essential
3.3	A C Function that do not return a value will be having	c)	Call by Reference.
3.4	A string in C is terminated by	d)	To open a file for writing after discarding it's previous content
3.5	To understand the Blocks of C	e)	An Integer type
3.6	Multiway branching in C can be	f)	A void type

implemented

3.7 All variables declared inside a function
3.8 A Pointer Parameter in a C Function simulates
3.9 A Linked List represents
3.10 In C fopen "w" mode is used

g) Are Local to that function
h) Opening a file in Read mode , retaining the previous content
i) A white space character.
j) 4 Bytes
k) A '\0' Charcter
l) A dynamic Data Structure
m) Using switch – case statement
n) Can be achieved by break statement

4 Fill in the blanks in 4.1 to 4.10 below, by choosing appropriate words and phrases given in the list below:

(a) Dividing	(b) One or ZERO	(c) CPU Register	(d) extern
(e) Optional	(f) Randomly	(g) At least once	(h At run time
(i) Linked List	(j) An Array	(k) Fields	

4.1 The Operator I_Value >> 2 is equivalent to _____ I_Value by 4 .
4.2 The Declaration reg int IReg_C will allocate a _____ for the variable IReg_C.
4.3 On executing f = ! (K >10) f will have a value _____
4.4 The individual Elements of any Array can be accessed _____
4.5 The else portion of an if else statement in c is _____
4.6 In C the body of do-while loop will be executed _____
4.7 Any variable starting with _____ in the declaration will be treated as an External variable

4.8 In C a polynomial of the form 100 M 34 – 20M + 10 can be efficiently represented by a _____
4.9 The Components of a Records are termed as _____
4.10 In C any dynamic data structure is created _____ .

PART TWO
(Answer ANY FOUR questions)

5. Consider the following C program Outline that DOES NOT USE any Structured Data Type like ARRAY or STRUCTURE or POINTER whatsoever ANYWHERE

```
:
#include  <stdio.h>
#include  <math.h>


/*  NO  OTHER  LIBRARY  CAN  BE  USED*/


#define  CI_Max  9999
```

```
#define CI_Min -9999
/* NO OTHER USER DEFINED CONSTANTS, DATA TYPES OR GLOBALS
CAN BE USED*/
/* User Defined Function Prototypes. NO OTHER FUNCTIONS are used */
void VF_Read_Int ( int, int, int*); /* READS and Returns an Integer through
it's pointer parameter provided it lies between a specific range passed as the
other two parameters . If the value read in within the happens to be
OUTSIDE this range, it will continue to loop & print the message Input
OUT of range ,

Give Again and wait for a proper value to be inputted by the user. */

int IF_Test_Prime (int) ; /* Used to Test MOST EFFICIENTLY whether the
Integer passed as it's only parameter happens to be Prime or Not. It Returns
1 if the passed Integer is prime returns 0 if it is Non Prime. In each case ,
it prints an appropriate message within it */

int main ()
{//begin main int I_Value;
/* You May Employ other Simple Variables */ VF_Read_Int (CI_Max, CI_Min,
&I_Value);
/* Reads in an Integer Value within a Specified Range */
VF_Print_NON_Prime_Factors (I_Value);
/* Displays all the NON Prime Factors of the value I_Value*/
return(0);
}//end main
```

a. Frame the body of the function

VF_Read_Int

. The Function heading is as illustrated below :

```
void VF_Read_Int (int I_High, int I_Low, int *PI_X)
/*  READS and Returns an Integer through it's pointer parameter
provided it lies between a specific range passed as the other two
parameters . If the value read in within the happens to be OUTSIDE
this range, it will continue to loop print the message {\bf Input OUT of
range , Give Again and wait for a proper value to be inputted by
the user. NO OTHER PARAMETER CAN BE USED. */
```

b. Frame the body of the function

IF_Test_Prime

. The Function heading is as illustrated below :

```
int IF_Test_Prime ( int I_Num) /* Used to Test MOST EFFICIENTLY
whether the Integer passed as it's only parameter happens to be prime
or Not. It Returns 1 if the passed Integer is prime returns 0 if it is
Non Prime. In each case , it prints an appropriate message within it */
```

(6+9)

6. Consider two integer data files F1 and F2 having following features.

a. Number of data values (key) in each file is unknown and the files may be of different sizes.
b. The values / Integer Keys in both the files F1 & F2 are Sorted in Descending Order.
c. Same data (key) can appear more than once in F1 or F2.
d. F1 and F2 may share common data values i.e. same key item may appear both the files .

Write a C function to merge the two files F1 and F2 to form a third file F3 having the following features.

-Elements in F3 are sorted in ascending order.
- Duplicate entries are not permitted (i. e. ,No element appears more than once).

(15)

7. The following operations are defined on a sorted Doubly linked list of Integers L where elements are arranged in Descending order from left.

INSERT (L,X) : Insert the integer X in the list L if X is not present.

DELETE (L,X) : Delete the integer X from the list L (if it exists).

SHOW-MID (L) : Print the n/2 th element of the list from left where n is the Number of elements in the current list and we use integer Division where 5/2 = 2

Frame C functions to implement each of the above functions INSERT (L,X) , DELETE (L,X) and SHOW_MID(L)

(6+6+3)

8.

a. In 2 (two) dimensions, a point can be described by its two coordinates namely X & Y both of which can be real numbers. A line can be described in the following manner :

(i) The co-ordinates of its two end points (X1, Y1) & (X2, Y2)
(ii) Its gradient 'm' & intersection 'c' (in the form Y = mx + c)
(iii) The length of the line is also stored along with.

Specify appropriate data types to store a point as well as a line in C.

(1+2)

b. Write a C function Point_to_Line (P1, P2) that will accept as parameters the coordinates of two points P1 & P2 and return a line that has the aforesaid 2 points as its end points.

(5)

c. A quadrilateral can be described by a sequence of 4(four) lines such that one end point of one line happens to be the starting point of the next line. Specify a suitable data structure in C to represent a quadrilateral. [2]

(2)

d. Write a C function that will accept a quadrilateral as a parameter and classify it whether it is a
[2+2+3=7]
A Square.
A Rhombus.
A Rectangle.

in each case it computes the perimeter as well.

(2+2+3)

9.

a. Write a single Recursive C function to generate the n th Fibonacci number Fib(n) (n being a +ve non zero integer) . You cannot use any array, global variables and/or additional parameters/functions. Trace out the Call & Return sequences along with return values clearly by a schematic diagram when your function Fib(n) is invoked from main() with n = 6. Also mention the TOTAL no. of times any Fib(n) is called for each value of n for invoking Fib(6) from main(), e.g. Fib(2) is called a total of 4 times etc.

(2+5+2)

b. What will be the value of A(1, 3) if A(m, n) happens to be defined in the following manner? Specify each computation step in detail .

$A(0, n) = n + 1$ for $n \geq 0$
$A(m, 0) = A(m - 1, 1)$ for $m > 0$
$A(m, n) = A(m - 1, A(m, n - 1))$ for $m, n > 0$

(6)

Assignment 1.

Write a program to find sum of all prime numbers between 100 and 500.

Assignment 2.

Write a program to obtain sum of the first 10 terms of the following series for any positive integer value of X :

$X + X^3/3! + X^5/5! ! + X^7/7! + \ldots$

Assignment 3.

Write a program to reverse the digits of a given number. For example, the number 9876 should be returned as 6789.

Assignment 4.

Write a program to compute the wages of a daily laborer as per the following rules :-

Hours Worked	Rate Applicable
Upto first 8 hrs	Rs 50/-
For next 4 hrs	Rs 10/- per hr extra
For next 4 hrs	Rs 20/- per hr extra
For next 4 hrs	Rs 25/- per hr extra
For rest	Rs 40/- per hr extra

Accept the name of the laborer and no. of hours worked. Calculate and display the wages. The program should run for N number of laborers as specified by the user.

Assignment 5.

Write a program to input 20 arbitrary numbers in one-dimensional array. Calculate Frequency of each number. Print the number and its frequency in a tabular form.

Assignment 6.

Define 2 dimensional array a (3,3), b(3,3),sum(3,3),diff(3,3),mult(3,3). Store 9 arbitrary numbers in a(3,3) and 9 arbitrary numbers in b(3,3). Do the following:

a) Calculate sum of a(3,3) and b(3,3) and store in sum(3,3) where sum(i,j)=a(i,j)+b(i,j)
b) Calculate difference of a(3,3) and b(3,3) and store in diff(3,3) where diff(i,j)=a(i,j)-b(i,j)
c) Calculate product of two arrays a(3,3) and b(3,3) and store in mult(3,3) where mult(i,j)= summation of a(i,k)*b(k,j) over k where k=1 to 3.

Print the result in a tabular form

Assignment 7.

Write a function, str_search(char* s1,char* s2, int n) , that takes two strings and an integer, as arguments and returns a pointer to the n^{th} occurrence of 1^{st} string s1 in 2^{nd} string s2, or NULL if it is not present.

Assignment 8.

Write a C function to remove duplicates from an ordered array. For example, if input array contains 10,10,10,30,40,40,50,80,80,100 then output should be 10,30,40,50,80,100.

Assignment 9.

Apply recursive call to do the following:

(i) Input 'n'(1-200). Calculate sum of 'n' numbers.

(ii) Input 'n'(1-20). Calculate product of 'n' numbers.

(iii) Input 'n'(2-20). Print 'n' number of Fibonacci numbers. In Fibonacci sequence the sum of two successive terms gives the third term. The following are few terms of Fibonacci sequence :-

1 1 2 3 5 8 13 ……

Assignment 10.

Write a program which will arrange the positive and negative numbers in a one-dimensional array in such a way that all positive numbers should come first and then all the negative numbers will come without changing original sequence of the numbers.

Example:

Original array contains: 10,-15,1,3,-2,0,-2,-3,2,-9

Modified array: 10,1,3,0,2,-15,-2,-2,-3,-9

Assignment 11.

Write a menu driven program to maintain a Telephone Directory having following file structure:

1. Name : Character type : Length =20 characters.
2. Address : Character type : Length =40 characters.
3. Phone: Character type : Length =12 characters.

Menu

1. Add record(s)
2. Display record(s)
3. Search record(s)
4. Modify record(s)
5. Delete record(s)
6. Backup copy of File
7. Exit

Type your choice= 1,2,3,4,5,6,7— ->

Assignment 12.

Write a program to extract words form any text file and store in another file. Sort the words in alphabetical order and store them in the same file. Read the sorted file and print the frequency of each word.

Assignment 13.

Write a program to remove all occurrences of word “the” and “The” from an input string. For example

Input : The Dhillon Theatre is now the Fun Republic.
Output : Dhillon atre is now Fun Republic.

Assignment 14.

Write a program to display the Following pattern called Floyed’s Triangle.

```
1
2  3
4  5  6
7  8  9  10
11 12 13 14 15
```

Assignment 15.

Write a program that accepts an input integer ‘n’ in the range 3-9 inclusive, and display the following pattern on a cleared screen.

Sample input for n=3
Sample output

```
  3
 3 2 3
3 2 1 2 3
 3 2 3
  3
```

Sample input for n=4
Sample output

```
   4
  4 3 4
 4 3 2 3 4
4 3 2 1 2 3 4
 4 3 2 3 4
  4 3 4
   4
```

Assignment 16.

Write a program to count the vowels in free text given as standard input. Read text one character at a time until you encounter end-of-data. Then print out the number of occurrences of each of these vowels.

Assignment 17.

Write a program to copy one file to another such that every word is reversed before being written to the target file. Assume the maximum size of each word is 10 characters and each word is separated either by new line(s), tab(s) or space(s). For example, if source file contains “I am an Indian”, the target file should contain “I ma na naidnI”.

Assignment 18.

Define a structure for an Employee having EmployeeName, EmployeeCode, BasicPay, DearnessAllowance, HRA, PF, GrossPay, NetPay Take an array of 10 Employees. Write 'C' functions to :-

a) Accept data for EmployeeName, EmployeeCode, BasicPay for all the employees.
b) Compute :-
    a. DearnessAllowance = 50% of BasicPay
    b. HRA = 20% of BasicPay + DearnessAllowance
    c. PF = 12% of BasicPay + DearnessAllowance
    d. GrossPay = BasicPay + DearnessAllowance + HRA
    e. NetPay = GrossPay – PF
c) Display the name of employee who has highest GrossPay.
d) Compute and display average net pay.
e) Display list of all employees in the alphabetical order of employee name.

Assignment 19.

Write a program to convert a given decimal number to its binary equivalent and vice versa.

Assignment 20.

Input any positive integer number (n<=9999999). Convert the number into words.

Assignment 21.

a) Define a structure of a node of a linked list having an integer data member x.
b) Use the above structure in (a) and write the functions for the following parts;

i) a function which takes a pointer to the head of linked list, which is in ascending order and an integer, x to be inserted in the linked list, as arguments. The node must be inserted in such a way that the linked list remains in ascending order after insertion.

ii) a function which takes a pointer to the head of a linked list and an integer, x to be removed from the linked list, as arguments. If x is not found in the linked list, then it should display an appropriate message.

Assignment 22.

Write a program to replace 'a' with 'b', 'b' with 'c',….,'z' with 'a' and similarly for 'A' with 'B','B' with 'C', …., 'Z' with 'A' in a file. The other characters should remain unchanged.

Assignment 23.

Write a function char* stuff(char* s1,char* s2,int sp, int rp) to stuff string s2 in string s1 at position sp, replacing rp number of characters (rp may be zero).

Assignment 24.

Write a program to display the content of a Text file which means it will behave like TYPE command of MSDOS. Suppose the name of your program file: FILETYPE.C and FILETYPE.EXE and the name of the source file is MYFILE.TXT. The following command should work: C: \PROGRAM> FILETYPE MYFILE.TXT

Assignment 25.

Write a program to input name, address and telephone number of ‘n’ persons (n<=20). Sort according to the name as a primary key and address as the secondary key. Print the sorted telephone directory.