# COMPUTER PROGRAM ON DRIS, MDRIS AND CND

- BIVARIATE AND MULTIVARIATE ANALYSES TOOLS FOR MONITORING THE SOIL AND PLANT NUTRIENT IMBALANCES

Senthil-Kumar Selvaradjou, Luca Montanarella and Aruna Geetha

EUROPEAN COMMISSION
DIRECTORATE-GENERAL
Joint Research Centre

Institute for Environment and Sustainability

2005

EUR 21505 EN

**This document may be cited as follows:**

Selvaradjou, S-K, Montanarella, L and Aruna-Geetha. (2005). Computer Program on DRIS, MDRIS and CND – Bivariate and Multivariate Analyses Tools for Monitoring the Soil and Plant Nutrient Imbalances. EUR 21505 EN, 49 pp. Office for Official Publications of the European Communities, Luxembourg.

# COMPUTER PROGRAM ON DRIS, MDRIS AND CND

- BIVARIATE AND MULTIVARIATE ANALYSES TOOLS FOR MONITORING THE SOIL AND PLANT NUTRIENT IMBALANCES

Senthil-Kumar Selvaradjou[1], Luca Montanarella[1] and Aruna Geetha[2]

[1] Institute of Environment & Sustainability, Joint Research Centre, Ispra, Italy
[2] Tamil Nadu Agricultural University, Coimbatore, India

EUROPEAN COMMISSION
DIRECTORATE-GENERAL
Joint Research Centre

Institute for Environment and Sustainability

2005

EUR 21505 EN

## MISSION OF THE INSTITUTE FOR ENVIRONMENT AND SUSTAINABILITY

The mission of the Institute of Environment and Sustainability is to provide scientific and technical support to EU strategies for the protection of the environment and sustainable development. Employing an integrated approach to the investigation of air, water and soil contaminants, its goals are sustainable management of water resources, protection and maintenance of drinking waters, good functioning of aquatic ecosystems and good ecological quality of surface waters.

## LEGAL NOTICE



A great deal of additional information on the
European Union is available on the Internet.
It can be accessed through the Europa server
(http://europa.eu.int)

EUR 21505 EN



*Printed in Italy*

# CONTENTS

CHAPTER - I

# INTRODUCTION

The intensive cropping system pushing up the agricultural output level parallel with the present demographic transition imparts a cruel attack on the scarce and precious soil resources. The path of agricultural dynamism has closed the eye towards soil quality, which would challenge the agricultural production pertaining to human survival, is not a distant future. The unprincipled exploitation of soil resource has threatened soil quality, which is moving forward to touch the barren point. The need of the hour is not only to exploit the soil resource to fulfill the growing demand for food but also to sustain and conserve it. The present pace of soil degradation and environmental concerns of high input intensive agriculture are the major issues pertaining to soil fertility management and in developing strategies for sustainable agriculture through integrated and balanced nutrient management.

Indiscriminate use of unbalanced NPK application aggravates the micronutrient disorders, which act additively along with other biotic and abiotic stresses to limit the crop production. According to the law of minimum which governs maximizing the returns for the inputs, the largest response to a given input comes where there is no other limiting factor and the magnitude of response will increase as more and more limiting factors are corrected. Developing suitable management strategies based on these needs would help us in exploiting the full potentials of the soil by scooping out the obstacle of nutrient imbalance in the crops without spoiling the soil health.

In this context, the most popular nutrient diagnostic model of Beaufils (1973) with the acronym "DRIS" (Diagnosis and Recommendation Integrated System) and the most recent Compositional Nutritional Diagnosis model of Parent et al (1992) serves the purpose to diagnose the nutritional imbalance provoked inside the crop plant due to improper nourishment of the soil. But, the main limitations of these analysis methods were the involvement of extensive and voluminous rigorous computational steps. Due to the advancement in the computer software and hardware resources in the present era, calculations involving more than 4-5 nutrients in DRIS and CND which were considered to be Herculean task in the past, is now transformed to be comparatively simple and feasible. The computer programs for calculating Diagnosis and Recommendation Integrated System (DRIS), Modified- Diagnosis and Recommendation Integrated System (MDRIS) and Compositional Nutritional Diagnosis (CND) approaches were developed in Microsoft VISUAL FOXPRO – 6.0. Using these programs and carefully following the guidelines given in this book, the nutrient norms and indices upto 12 nutrients can be calculated in easy steps.

CHAPTER - II

# DRIS Model

## (DIAGNOSIS AND RECOMMENDATION INTEGRATED SYSTEM)

## i. Brief description of methodology - DRIS and MDRIS models

DRIS / MDRIS (Diagnosis Recommendation Integrated System (DRIS) of Beaufils, (1973) / Modified Diagnosis Recommendation Integrated System (M-DRIS) of Beaufils, (1973)) provides a means of ordering nutrient ratios or products into meaningful expressions called DRIS / MDRIS indices. Essentially, a nutrient index is a mean of the deviations of the ratios constraining a given nutrient from their respective optimum or norm values. The first step in implementing DRIS / MDRIS is the establishment of these standard values or norms. This is done using a survey data in which yield data are collected from cropping enterprise and nutrients concentration from the plant analysis data (index tissue of the plant) in order to build up a data bank representative of the crops.

Using yield and plant tissue nutrient concentration from the survey data, DRIS norms and coefficients of variations (CVs) are derived according to the procedure by Walworth and Sumner (1987). The statistical Critical Value Approach (CVA) of Cate and Nelson (1971) is used to derive the cut off for the high yielding and low yielding populations. Mean values for each nutrient expression together with their associated CVs and variances are then calculated for the two populations. The mean values (high yielding population) of nutrient expressions are ultimately chosen as diagnostic norms. The selection of nutrient ratio expression values with relatively large variance ratios (variance of low yielding population / variance of high yielding population) were done. DRIS indices are calculated for nutrients A – N using the following generalized equations:

$$\text{A index} = \frac{f(A/B)+f(A/C)+f(A/D)+....+f(A/N)}{Z}$$

$$\text{B index} = \frac{f(A/B)+f(B/C)+f(B/D)+....+f(B/N)}{Z}$$

$$\text{N index} = \frac{f(N/B)+f(N/C)+f(N/D)+....+f(N/M)}{Z}$$

where

$$f(A/B) = \left(\frac{A/B}{a/b}-1\right)x\frac{1000}{CV} \quad \text{when } A/B > a/b$$

$$f(A/B) = \left(1-\frac{a/b}{A/B}\right)x\frac{1000}{CV} \quad \text{when } A/B < a/b$$

in which A/B is the value of the ratio of the two element in the tissue under diagnosis and a / b is the value of the corresponding norms, Z is the number of functions and CV is the coefficient of variation associated with each nutrient ratio norm a / b – a / n. In the case of MDRIS the yield is included as one of the nutrient parameter and is attached to the denominator in the expression for calculation of functions to be used for indices calculation.

## ii. Algorithm for DRIS and MDRIS nutrient models

SET ENVIRONMENT

INPUT VARIABLE DECLARATION

GET - Input file(source file : EXCEL input FILE), Sample number, no of parameters/nutrients studied, Enter output file

CREATE database tables

DECLARE ARRAYS wholefile(sample number, total parmeters +1), class sum of square (sample no), class sum of square1 (sample no), class sum of square2 (sample no), class sum of square3 (sample no), class sum of square4 (sample no)....

DECLARE Memory variables

USE whole data file

APPEND FROM input file

COPY TO ARRAY whole file [sample number, total parmeters +1]

SORT ARRAY -descending order of yield data e.g. (ASORT (wholefile, 1, -1,1))

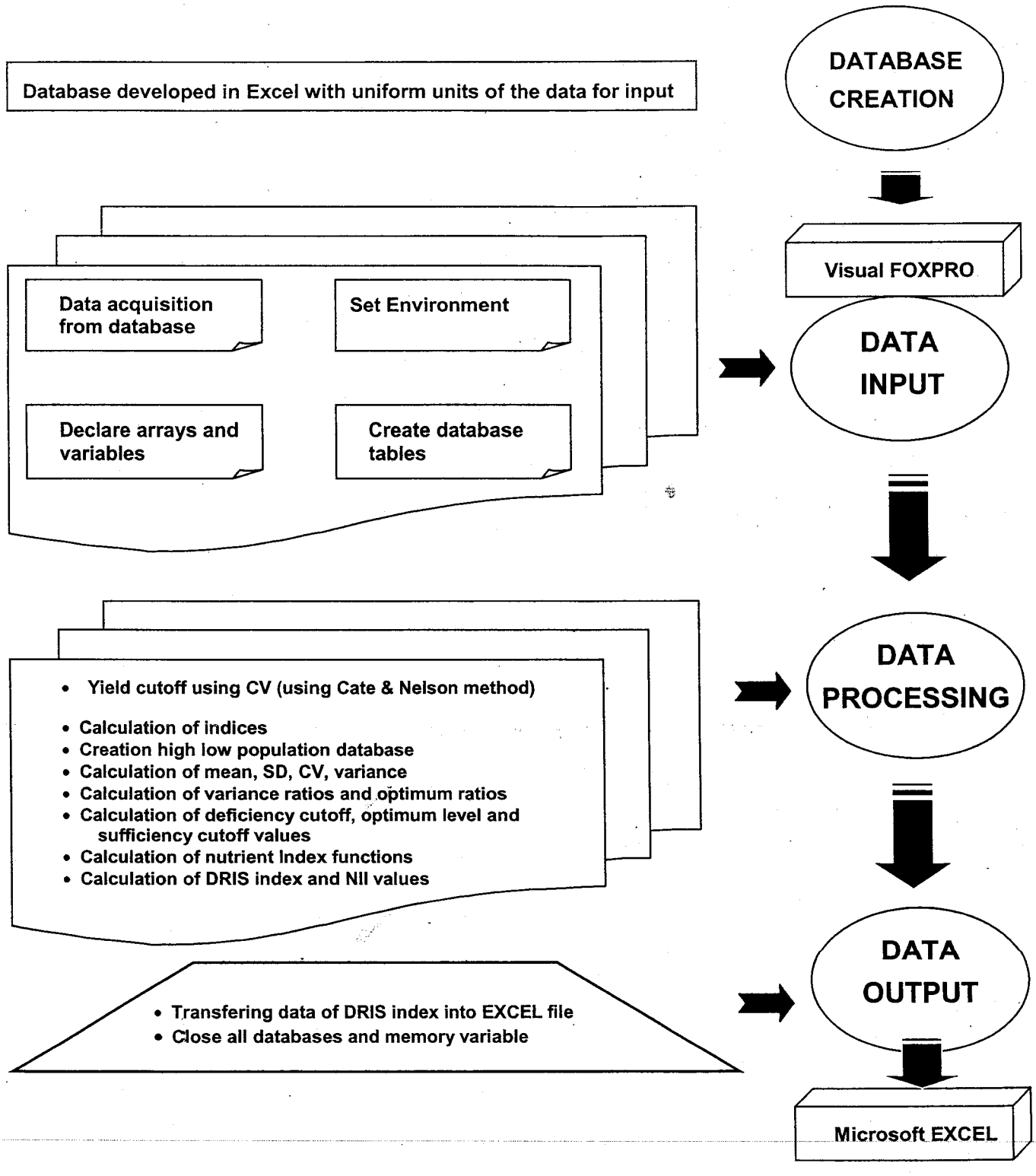
APPEND FROM ARRAY whole file database TO whole file array

- Critical value calculation for yield cut off using Cate and Nelson's statistical class sum of square technique
- Creation of high and low population nutrient data arrays based on the cut off derived from above method
- Calculation for both combinations of various nutrient ratios for high and low populations separately in arrays
- Calculation of mean, variance, CV for both combinations of nutrient ratios for high and low population data separately in arrays
- Calculation of variance ratios between low population and high population files and selection of opt ratio for indices calculation
- Sending the parameters like mean, CV, Variance, Variance ratios, choice of ratio for indices
- Calculation, deficiency cut off, low level cut off, optimum level cut off, and high or sufficient level cutoff for both high and low population file
- Calculation of index functions for various nutrient ratios for both high and low population files
- Calculation of DRIS index value based on the index function, NII for both high and low population in arrays
- Transferring data from array to subsequent DRIS index EXCEL files

CLOSE ALL DATABASES

CLEAR ALL memory variables

iii. Flow chart of the computer program – DRIS model



## iv. Computer program - DRIS model

```
*****************************************************************************************
                                SET ENVIRONMENT
*****************************************************************************************
CLEAR; CLEAR ALL; CLOSE ALL; SET SCOREBOARD OFF; SET STATUS OFF; SET TALK OFF: SET CONFIRM OFF; SET SAFETY OFF
CLOSE ALL
CLEAR

F1 = SPACE(20); T1=SPACE(20) ; T2=SPACE(20) ; XX1=SPACE(40)
STORE 0 TO SAMPLENO, PARANO, INDNO

    @ 7, 10 SAY "ENTER INPUT FILE NAME" FONT 'COMIC SANS MS', 13 STYLE 'S'
    @ 7, 50 GET F1 FONT 'COMIC SANS MS', 13 STYLE 'S'
    @ 11,10 SAY "ENTER TOTAL NO. OF SAMPLES" FONT 'COMIC SANS MS', 13 STYLE 'S'
    @ 11, 50 GET SAMPLENO FONT 'COMIC SANS MS', 13 STYLE 'S'
    @ 15, 10 SAY "ENTER NO. OF PARAMETERS" FONT 'COMIC SANS MS', 13 STYLE 'S'
    @ 15, 50 GET PARANO FONT 'COMIC SANS MS', 13 STYLE 'S'
    @ 19, 10 SAY "ENTER OUTPUT FILE NAME" FONT 'COMIC SANS MS', 13 STYLE 'S'
    @ 19, 50 GET XX1 FONT 'COMIC SANS MS', 13 STYLE 'S'
    READ

T1= ALLTRIM(XX1)+"DMAINSORT"
T2=ALLTRIM(XX1)+"DDRISLOW"

    CREATE TABLE &T1 (Y N(6,2), N N(9,4), P N(9,4), K N(9,5), CA N(9,5), MG N(9,5), NA N(9,5), S N(9,5), B N(9,5), ZN N(9,5), CU N(9,5), FE N(9,5), MN N(9,5), FNO
                        N(4))
    CREATE TABLE &T2 (N N(9,4), P N(9,4), K N(9,4), CA N(9,4), MG N(9,4), NA N(9,4), S N(9,4), B N(9,4), ZN N(9,4), CU N(9,4), FE N(9,4), MN N(9,4), TOT
                        N(9,4),FNO N(4))
    DECLARE DAT1 [SAMPLENO, PARANO+2], CSS1[SAMPLENO], CSS2[SAMPLENO], CSS[SAMPLENO], CSS3[SAMPLENO], CSS4[SAMPLENO], R2[SAMPLENO]
    DECLARE TX1[SAMPLENO], TX2[SAMPLENO]
    STORE 0 TO SS1, SS2, SS3, SS4, SS9, SS10, CAL1, I, LL1, XX2, AA3, AA4, XX5, AA5, AA6, AA7, AA9, ZZZ, MAX1
    STORE 0 TO MAX2, MAX3, MAX4, MAX5, LOWPOP, HIGHPOP, D, XXX, YYY, I, J, KKK, AAA, BBB, Z, FFF, GGG
    STORE 0 TO SSS1, SUM1, MEAN1, VARR1, AAAA, BBBB, DRISTEMP1, DRISTEMP2, AA, BB
    STORE 0 TO TT, CUT, SS, K1, X, Z1, G, H, X1,L1, TOTD1, TOTD
SELECT 1
        USE &T1 ALIAS X1   ZAP
        APPEND FROM &F1 DELIMITED WITH TAB
        COPY TO ARRAY DAT1
        ASORT (DAT1, 1, -1,1)
        CLEAR; ZAP
        APPEND FROM ARRAY DAT1
CLOSE ALL
SELECT 2
        USE &T2 ALIAS X14; ZAP
*****************************************************************************************
                    CV CALCULATION AND SELECTION OF CUT OFF FOR POPULATION
*****************************************************************************************
TT=0; SS=0
        FOR I=1 TO SAMPLENO-1
                TT=TT+DAT1[I,1] ; SS=SS+(DAT1[I,1]*DAT1[I,1]); TX1[I]=TT; CSS1[I]=SS
        ENDFOR
        TT=TT+DAT1[SAMPLENO,1]; SS=SS+DAT1[SAMPLENO,1]*DAT1[SAMPLENO,1]
        TX1[SAMPLENO]=TT; CSS1[SAMPLENO]=SS
        TSS=CSS1[SAMPLENO]-TX1[SAMPLENO]*TX1[SAMPLENO]/SAMPLENO

        FOR I=2 TO SAMPLENO-1
```

```
                TX2[I]=TX1[SAMPLENO]-TX1[I]; CSS2[I]=CSS1[SAMPLENO]-CSS1[I]
                CSS1[I]=CSS1[I]-TX1[I]*TX1[I]/I; CSS2[I]=CSS2[I]-TX2[I]*TX2[I]/(SAMPLENO-I)
                R2[I]=TSS-(CSS1[I]+CSS2[I]); R2[I]=R2[I]/TSS*100
        ENDFOR
CUT=1
        FOR I = 2 TO SAMPLENO-1
                IF R2[I+1] > R2[I]
                        CUT=CUT+1
                ELSE
                        EXIT
                ENDIF
        ENDFOR
```

**GETTING PARAMETERS IN ORDER**

```
DECLARE HIGH1[CUT, PARANO+1], LOW1[SAMPLENO-CUT, PARANO+1]
LOWPOP=SAMPLENO-CUT; HIGHPOP=CUT
        FOR KKK= 1 TO PARANO+1
                FOR I = 1 TO HIGHPOP
                        HIGH1[I,KKK]=DAT1[I, KKK+1]
                ENDFOR
        ENDFOR
KKK=0; X=1
        FOR I= 1 TO PARANO+1
                FOR J= HIGHPOP + 1 TO SAMPLENO
                        LOW1[X,I]=DAT1[J,I+1]; IF X=SAMPLENO-(HIGHPOP)
                                X=1
                        ELSE
                                X=X+1
                        ENDIF
                ENDFOR
        ENDFOR
```

**INDICES CALCULATION - * CALCULATION OF INDICES A/B FOR HIGH POPULATION**

```
INDNO=INT((PARANO*(PARANO-1))/2)
DECLARE IND1[HIGHPOP, INDNO], IND2[HIGHPOP, INDNO]
I=0; XXX=1; K1=0
        FOR J= 1 TO PARANO
                FOR K1=(J+1) TO PARANO
                        FOR I= 1 TO HIGHPOP
                                IND1[I,XXX]=HIGH1[I,J]/HIGH1[I,INT(K1)]; IND2[I,XXX]=HIGH1[I,INT(K1)]/HIGH1[I,J]
                        ENDFOR
                        XXX=XXX+1
                ENDFOR
        ENDFOR
```

**CALCULATION OF MEAN, SD, CV, VARIANCE,**

```
DECLARE SUMM1[INDNO], SSQ1[INDNO], VAR1[INDNO], CV1[INDNO], SSD1[INDNO], MEAN1[INDNO]
X=0; Y=0
        FOR I= 1 TO INDNO
                FOR J= 1 TO HIGHPOP
                        SUM1=SUM1+IND1[J,I]; SSS1=SSS1+(IND1[J,I]*IND1[J,I])
```

```
                ENDFOR
                        SUMM1[I]=SUM1; MEAN1[I]=SUMM1[I]/HIGHPOP; SSQ1[I]=SSS1
                        VAR1[I]=((HIGHPOP*SSQ1[I])-SUMM1[I])/(HIGHPOP*(HIGHPOP-1))
                        VAR1[I]=ABS(VAR1[I]); SSD1[I]=SQRT(VAR1[I]); CV1[I]=(SQRT(VAR1[I]))/MEAN1[I]; SUM1=0; SSS1=0
        ENDFOR
SSS1=0; SUM1=0
DECLARE SUMM2[INDNO], SSQ2[INDNO], VAR2[INDNO], CV2[INDNO], SSD2[INDNO], MEAN2[INDNO]
        FOR I= 1 TO INDNO
                FOR J= 1 TO HIGHPOP
                        SUM1=SUM1+IND2[J,I]; SSS1=SSS1+(IND2[J,I]*IND2[J,I])
                ENDFOR
                        SUMM2[I]=SUM1; MEAN2[I]=SUMM2[I]/HIGHPOP; SSQ2[I]=SSS1
                        VAR2[I]=((HIGHPOP*SSQ2[I])-SUMM2[I])/(HIGHPOP*(HIGHPOP-1))
                        VAR2[I]=ABS(VAR2[I]); SSD2[I]=SQRT(VAR2[I]); CV2[I]=(SQRT(VAR2[I]))/MEAN2[I]
                        SUM1=0; SSS1=0
        ENDFOR
*****************************************************************************************************************
                        CALCULATION FOR LOW POPULATION SAMPLES
*****************************************************************************************************************
DECLARE IND3[LOWPOP, INDNO], IND4[LOWPOP, INDNO]
K1=0; X=1
        FOR J= 1 TO PARANO
                FOR K1=(J+1) TO PARANO
                        FOR I= 1 TO LOWPOP
                                IND3[I,X]=LOW1[I,J]/LOW1[I,K1]; IND4[I,X]=LOW1[I,K1]/LOW1[I,J]
                        ENDFOR
                        X=X+1
                ENDFOR
        ENDFOR
SUM1=0; SSS1=0
DECLARE SUMM3[INDNO], SSQ3[INDNO], VAR3[INDNO], CV3[INDNO], SSD3[INDNO], MEAN3[INDNO], VARRATIO1[INDNO]
        FOR I= 1 TO INDNO
                FOR J= 1 TO LOWPOP
                        SUM1=SUM1+IND3[J,I]; SSS1=SSS1+(IND3[J,I]*IND3[J,I])
                ENDFOR
                        SUMM3[I]=SUM1; MEAN3[I]=SUMM3[I]/LOWPOP; SSQ3[I]=SSS1
                        VAR3[I]=(( LOWPOP *SSQ3[I])-SUMM3[I])/( LOWPOP *( LOWPOP -1)); VAR3[I]=ABS(VAR3[I])
                        SSD3[I]=SQRT(VAR3[I]); CV3[I]=(SQRT(VAR3[I]))/MEAN3[I]; VARRATIO1[I]=VAR3[I]/VAR1[I]; SUM1=0; SSS1=0
        ENDFOR
SSS1=0; SUM1=0
DECLARE SUMM4[INDNO], SSQ4[INDNO], VAR4[INDNO], CV4[INDNO], SSD4[INDNO], MEAN4[INDNO], VARRATIO2[INDNO]
        FOR I= 1 TO INDNO
                FOR J= 1 TO LOWPOP
                        SUM1=SUM1+IND4[J,I]; SSS1=SSS1+(IND4[J,I]*IND4[J,I])
                ENDFOR
                        SUMM4[I]=SUM1; MEAN4[I]=SUMM4[I]/ LOWPOP; SSQ4[I]=SSS1
                        VAR4[I]=((HIGHPOP*SSQ4[I])-SUMM4[I])/(LOWPOP*( LOWPOP -1))
                        VAR4[I]=ABS(VAR4[I]); SSD4[I]=SQRT(VAR4[I]); CV4[I]=(SQRT(VAR4[I]))/MEAN4[I]
                        VARRATIO2[I]=VAR4[I]/VAR2[I]; SUM1=0; SSS1=0
        ENDFOR
*****************************************************************************************************************
                CALCULATION OF DEFICIENCY, OPTIMUM AND SUFFICIENCY CUTOFF LEVELS
*****************************************************************************************************************
```

```
DECLARE INDF1[HIGHPOP, INDNO], INDF2[LOWPOP, INDNO], INDIC1[INDNO], PARAM1[INDNO, 16], LOW[INDNO], DEF[INDNO],
        OPT[INDNO], HIG[INDNO]
        FOR I= 1 TO INDNO
                PARAM1[I,15]=VARRATIO1[I]; PARAM1[I,16]=VARRATIO2[I]
                        IF VARRATIO1[I] > VARRATIO2[I]

                                FOR J= 1 TO HIGHPOP
                                        INDIC1[I]=1; INDF1[J, I]= IND1[J, I]
                                ENDFOR
                                FOR J= 1 TO LOWPOP
                                        INDF2[J,I]=IND3[J,I]
                                ENDFOR
                                DEF[I]=MEAN1[I]-(SSD1[I]*(2.67)); LOW[I]=MEAN1[I]-(SSD1[I]*(1.33));
OPT[I]=MEAN1[I]+(SSD1[I]*(1.33))
                                HIG[I]=MEAN1[I]+(SSD1[I]*(2.67)); PARAM1[I,1]=MEAN1[I]; PARAM1[I,2]=CV1[I]
                                PARAM1[I,3]=SSD1[I]; PARAM1[I,4]=VAR1[I]; PARAM1[I,5]=MEAN3[I]
                                PARAM1[I,6]=CV3[I]; PARAM1[I,7]=SSD3[I]; PARAM1[I,8]=VAR3[I]
                                PARAM1[I,9]=VARRATIO1[I]; PARAM1[I,10]=INDIC1[I]; PARAM1[I,11]=DEF[I]
                                PARAM1[I,12]=LOW[I]; PARAM1[I,13]=OPT[I]; PARAM1[I,14]=HIG[I]
                        ELSE
                                FOR J= 1 TO HIGHPOP
                                        INDIC1[I]=2; INDF1[J,I]=IND2[J,I]
                                ENDFOR
                                FOR J= 1 TO LOWPOP
                                        INDF2[J,I]=IND4[J,I]
                                ENDFOR
                                DEF[I]=MEAN2[I]-(SSD2[I]*(2.67)); LOW[I]=MEAN2[I]-(SSD2[I]*(1.33));
OPT[I]=MEAN2[I]+(SSD2[I]*(1.33))
                                HIG[I]=MEAN2[I]+(SSD2[I]*(2.67)); PARAM1[I,1]=MEAN2[I];PARAM1[I,2]=CV2[I]
                                PARAM1[I,3]=SSD2[I]; PARAM1[I,4]=VAR2[I]; PARAM1[I,5]=MEAN4[I]
                                PARAM1[I,6]=CV4[I]; PARAM1[I,7]=SSD4[I]; PARAM1[I,8]=VAR4[I]
                                PARAM1[I,9]=VARRATIO2[I]; PARAM1[I,10]=INDIC1[I]; PARAM1[I,11]=DEF[I]
                                PARAM1[I,12]=LOW[I]; PARAM1[I,13]=OPT[I]; PARAM1[I,14]=HIG[I]
                        ENDIF
        ENDFOR
```

**********

**CALCULATION OF NUTRIENT INDEX FUNCTIONS**

**********

```
DECLARE INDICESF1[HIGHPOP, INDNO], INDICESF2[LOWPOP, INDNO]

        FOR J= 1 TO INDNO
                FOR I= 1 TO HIGHPOP
                        IF INDF1[I,J] >= PARAM1[J, 1]
                                INDICESF1[I,J]=(((INDF1[I,J]/ABS(PARAM1[J,1]))-1)*(1000/PARAM1[J,2]))
                        ELSE
                                INDICESF1[I,J]=((1-(PARAM1[J,1]/INDF1[I,J]))*(1000/PARAM1[J,2]))
                        ENDIF
                ENDFOR
        ENDFOR

        FOR J= 1 TO INDNO
                FOR I= 1 TO LOWPOP
                        IF INDF2[I,J] >= PARAM1[J, 1]
                                INDICESF2[I,J]=(((INDF2[I,J]/PARAM1[J,1])-1)*(1000/PARAM1[J,2]))
```

```
                    ELSE
                            INDICESF2[I,J]=((1-(PARAM1[J,1]/INDF2[I,J]))*(1000/PARAM1[J,2]))
                    ENDIF
            ENDFOR
        ENDFOR
DECLARE CLUE[PARANO, PARANO-1]
X=1
Z1=1
        FOR I = 1 TO PARANO
            FOR J= Z1 TO PARANO-1
                            IF INDIC1[X]#1
                                    CLUE[I,J]=-1
                            ELSE
                                    CLUE[I,J]=1
                            ENDIF
                    X=X+1
            ENDFOR
                    Z1=Z1+1
        ENDFOR
X=1
        FOR I = 1 TO PARANO-1
            FOR J= I+1 TO PARANO
                    IF INDIC1[X]#1
                            CLUE[J,I]=1
                    ELSE
                            CLUE[J,I]=-1
                    ENDIF
                    X=X+1
            ENDFOR
        ENDFOR
```

**********

**CALCULATION OF DRIS INDEX AND NII VALUES**

**********

```
DECLARE RAW1[PARANO, PARANO], TOTDRIS1[HIGHPOP], DRIS1[HIGHPOP, PARANO+2], TOTDRIS2[HIGHPOP]
A1=0; Y=1; Z=1
FOR X1= 1 TO HIGHPOP
X=1; Z1=1
        FOR I = 1 TO PARANO
            FOR J= Z1 TO PARANO-1
                            RAW1[I,J]=INDICESF1[X1 ,X]*CLUE[I,J]
                    X=X+1
            ENDFOR
                    Z1=Z1+1
        ENDFOR
X=1
        FOR I = 1 TO PARANO-1
            FOR J= I+1 TO PARANO
                    RAW1[J,I]=INDICESF1[X1 ,X]*CLUE[J,I]
            X=X+1
            ENDFOR
        ENDFOR
TOTDRIS1[X1]=0; TOTDRIS2[X1]=0
TOTD=0
        FOR G= 1 TO PARANO
```

```
                DRIS1[X1,Y]=0
                        FOR H= 1 TO PARANO-1
                                DRIS1[X1,Y]=DRIS1[X1,Y]+RAW1[G, H]
                        ENDFOR
                DRIS1[X1,Y]=DRIS1[X1,Y]/PARANO; TOTD1=ABS(DRIS1[X1,Y]); TOTD=TOTD+TOTD1; Y=Y+1
                        IF Y>PARANO; Y=1; ENDIF
        ENDFOR
TOTDRIS1[X1]=TOTD
ENDFOR
        FOR I= 1 TO HIGHPOP
                DRIS1[I, PARANO+1]=TOTDRIS1[I]; DRIS1[I, PARANO+2]=HIGH1[I,PARANO+1]
        ENDFOR

DECLARE RAW2[PARANO, PARANO], TOTDRIS3[LOWPOP], DRIS2[LOWPOP, PARANO+2], TOTDRIS4[LOWPOP]
A1=0; Y=1; Z=1
FOR X1 = 1 TO LOWPOP
        X=1; Z1=1
        FOR I = 1 TO PARANO
                FOR J= Z1 TO PARANO-1
                                RAW2[I,J]=INDICESF2[X1 ,X]*CLUE[I,J]
                        X=X+1
                ENDFOR
                        Z1=Z1+1
        ENDFOR
        X=1
        FOR I = 1 TO PARANO-1
                FOR J= I+1 TO PARANO
                        RAW2[J,I]=INDICESF2[X1 ,X]*CLUE[J,I]
                X=X+1
                ENDFOR
        ENDFOR
        TOTDRIS3[X1]=0; TOTDRIS4[X1]=0; TOTD=0
        FOR G= 1 TO PARANO
        DRIS2[X1,Y]=0
                FOR H= 1 TO PARANO-1
                        DRIS2[X1,Y]=DRIS2[X1,Y]+RAW2[G, H]
                ENDFOR
                DRIS2[X1,Y]=DRIS2[X1,Y]/PARANO; TOTD1=ABS(DRIS2[X1,Y]); TOTD=TOTD+TOTD1; Y=Y+1
                IF Y>PARANO
                        Y=1
                ENDIF
        ENDFOR
        TOTDRIS3[X1]=TOTD
ENDFOR

        FOR I= 1 TO LOWPOP
                DRIS2[I, PARANO+1]=TOTDRIS3[I]; DRIS2[I, PARANO+2]=LOW1[I,PARANO+1]
        ENDFOR
*****************************************************************************************************************
                TRANSFER OF DATA OUTPUT TO EXCEL & CLOSING ALL DATABASES AND MEMORY VARIABLES
*****************************************************************************************************************
SELE 2
        APPEND FROM ARRAY DRIS2
```

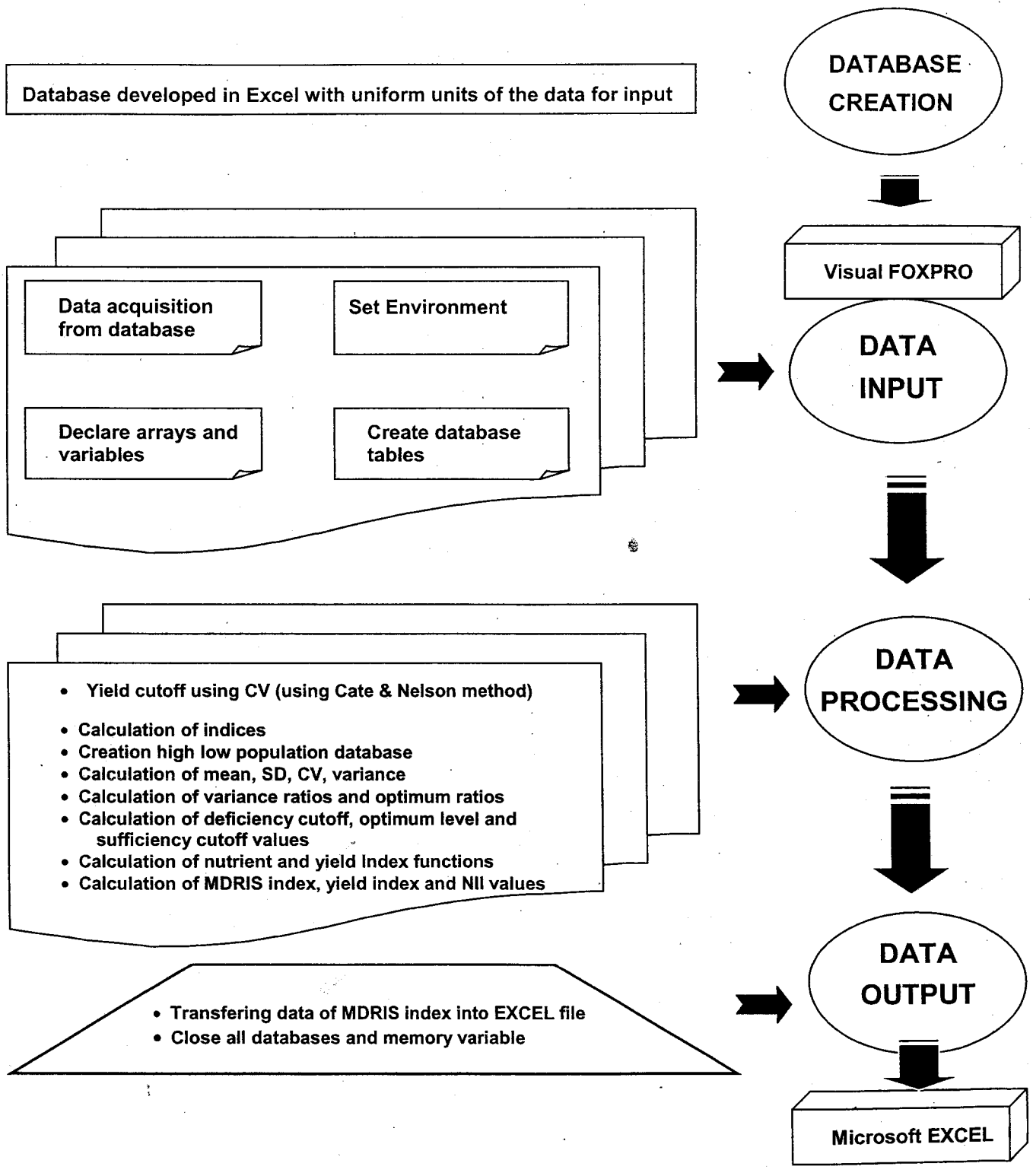
```
        COPY TO &T2 TYPE XLS
CLEAR ALL; CLEAR; CLOSE ALL
RETURN
```

CHAPTER - III

# MDRIS Model

## (MODIFIED DIAGNOSIS AND RECOMMENDATION INTEGRATED SYSTEM)

i. Flow chart of the computer program - MDRIS



## ii. Computer program - MDRIS model

```
*****************************************************************************************
                                SET ENVIRONMENT
*****************************************************************************************
CLEAR; CLEAR ALL; CLOSE ALL
SET SCOREBOARD OFF; SET STATUS OFF; SET TALK OFF; SET CONFIRM OFF; SET SAFETY OFF
CLOSE ALL
CLEAR

F1 = SPACE(20) ; T1=SPACE(20) ; T16=SPACE(20) ; XX1=SPACE(40)
STORE 0 TO SAMPLENO, PARANO, INDNO

        @ 2, 10 SAY " FILE NAME"
        @ 2, 30 GET F1
        @4,10 SAY " ENTER TOTAL NUMBER OF SAMPLES"
        @4, 70 GET SAMPLENO
        @ 6, 10 SAY " ENTER NO OF PARAMETERS"
        @ 6, 60 GET PARANO
        @ 8, 10 SAY "ENTER OUTPUT FILE NAME"
        @ 8, 60 GET XX1
        READ

T1= ALLTRIM(XX1)+"MMAINSORT"
T16=ALLTRIM(XX1)+"MMDRISLOW"

CREATE TABLE &T1 (Y N(6,2), N N(9,4), P N(9,4), K N(9,5), CA N(9,5), MG N(9,5), NA N(9,5), S N(9,5), B N(9,5), ZN N(9,5), CU N(9,5), FE N(9,5), MN N(9,5), FNO N(4))
CREATE TABLE &T16 (N N(9,4), P N(9,4), K N(9,4), CA N(9,4), MG N(9,4), NA N(9,4), S N(9,4), B N(9,4), ZN N(9,4), CU N(9,4), FE N(9,4), MN N(9,4), A13 N(9,4), TOT
                N(9,4), FNO N(4))
        DECLARE DAT1 [SAMPLENO, PARANO+2], CSS1[SAMPLENO], CSS2[SAMPLENO], CSS[SAMPLENO], CSS3[SAMPLENO], CSS4[SAMPLENO],
                R2[SAMPLENO]
        DECLARE TX1[SAMPLENO], TX2[SAMPLENO]
        STORE 0 TO SS1, SS2, SS3, SS4, SS9, SS10, CAL1, I, LL1, XX2, AA3, AA4, XX5, AA5, AA6, AA7, AA9, ZZZ, MAX1
        STORE 0 TO MAX2, MAX3, MAX4, MAX5, LOWPOP, HIGHPOP, D, XXX, YYY, I, J, KKK, AAA, BBB, Z, FFF, GGG
        STORE 0 TO SSS1, SUM1, MEAN1, VARR1, AAAA, BBBB, DRISTEMP1, DRISTEMP2, AA, BB
        STORE 0 TO TT, CUT, SS, K1, X, Z1, G, H, X1,L1, TOTD1, TOTD, MMEAN1, MMEAN2, TMDRIS, TMDRIS1, TMDRIS2, YYDRIS1,
YYDRIS2
SELECT 1
        USE &T1 ALIAS X1; ZAP
        APPEND FROM &F1 DELIMITED WITH TAB
        COPY TO ARRAY DAT1; ASORT (DAT1, 1, -1,1)
        CLEAR; ZAP
CLOSE ALL
SELE 16
USE &T16 ALIAS X16; ZAP

*****************************************************************************************
                        CV CALCULATION AND SELECTION OF CUT OFF FOR POPULATION
*****************************************************************************************
TT=0; SS=0
        FOR I=1 TO SAMPLENO-1
                TT=TT+DAT1[I,1] ; SS=SS+(DAT1[I,1]*DAT1[I,1]); TX1[I]=TT; CSS1[I]=SS
        ENDFOR
TT=TT+DAT1[SAMPLENO,1]; SS=SS+DAT1[SAMPLENO,1]*DAT1[SAMPLENO,1]; TX1[SAMPLENO]=TT
CSS1[SAMPLENO]=SS; TSS=CSS1[SAMPLENO]-TX1[SAMPLENO]*TX1[SAMPLENO]/SAMPLENO
```

```
        FOR I=2 TO SAMPLENO-1
                TX2[I]=TX1[SAMPLENO]-TX1[I]; CSS2[I]=CSS1[SAMPLENO]-CSS1[I]; CSS1[I]=CSS1[I]-TX1[I]*TX1[I]/I
                CSS2[I]=CSS2[I]-TX2[I]*TX2[I]/(SAMPLENO-I); R2[I]=TSS-(CSS1[I]+CSS2[I]); R2[I]=R2[I]/TSS*100
        ENDFOR
CUT=1
        FOR I = 2 TO SAMPLENO-1
                IF R2[I+1] > R2[I]
                        CUT=CUT+1
                ELSE
                        EXIT
                ENDIF
        ENDFOR
```

**********

GETTING PARAMETERS IN ORDER

**********

```
DECLARE HIGH1[CUT, PARANO+1], LOW1[SAMPLENO-CUT, PARANO+1]
LOWPOP=SAMPLENO-CUT; HIGHPOP=CUT
        FOR KKK= 1 TO PARANO+1
                FOR I = 1 TO HIGHPOP
                        HIGH1[I,KKK]=DAT1[I, KKK+1]
                ENDFOR
        ENDFOR
KKK=0; X=1
        FOR I= 1 TO PARANO+1
                FOR J= HIGHPOP + 1 TO SAMPLENO
                        LOW1[X,I]=DAT1[J,I+1]
                        IF X=SAMPLENO-(HIGHPOP)
                                X=1
                        ELSE
                                X=X+1
                        ENDIF
                ENDFOR
        ENDFOR
```

**********

**CALCULATION OF INDICES A/B FOR HIGH POPULATION**

**********

```
INDNO=INT((PARANO*(PARANO-1))/2)
DECLARE IND1[HIGHPOP, INDNO], IND2[HIGHPOP, INDNO]
I=0; XXX=1; K1=0
        FOR J= 1 TO PARANO
                FOR K1=(J+1) TO PARANO
                        FOR I= 1 TO HIGHPOP
                                IND1[I,XXX]=HIGH1[I,J]/HIGH1[I,INT(K1)]; IND2[I,XXX]=HIGH1[I,INT(K1)]/HIGH1[I,J]
                        ENDFOR
                        XXX=XXX+1
                ENDFOR
        ENDFOR
```

**********

**CALCULATION OF MEAN, SD, CV, VARIANCE,**

**********

```
DECLARE SUMM1[INDNO], SSQ1[INDNO], VAR1[INDNO], CV1[INDNO], SSD1[INDNO], MEAN1[INDNO]
X=0; Y=0
        FOR I= 1 TO INDNO
                FOR J= 1 TO HIGHPOP
```

```
                SUM1=SUM1+IND1[J,I]; SSS1=SSS1+(IND1[J,I]*IND1[J,I])
            ENDFOR
            SUMM1[I]=SUM1; MEAN1[I]=SUMM1[I]/HIGHPOP; SSQ1[I]=SSS1
            VAR1[I]=((HIGHPOP*SSQ1[I])-(SUMM1[I]*SUMM1[I]))/(HIGHPOP*(HIGHPOP-1))
            VAR1[I]=ABS(VAR1[I]); SSD1[I]=SQRT(VAR1[I]); CV1[I]=(SQRT(VAR1[I]))/MEAN1[I]; SUM1=0; SSS1=0
        ENDFOR
SSS1=0; SUM1=0
DECLARE SUMM2[INDNO], SSQ2[INDNO], VAR2[INDNO], CV2[INDNO], SSD2[INDNO], MEAN2[INDNO]
        FOR I= 1 TO INDNO
            FOR J= 1 TO HIGHPOP
                SUM1=SUM1+IND2[J,I]; SSS1=SSS1+(IND2[J,I]*IND2[J,I])
            ENDFOR
                SUMM2[I]=SUM1; MEAN2[I]=SUMM2[I]/HIGHPOP; SSQ2[I]=SSS1
                VAR2[I]=((HIGHPOP*SSQ2[I])-(SUMM2[I]*SUMM2[I]))/(HIGHPOP*(HIGHPOP-1))
                VAR2[I]=ABS(VAR2[I]); SSD2[I]=SQRT(VAR2[I]); CV2[I]=(SQRT(VAR2[I]))/MEAN2[I]; SUM1=0; SSS1=0
        ENDFOR
*****************************************************************************************************************
                        CALCULATION FOR LOW POPULATION SAMPLES
*****************************************************************************************************************
DECLARE IND3[LOWPOP, INDNO], IND4[LOWPOP, INDNO]
K1=0; X=1
        FOR J= 1 TO PARANO
            FOR K1=(J+1) TO PARANO
                FOR I= 1 TO LOWPOP
                    IND3[I,X]=LOW1[I,J]/LOW1[I,K1]; IND4[I,X]=LOW1[I,K1]/LOW1[I,J]
                ENDFOR
                X=X+1
            ENDFOR
        ENDFOR

SUM1=0; SSS1=0
DECLARE SUMM3[INDNO], SSQ3[INDNO], VAR3[INDNO], CV3[INDNO], SSD3[INDNO], MEAN3[INDNO], VARRATIO1[INDNO]
        FOR I= 1 TO INDNO
            FOR J= 1 TO LOWPOP
                SUM1=SUM1+IND3[J,I]; SSS1=SSS1+(IND3[J,I]*IND3[J,I])
            ENDFOR
                SUMM3[I]=SUM1; MEAN3[I]=SUMM3[I]/LOWPOP; SSQ3[I]=SSS1
                VAR3[I]=(( LOWPOP *SSQ3[I])-(SUMM3[I]*SUMM3[I]))/( LOWPOP *( LOWPOP -1))
                VAR3[I]=ABS(VAR3[I]); SSD3[I]=SQRT(VAR3[I]); CV3[I]=(SQRT(VAR3[I]))/MEAN3[I]
                VARRATIO1[I]=VAR3[I]/VAR1[I]; SUM1=0; SSS1=0
        ENDFOR
SSS1=0; SUM1=0
DECLARE SUMM4[INDNO], SSQ4[INDNO], VAR4[INDNO], CV4[INDNO], SSD4[INDNO], MEAN4[INDNO], VARRATIO2[INDNO]
        FOR I= 1 TO INDNO
            FOR J= 1 TO LOWPOP
                SUM1=SUM1+IND4[J,I]; SSS1=SSS1+(IND4[J,I]*IND4[J,I])
            ENDFOR
                SUMM4[I]=SUM1; MEAN4[I]=SUMM4[I]/ LOWPOP; SSQ4[I]=SSS1
                VAR4[I]=((HIGHPOP*SSQ4[I])-(SUMM4[I]*SUMM4[I]))/(LOWPOP*( LOWPOP -1))
                VAR4[I]=ABS(VAR4[I]); SSD4[I]=SQRT(VAR4[I]); CV4[I]=(SQRT(VAR4[I]))/MEAN4[I]
                VARRATIO2[I]=VAR4[I]/VAR2[I]; SUM1=0; SSS1=0
        ENDFOR
DECLARE INDF1[HIGHPOP, INDNO], INDF2[LOWPOP, INDNO], INDIC1[INDNO], PARAM1[INDNO, 16], LOW[INDNO], DEF[INDNO],
OPT[INDNO], HIG[INDNO]
```

```
FOR I= 1 TO INDNO
        PARAM1[I,15]=VARRATIO1[I]; PARAM1[I,16]=VARRATIO2[I]
                IF VARRATIO1[I] > VARRATIO2[I]
                        FOR J= 1 TO HIGHPOP
                                INDIC1[I]=1; INDF1[J, I]= IND1[J, I]
                        ENDFOR
                        FOR J= 1 TO LOWPOP
                                INDF2[J,I]=IND3[J,I]
                        ENDFOR
                DEF[I]=MEAN1[I]-(SSD1[I]*(2.67)); LOW[I]=MEAN1[I]-(SSD1[I]*(1.33)); OPT[I]=MEAN1[I]+(SSD1[I]*(1.33))
                        HIG[I]=MEAN1[I]+(SSD1[I]*(2.67)); PARAM1[I,1]=MEAN1[I]; PARAM1[I,2]=CV1[I]
                        PARAM1[I,3]=SSD1[I]; PARAM1[I,4]=VAR1[I]; PARAM1[I,5]=MEAN3[I]; PARAM1[I,6]=CV3[I]
                        PARAM1[I,7]=SSD3[I]; PARAM1[I,8]=VAR3[I]; PARAM1[I,9]=VARRATIO1[I]; PARAM1[I,10]=INDIC1[I]
                        PARAM1[I,11]=DEF[I]; PARAM1[I,12]=LOW[I]; PARAM1[I,13]=OPT[I]; PARAM1[I,14]=HIG[I]
                ELSE
                        FOR J= 1 TO HIGHPOP
                                INDIC1[I]=2; INDF1[J,I]=IND2[J,I]
                        ENDFOR
                        FOR J= 1 TO LOWPOP
                                INDF2[J,I]=IND4[J,I]
                        ENDFOR
                DEF[I]=MEAN2[I]-(SSD2[I]*(2.67)); LOW[I]=MEAN2[I]-(SSD2[I]*(1.33)); OPT[I]=MEAN2[I]+(SSD2[I]*(1.33))
                HIG[I]=MEAN2[I]+(SSD2[I]*(2.67)); PARAM1[I,1]=MEAN2[I]; PARAM1[I,2]=CV2[I]; PARAM1[I,3]=SSD2[I]
                        PARAM1[I,4]=VAR2[I]; PARAM1[I,5]=MEAN4[I]; PARAM1[I,6]=CV4[I]; PARAM1[I,7]=SSD4[I]
                        PARAM1[I,8]=VAR4[I]; PARAM1[I,9]=VARRATIO2[I]; PARAM1[I,10]=INDIC1[I]; PARAM1[I,11]=DEF[I]
                        PARAM1[I,12]=LOW[I]; PARAM1[I,13]=OPT[I]; PARAM1[I,14]=HIG[I]
                ENDIF
ENDFOR
DECLARE INDICESF1[HIGHPOP, INDNO], INDICESF2[LOWPOP, INDNO]
FOR J= 1 TO INDNO
        FOR I= 1 TO HIGHPOP
                IF INDF1[I,J] >= PARAM1[J, 1]
                        INDICESF1[I,J]=(((INDF1[I,J]/ABS(PARAM1[J,1]))-1)*(1000/PARAM1[J,2]))
                ELSE
                        INDICESF1[I,J]=((1-(PARAM1[J,1]/INDF1[I,J]))*(1000/PARAM1[J,2]))
                ENDIF
        ENDFOR
ENDFOR
FOR J= 1 TO INDNO
        FOR I= 1 TO LOWPOP
                IF INDF2[I,J] >= PARAM1[J, 1]
                        INDICESF2[I,J]=(((INDF2[I,J]/PARAM1[J,1])-1)*(1000/PARAM1[J,2]))
                ELSE
                        INDICESF2[I,J]=((1-(PARAM1[J,1]/INDF2[I,J]))*(1000/PARAM1[J,2]))
                ENDIF
        ENDFOR
ENDFOR
DECLARE CLUE[PARANO, PARANO-1]
X=1; Z1=1
FOR I = 1 TO PARANO
        FOR J= Z1 TO PARANO-1
                IF INDIC1[X]#1
                        CLUE[I,J]=-1
```

```
                                ELSE
                                        CLUE[I,J]=1
                                ENDIF
                        X=X+1
                ENDFOR
                        Z1=Z1+1
        ENDFOR
X=1
        FOR I = 1 TO PARANO-1
                FOR J= I+1 TO PARANO
                        IF INDIC1[X]#1
                                CLUE[J,I]=1
                        ELSE
                                CLUE[J,I]=-1
                        ENDIF
                        X=X+1
                ENDFOR
        ENDFOR
DECLARE RAW1[PARANO, PARANO], TOTDRIS1[HIGHPOP], DRIS1[HIGHPOP, PARANO+1], TOTDRIS2[HIGHPOP]
A1=0; Y=1; Z=1
FOR X1= 1 TO HIGHPOP
X=1; Z1=1
        FOR I = 1 TO PARANO
                FOR J= Z1 TO PARANO-1
                        RAW1[I,J]=INDICESF1[X1 ,X]*CLUE[I,J]
                        X=X+1
                ENDFOR
                        Z1=Z1+1
        ENDFOR
X=1
        FOR I = 1 TO PARANO-1
                FOR J= I+1 TO PARANO
                        RAW1[J,I]=INDICESF1[X1 ,X]*CLUE[J,I]
                X=X+1
                ENDFOR
        ENDFOR

TOTDRIS1[X1]=0; TOTDRIS2[X1]=0; TOTD=0
        FOR G= 1 TO PARANO
                DRIS1[X1,Y]=0
                        FOR H= 1 TO PARANO-1
                                DRIS1[X1,Y]=DRIS1[X1,Y]+RAW1[G, H]
                        ENDFOR
                        TOTD1=ABS(DRIS1[X1,Y]); TOTD=TOTD+TOTD1; Y=Y+1
                IF Y> PARANO
                        Y=1
                ENDIF
        ENDFOR
TOTDRIS1[X1]=TOTD
ENDFOR

DECLARE RAW2[PARANO, PARANO], TOTDRIS3[LOWPOP], DRIS2[LOWPOP, PARANO+1], TOTDRIS4[LOWPOP]
A1=0; Y=1; Z=1
```

```
FOR X1 = 1 TO LOWPOP
X=1; Z1=1
        FOR I = 1 TO PARANO
                FOR J= Z1 TO PARANO-1
                        RAW2[I,J]=INDICESF2[X1 ,X]*CLUE[I,J]
                X=X+1
                ENDFOR
        Z1=Z1+1
        ENDFOR
X=1
        FOR I = 1 TO PARANO-1
                FOR J= I+1 TO PARANO
                        RAW2[J,I]=INDICESF2[X1 ,X]*CLUE[J,I]
                X=X+1
                ENDFOR
        ENDFOR

TOTDRIS3[X1]=0; TOTDRIS4[X1]=0; TOTD=0
        FOR G= 1 TO PARANO
                DRIS2[X1,Y]=0
                        FOR H= 1 TO PARANO-1
                        DRIS2[X1,Y]=DRIS2[X1,Y]+RAW2[G, H]
                        ENDFOR
                DRIS2[X1,Y]=DRIS2[X1,Y]/PARANO; Y=Y+1
                IF Y>PARANO
                        Y=1
                ENDIF
        ENDFOR
ENDFOR
```

**MDRIS INDICES CALCULATION**

```
DECLARE MIND1[HIGHPOP, PARANO], MMEANN1[PARANO], MIND2[LOWPOP, PARANO], MMEANN2[PARANO], MIND3[HIGHPOP,
PARANO], MIND4[LOWPOP, PARANO], MDRIS1[HIGHPOP, PARANO+3], MDRIS2[LOWPOP, PARANO+3], TOTMDRIS1[HIGHPOP],
TOTMDRIS2[LOWPOP], YDRIS1[HIGHPOP], YDRIS2[LOWPOP], SUMMM2[PARANO], SSSQ2[PARANO], VARR1[PARANO],
CVV2[PARANO], SSSD2[PARANO], MMEAN1[PARANO]
K=1; MMEAN1=0; SSSS1=0; SUMM1=0
        FOR I=1 TO PARANO
                FOR J= 1 TO HIGHPOP
                        MIND1[J,I]=HIGH1[J,I]/DAT1[K,1]; MMEAN1=MMEAN1+MIND1[J,I]; SUMM1=SUMM1+MIND1[J,I]
                        SSSS1=SSSS1+(MIND1[J,I]*MIND1[J,I]); K=K+1
                ENDFOR
                        SUMMM2[I]=SUMM1; MMEANN1[I]=MMEAN1/HIGHPOP; SSSQ2[I]=SSSS1
                        VARR1[I]=((HIGHPOP*SSSQ2[I])-SUMMM2[I])/(HIGHPOP*(HIGHPOP-1))
                        VARR1[I]=ABS(VARR1[I]); SSSD2[I]=SQRT(VARR1[I]); CVV2[I]=(SQRT(VARR1[I]))/MMEANN1[I]
                MMEAN1=0; SUMM1=0; SSSS1=0; K=1
        ENDFOR
K=HIGHPOP+1; MMEAN2=0
        FOR I=1 TO PARANO
                FOR J= 1 TO LOWPOP
                        MIND2[J,I]=LOW1[J,I]/DAT1[K,1]; K=K+1; MMEAN2=MMEAN2+MIND2[J,I]
                ENDFOR
                K=HIGHPOP+1; MMEANN2[I]=MMEAN2/LOWPOP; MMEAN2=0
        ENDFOR
K=1; TMDRIS=0; TMDRIS1=0; YYDRIS1=0
```

```
        FOR J= 1 TO HIGHPOP
                FOR I= 1 TO PARANO
                        IF MIND1[J,I] >= MMEANN1[K]
                                MIND3[J,I]=(((MIND1[J,I]/ABS(MMEANN1[K]))-1)*(1000/CVV2[I]))
                                MDRIS1[J,I]=(DRIS1[J,I]+MIND3[J,I])/(PARANO+1)
                        ELSE
                                MIND3[J,I]=((1-(ABS(MMEANN1[K])/MIND1[J,I]))*(1000/CVV2[I]))
                                MDRIS1[J,I]=(DRIS1[J,I]+MIND3[J,I])/(PARANO+1)
                        ENDIF
                        K=K+1
                        TMDRIS1=ABS(MDRIS1[J,I]); TMDRIS=TMDRIS+TMDRIS1; YYDRIS1=YYDRIS1-MIND3[J,I]
                ENDFOR
                YDRIS1[J]=YYDRIS1; K=1; TOTMDRIS1[J]=TMDRIS; TMDRIS=0; YYDRIS1=0
        ENDFOR
                FOR I= 1 TO HIGHPOP
                        MDRIS1[I, PARANO+1]=(YDRIS1[I]/(PARANO+1))
                        MDRIS1[I, PARANO+2]=TOTMDRIS1[I]+(ABS(YDRIS1[I])/(PARANO+1))
                        MDRIS1[I, PARANO+3]=HIGH1[I, PARANO+1]
                ENDFOR

TMDRIS2=0; TMDRIS=0; YYDRIS2=0; K=1
        FOR J= 1 TO LOWPOP
                FOR I= 1 TO PARANO
                        IF MIND2[J,I] >= MMEANN2[K]
                                MIND4[J,I]=(((MIND2[J,I]/MMEANN1[K])-1)*(1000/CVV2[I]))
                                MDRIS2[J,I]=(DRIS2[J,I]+MIND4[J,I])/(PARANO+1)
                        ELSE
                                MIND4[J,I]=((1-(MMEANN1[K]/MIND2[J,I]))*(1000/CVV2[I]))
                                MDRIS2[J,I]=(DRIS2[J,I]+MIND4[J,I])/(PARANO+1)
                        ENDIF
                        K=K+1; YYDRIS2=YYDRIS2-MIND4[J,I]; TMDRIS2=ABS(MDRIS2[J,I]); TMDRIS=TMDRIS+TMDRIS2
                ENDFOR
                YDRIS2[J]=YYDRIS2;  K=1; TOTMDRIS2[J]=TMDRIS; TMDRIS=0; YYDRIS2=0
        ENDFOR
FOR I= 1 TO LOWPOP
        MDRIS2[I,PARANO+1]=(YDRIS2[I]/(PARANO+1));
        MDRIS2[I, PARANO+2]=TOTMDRIS2[I]+(ABS(YDRIS2[I])/(PARANO+1))
        MDRIS2[I, PARANO+3]=LOW1[I, PARANO+1]
ENDFOR
SELE 16
APPEND FROM ARRAY MDRIS2
        COPY TO &T16 TYPE XLS
CLEAR ALL; CLEAR; CLOSE ALL
```

CHAPTER - IV

# CND Model (COMPOSITIONAL NUTRIENT DIAGNOSIS)

## i. Brief description of methodology – CND model

The data are processed similar way as that of DRIS for Compositional Nutritional Diagnosis (CND) approach (Parent and Dafir, 1992). In this model the full composition array for D nutrient compositions in plant tissues can be described by the following simplex $S^D$ contained to 100 %

$S^D = [(N,P,K,....R) : N > 0, P > O, K > 0.....R > 0; N + P + K + ..... + R = 100\%]$

Where 100 % is the dry matter content; N, P, K, are the nutrient concentrations and R is the filling value between 100 % and sum of the nutrients concentrations.

Nutrient concentration is corrected by geometric mean, G of all the D components including R.

$$G = (N \times P \times K \times ..... \times R)^{\frac{1}{D}}$$

The row centered log ratios are generated as follows :

$$V_N = \ln\left(\frac{N}{G}\right).....V_{Mn} = \ln\left(\frac{Mn}{G}\right)$$

The $V_N^*$ to $V_{Mn}^*$ and $SD_N^*$ to $SD_{Mn}^*$ are the CND norms (indicated by astrisks) *i.e.* mean and standard deviation of each row centered log ratio in the high yielding sub-population. The standardized variables $\frac{V_N - V_N^*}{SD_N^*}$ to $\frac{V_{Mn} - V_{Mn}^*}{SD_{Mn}^*}$ are the CND nutrient indices which are CND analogs of the DRIS nutrient indices

$$I_N = \frac{V_N - V_N^*}{SD_N^*}, ..... I_{Mn} = \frac{V_{Mn} - V_{Mn}^*}{SD_{Mn}^*}$$

Independent values for $V_N$ to $V_{Zn}$ were introduced in the equation for diagnostic purpose.

These following approaches are computed using the package developed in VISUAL FOXPRO 98, the algorithm of the program is given below.

### ii. Algorithm for CND nutrient model

SET ENVIRONMENT

INPUT VARIABLE DECLARATION

GET - Input file(source file : EXCEL input FILE), Sample number,
no of parameters/nutrients studied, Enter output file

CREATE database tables

DECLARE ARRAYS wholefile(sample number, total parmeters +1), class sum of square (sample no),
class sum of square1 (sample no), class sum of square2 (sample no),
class sum of square3 (sample no), class sum of square4 (sample no)....

DECLARE Memory variables

USE whole data file

APPEND FROM input file

COPY TO ARRAY whole file [sample number, total parmeters +1]

SORT ARRAY -descending order of yield data e.g. (ASORT (wholefile, 1, -1,1))

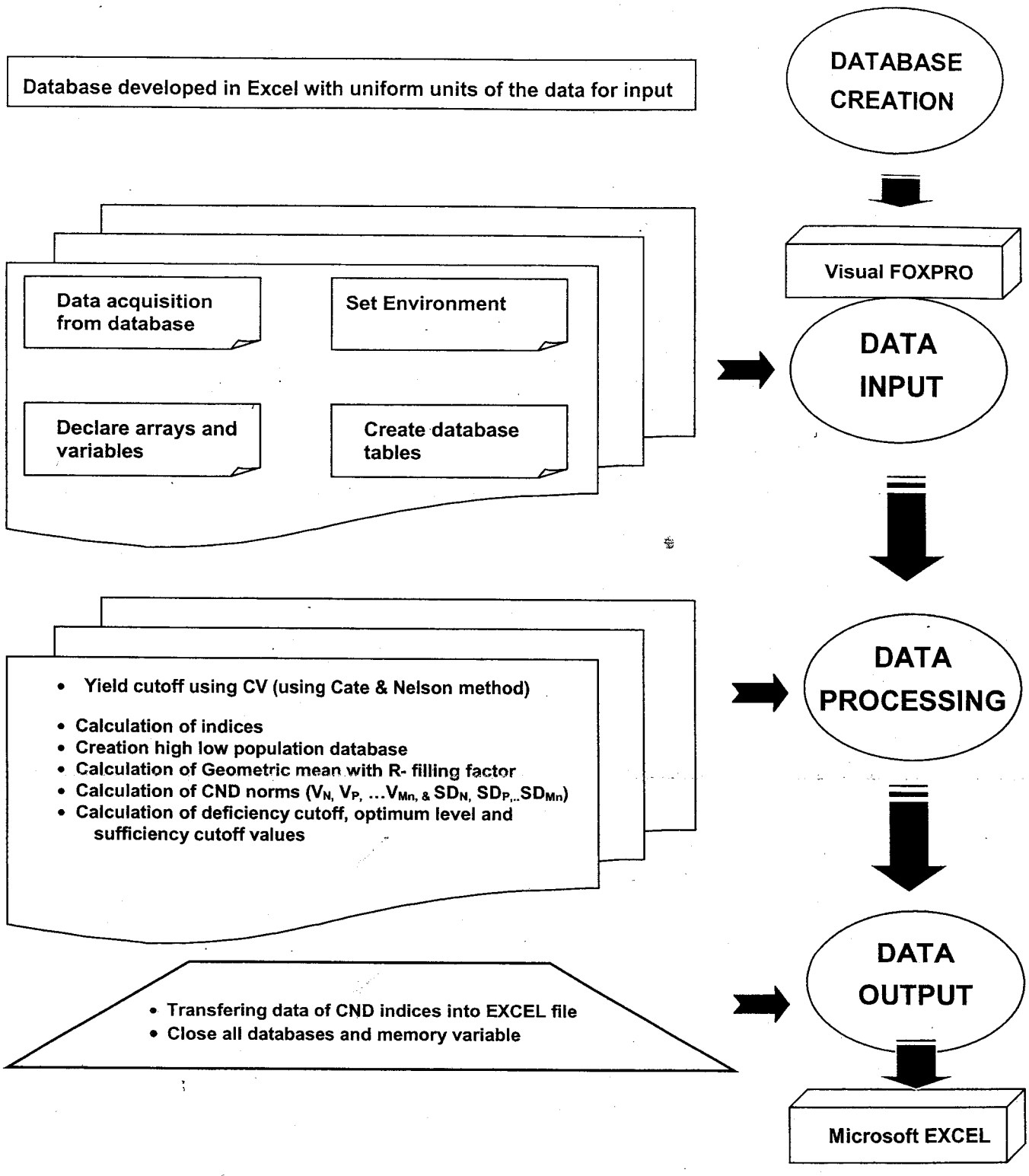
APPEND FROM ARRAY whole file database TO whole file array

- Critical value calculation for yield cut off using Cate and Nelson's statistical class sum of square technique
- Creation of high and low population nutrient data arrays based on the cut off derived from above method
- Calculation of Geometic mean (G) $=(N \times P \times K \times \ldots \times R)^{1/12}$ where R – filling value
- CND norms ($V_N$, $V_P$, ...$V_{Mn}$ and $SD_N$, $SD_P$....$SD_{Mn}$) were calculated
- CND indices ($I_N$, $I_P$, ...$I_{Mn}$) were computed which are analogs to DRIS nutrient indices
- Calculation, deficiency cut off, low level cut off, optimum level cut off, and high or sufficient level cutoff for both high and low population file
- Transferring data from array to subsequent DRIS index EXCEL files

CLOSE ALL DATABASES

CLEAR ALL memory variables

iii. Flow chart of the computer program – CND model



## iv. Computer program - CND model

```
*****************************************************************************************
                                SET ENVIRONMENT
*****************************************************************************************
CLEAR; CLEAR ALL; CLOSE ALL
SET SCOREBOARD OFF; SET STATUS OFF; SET TALK OFF; SET CONFIRM OFF; SET SAFETY OFF; CLOSE ALL; CLEAR

F1 = SPACE(20) ; T1=SPACE(20) ; T2=SPACE(20) ; T3=SPACE(20) ; T4=SPACE(20) ; T5=SPACE(20) ; T6=SPACE(20) ; T7=SPACE(20)
XX1=SPACE(40)
STORE 0 TO SAMPLENO, PARANO, INDNO

        @ 2, 10 SAY " FILE NAME"
        @ 2, 30 GET F1

        @4,10 SAY " ENTER TOTAL NUMBER OF SAMPLES"
        @4, 70 GET SAMPLENO

        @ 6, 10 SAY " ENTER NO OF PARAMETERS"
        @ 6, 60 GET PARANO
        @ 8, 10 SAY "ENTER OUTPUT FILE NAME"
        @ 8, 60 GET XX1
        READ
T1= ALLTRIM(XX1)+"CWF"
T6=ALLTRIM(XX1)+"CILN"
T7=ALLTRIM(XX1)+"CNORMS"

CREATE TABLE &T1 (Y N(6,2), N N(9,4), P N(9,4), K N(9,5), CA N(9,5), MG N(9,5), NA N(9,5), S N(9,5), B N(9,5), ZN N(9,5), CU N(9,5), FE N(9,5), MN N(9,5), FNO N(4),
                RQ N(4,2))
CREATE TABLE &T6 (N N(10,7), P N(10,7), K N(10,7), CA N(10,7), MG N(9,7), NA N(9,7), S N(9,7), B N(15,10), ZN N(15,10), CU N(15,10), FE N(15,10), MN N(15,10))
CREATE TABLE &T7 (MEAN N(9,4), DEF N(9,4), LOW N(9,5), MED N(9,5), EXC N(9,5))

        DECLARE DAT1 [SAMPLENO, 17], CSS1[SAMPLENO], CSS2[SAMPLENO], CSS[SAMPLENO], CSS3[SAMPLENO], CSS4[SAMPLENO], R2[SAMPLENO]
        DECLARE TX1[SAMPLENO], TX2[SAMPLENO]
        STORE 0 TO LLLL, SS1, SS2, SS3, SS4, SS9, SS10, CAL1, I, LL1, XX2, AA3, AA4, XX5, AA5, AA6, AA7, AA9, ZZZ, MAX1
        STORE 0 TO SD11, G11,SSS1, SUM1, VN1, L1,L2, KKK, LLLLL
SELECT 1
        USE &T1 ALIAS X1; ZAP
        APPEND FROM &F1 DELIMITED WITH TAB
        COPY TO ARRAY DAT1
        ASORT (DAT1, 1, -1,1)
        CLEAR; ZAP
        APPEND FROM ARRAY DAT1
CLOSE ALL
SELECT 6
        USE &T6 ALIAS X6; ZAP
SELE 7
        USE &T7 ALIAS X7; ZAP
*****************************************************************************************
                CALCULATION AND SELECTION OF CUT OFF FOR POPULATION
*****************************************************************************************
TT=0; SS=0
        FOR I=1 TO SAMPLENO-1
                TT=TT+DAT1[I,1]; SS=SS+(DAT1[I,1]*DAT1[I,1]); TX1[I]=TT; CSS1[I]=SS
        ENDFOR
```

```
        TT=TT+DAT1[SAMPLENO,1]; SS=SS+DAT1[SAMPLENO,1]*DAT1[SAMPLENO,1]; TX1[SAMPLENO]=TT
        CSS1[SAMPLENO]=SS; TSS=CSS1[SAMPLENO]-TX1[SAMPLENO]*TX1[SAMPLENO]/SAMPLENO

        FOR I=2 TO SAMPLENO-1
                TX2[I]=TX1[SAMPLENO]-TX1[I]; CSS2[I]=CSS1[SAMPLENO]-CSS1[I]; CSS1[I]=CSS1[I]-TX1[I]*TX1[I]/I
                CSS2[I]=CSS2[I]-TX2[I]*TX2[I]/(SAMPLENO-I); R2[I]=TSS-(CSS1[I]+CSS2[I]); R2[I]=R2[I]/TSS*100
        ENDFOR
CUT=1
        FOR I = 2 TO SAMPLENO-1
                IF R2[I+1] > R2[I]
                        CUT=CUT+1
                ELSE
        EXIT
                ENDIF
        ENDFOR
```

**********

**GETTING PARAMETERS IN ORDER**

**********

```
DECLARE HIGH1[CUT, PARANO], LOW1[SAMPLENO-CUT, PARANO]
LOWPOP=SAMPLENO-CUT; HIGHPOP=CUT
        FOR KKK= 1 TO PARANO
                FOR I = 1 TO HIGHPOP
                        HIGH1[I,KKK]=DAT1[I, KKK+1]
                ENDFOR
        ENDFOR
KKK=0; X=1
        FOR I= 1 TO PARANO
                FOR J= HIGHPOP + 1 TO SAMPLENO
                                LOW1[X,I]=DAT1[J,I+1]
                        IF X=SAMPLENO-(HIGHPOP)
                                X=1
                        ELSE
                                X=X+1
                        ENDIF
                ENDFOR
        ENDFOR

DECLARE SD1[HIGHPOP], SD2[LOWPOP], G1[HIGHPOP], G2[LOWPOP]

SD11=0; G11=1
FOR J= 1 TO HIGHPOP
                FOR I= 1 TO PARANO
                        SD11=SD11+HIGH1[J,I]; G11=G11*HIGH1[J,I]
                ENDFOR
                        SD1[J]=100-SD11; LLLLL=(1/(PARANO+1)); G1[J]=((SD1[J]*G11)^LLLLL); SD11=0; G11=1
ENDFOR
G11=1; SD11=0
FOR J= 1 TO LOWPOP
                FOR I= 1 TO PARANO; SD11=SD11+LOW1[J,I]; G11=G11*LOW1[J,I]
                ENDFOR
                SD2[J]=100-SD11; LLLLL=ABS(1/(PARANO+1)); G2[J]=((SD2[J]*G11)^LLLLL); G11=1; SD11=0
ENDFOR
```

**********

**CALCULATION CND INDICES**

**********

```
DECLARE SUMM1[PARANO], SSQ1[PARANO], VAR1[PARANO], CV1[PARANO], SSD1[PARANO], MEAN1[PARANO], VN[HIGHPOP, PARANO],
        VN2[LOWPOP, PARANO],  N1[HIGHPOP, PARANO],  N2[LOWPOP, PARANO], MN1[PARANO], VNNORM[PARANO, 6],
        PARAM1[PARANO, 5]
X=1; Y=0; SUM1=0
    FOR I= 1 TO PARANO
        FOR J= 1 TO HIGHPOP
            L1=HIGH1[J,I]/G1[J]; VN[J,I]=LOG(L1); SUM1=SUM1+VN[J,I]; SSS1=SSS1+(VN[J,I]*VN[J,I])
        ENDFOR
            SUMM1[I]=SUM1; MEAN1[I]=SUMM1[I]/HIGHPOP; SSQ1[I]=SSS1
            VAR1[I]=((HIGHPOP*SSQ1[I])-(SUMM1[I]*SUMM1[I]))/(HIGHPOP*(HIGHPOP-1))
            VAR1[I]=ABS(VAR1[I]); SSD1[I]=SQRT(VAR1[I]); CV1[I]=(SQRT(VAR1[I]))/MEAN1[I]
            SUM1=0; SSS1=0
        FOR J= 1 TO LOWPOP
            L2=LOW1[J,I]/G2[J]; VN2[J,I]=LOG(L2); SUM1=SUM1+VN2[J,I]; SSS1=SSS1+(VN2[J,I]* VN2[J,I])
        ENDFOR
        X=X+1; SUM1=0; SSS1=0
    ENDFOR

FOR I= 1 TO PARANO
        KKK=0
        FOR J= 1 TO HIGHPOP
            N1[J,I]=(( VN[J,I] - MEAN1[I]) / SSD1[I] ); KKK = KKK + G1[J]
        ENDFOR
        KKK=KKK / HIGHPOP
    VNNORM[I]=( MEAN1[I]); PARAM1[I, 1]=(2.72^VNNORM[I])*KKK; PARAM1[I, 2]=(2.72^(VNNORM[I]-(2.67*(SSD1[I]))))*KKK
    PARAM1[I, 3]=(2.72^(VNNORM[I]-(1.33*(SSD1[I]))))*KKK; PARAM1[I, 4]=(2.72^(VNNORM[I]+(1.33*(SSD1[I]))))*KKK
    PARAM1[I, 5]=(2.72^(VNNORM[I]+(2.67*(SSD1[I]))))*KKK
ENDFOR

FOR I= 1 TO PARANO
    FOR J= 1 TO LOWPOP
        N2[J,I]=(( VN2[J,I] - MEAN1[I]) / SSD1[I] )
    ENDFOR
ENDFOR

SELE 6
APPEND FROM ARRAY N2
COPY TO &T6 TYPE XLS
SELE 7
APPEND FROM ARRAY PARAM1
COPY TO &T7 TYPE XLS
USE; CLOSE ALL; CLEAR; RETURN
```

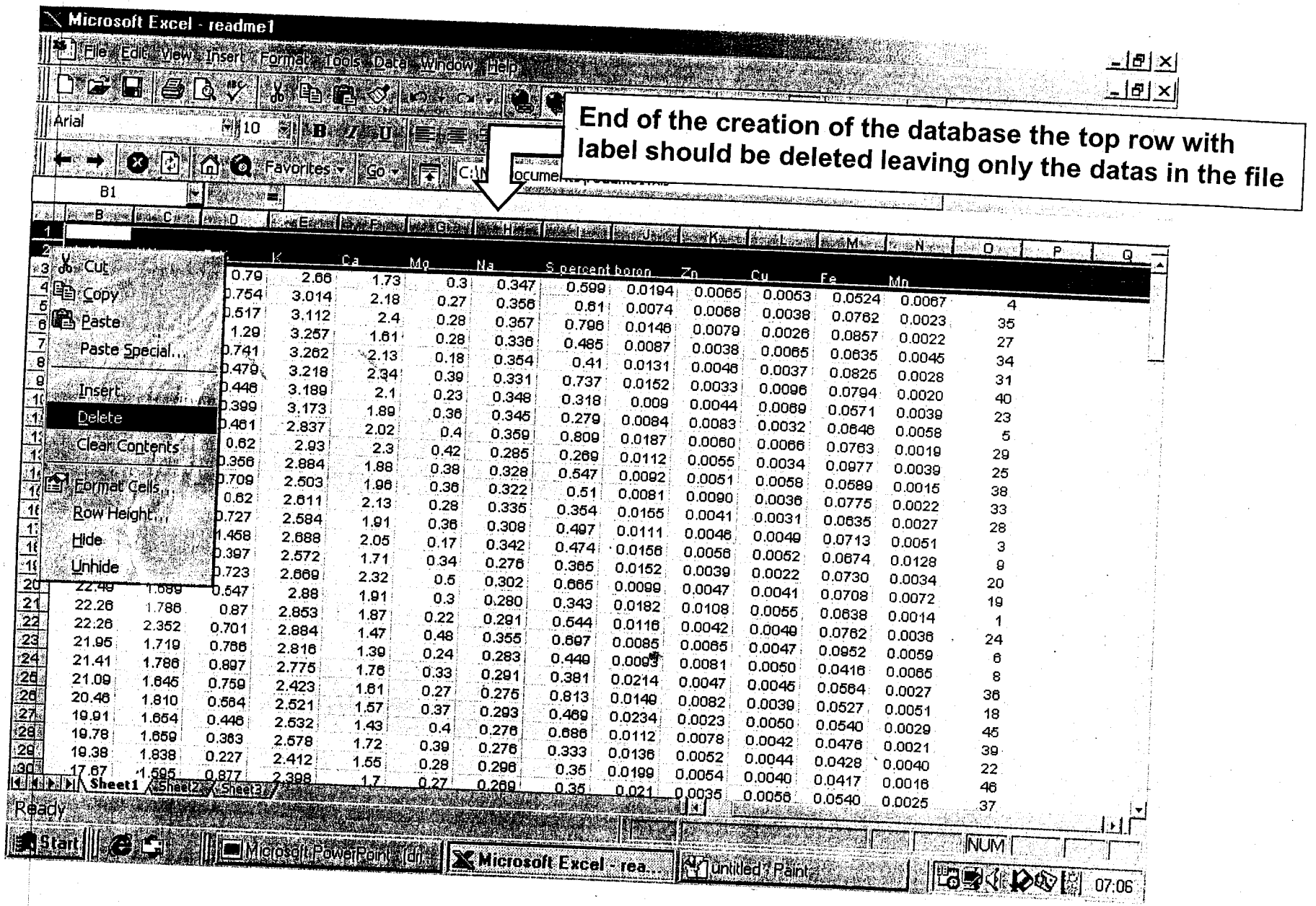
CHAPTER - V

# OUTPUT DISPLAY OF THE MODEL ON PROGRAM EXECUTION

Microsoft Excel - readme1

**Nutrient database is to be created in excel with common units as given in the figure**

| yield | N% | P % | K | Ca | Mg | Na | S percent | boron | Zn | Cu | Fe | Mn | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 31.75 | 2.284 | 0.79 | 2.66 | 1.73 | 0.3 | 0.347 | 0.599 | 0.0194 | 0.0065 | 0.0053 | 0.0524 | 0.0067 | 4 |
| 31.30 | 2.368 | 0.754 | 3.014 | 2.18 | 0.27 | 0.356 | 0.61 | 0.0074 | 0.0068 | 0.0038 | 0.0762 | 0.0023 | 35 |
| 29.75 | 2.372 | 0.517 | 3.112 | 2.4 | 0.28 | 0.357 | 0.796 | 0.0146 | 0.0079 | 0.0026 | 0.0857 | 0.0022 | 27 |
| 29.00 | 2.189 | 1.29 | 3.257 | 1.61 | 0.28 | 0.336 | 0.485 | 0.0087 | 0.0038 | 0.0065 | 0.0635 | 0.0045 | 34 |
| 28.85 | 2.344 | 0.741 | 3.262 | 2.13 | 0.18 | 0.354 | 0.41 | 0.0131 | 0.0046 | 0.0037 | 0.0825 | 0.0028 | 31 |
| 27.98 | 2.142 | 0.479 | 3.218 | 2.34 | 0.39 | 0.331 | 0.737 | 0.0152 | 0.0033 | 0.0096 | 0.0794 | 0.0020 | 40 |
| 27.83 | 2.297 | 0.446 | 3.189 | 2.1 | 0.23 | 0.348 | 0.318 | 0.009 | 0.0044 | 0.0069 | 0.0571 | 0.0039 | 23 |
| 26.76 | 2.267 | 0.399 | 3.173 | 1.89 | 0.36 | 0.345 | 0.279 | 0.0084 | 0.0083 | 0.0032 | 0.0646 | 0.0058 | 5 |
| 26.67 | 2.390 | 0.461 | 2.837 | 2.02 | 0.4 | 0.359 | 0.809 | 0.0187 | 0.0060 | 0.0066 | 0.0763 | 0.0019 | 29 |
| 26.63 | 2.174 | 0.62 | 2.93 | 2.3 | 0.42 | 0.285 | 0.269 | 0.0112 | 0.0055 | 0.0034 | 0.0977 | 0.0039 | 25 |
| 26.56 | 2.113 | 0.356 | 2.884 | 1.88 | 0.38 | 0.328 | 0.547 | 0.0092 | 0.0051 | 0.0058 | 0.0589 | 0.0015 | 38 |
| 26.46 | 2.061 | 0.709 | 2.503 | 1.96 | 0.36 | 0.322 | 0.51 | 0.0081 | 0.0090 | 0.0036 | 0.0775 | 0.0022 | 33 |
| 26.28 | 2.178 | 0.62 | 2.611 | 2.13 | 0.28 | 0.335 | 0.354 | 0.0155 | 0.0041 | 0.0031 | 0.0635 | 0.0027 | 28 |
| 24.32 | 1.936 | 0.727 | 2.584 | 1.91 | 0.36 | 0.308 | 0.497 | 0.0111 | 0.0046 | 0.0049 | 0.0713 | 0.0051 | 3 |
| 23.72 | 2.243 | 1.458 | 2.898 | 2.05 | 0.17 | 0.342 | 0.474 | 0.0156 | 0.0056 | 0.0052 | 0.0674 | 0.0128 | 9 |
| 23.50 | 1.661 | 0.397 | 2.572 | 1.71 | 0.34 | 0.276 | 0.365 | 0.0152 | 0.0039 | 0.0022 | 0.0730 | 0.0034 | 20 |
| 22.73 | 1.887 | 0.723 | 2.669 | 2.32 | 0.5 | 0.302 | 0.665 | 0.0099 | 0.0047 | 0.0041 | 0.0708 | 0.0072 | 19 |
| 22.49 | 1.689 | 0.547 | 2.88 | 1.91 | 0.3 | 0.280 | 0.343 | 0.0182 | 0.0108 | 0.0055 | 0.0638 | 0.0014 | 1 |
| 22.26 | 1.786 | 0.87 | 2.853 | 1.87 | 0.22 | 0.291 | 0.544 | 0.0116 | 0.0042 | 0.0049 | 0.0762 | 0.0036 | 24 |
| 22.26 | 2.352 | 0.701 | 2.884 | 1.47 | 0.48 | 0.355 | 0.697 | 0.0085 | 0.0065 | 0.0047 | 0.0952 | 0.0059 | 6 |
| 21.95 | 1.719 | 0.766 | 2.816 | 1.39 | 0.24 | 0.283 | 0.449 | 0.0093 | 0.0081 | 0.0050 | 0.0416 | 0.0065 | 8 |
| 21.41 | 1.786 | 0.897 | 2.775 | 1.76 | 0.33 | 0.291 | 0.381 | 0.0214 | 0.0047 | 0.0045 | 0.0564 | 0.0027 | 36 |
| 21.09 | 1.645 | 0.759 | 2.423 | 1.61 | 0.27 | 0.275 | 0.813 | 0.0149 | 0.0082 | 0.0039 | 0.0527 | 0.0051 | 18 |
| 20.46 | 1.810 | 0.564 | 2.521 | 1.57 | 0.37 | 0.293 | 0.469 | 0.0234 | 0.0023 | 0.0050 | 0.0540 | 0.0029 | 45 |
| 19.91 | 1.654 | 0.446 | 2.532 | 1.43 | 0.4 | 0.276 | 0.686 | 0.0112 | 0.0078 | 0.0042 | 0.0476 | 0.0021 | 39 |
| 19.78 | 1.659 | 0.363 | 2.578 | 1.72 | 0.39 | 0.276 | 0.333 | 0.0136 | 0.0052 | 0.0044 | 0.0428 | 0.0040 | 22 |
| 19.38 | 1.838 | 0.227 | 2.412 | 1.55 | 0.28 | 0.296 | 0.35 | 0.0199 | 0.0054 | 0.0040 | 0.0417 | 0.0016 | 46 |
| 17.67 | 1.595 | 0.877 | 2.398 | 1.7 | 0.27 | 0.289 | 0.35 | 0.021 | 0.0035 | 0.0056 | 0.0540 | 0.0025 | 37 |




End of the creation of the database the top row with label should be deleted leaving only the datas in the file

| K | Ca | Mg | Na | S percent | boron | Zn | Cu | Fe | Mn | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.79 | 2.66 | 1.73 | 0.3 | 0.347 | 0.599 | 0.0194 | 0.0065 | 0.0053 | 0.0524 | 0.0067 | 4 |
| 0.754 | 3.014 | 2.18 | 0.27 | 0.356 | 0.61 | 0.0074 | 0.0068 | 0.0039 | 0.0762 | 0.0023 | 35 |
| 0.517 | 3.112 | 2.4 | 0.28 | 0.357 | 0.796 | 0.0146 | 0.0079 | 0.0026 | 0.0857 | 0.0022 | 27 |
| 1.29 | 3.257 | 1.61 | 0.28 | 0.336 | 0.485 | 0.0087 | 0.0038 | 0.0065 | 0.0635 | 0.0045 | 34 |
| 0.741 | 3.262 | 2.13 | 0.18 | 0.354 | 0.41 | 0.0131 | 0.0046 | 0.0037 | 0.0825 | 0.0028 | 31 |
| 0.479 | 3.218 | 2.34 | 0.39 | 0.331 | 0.737 | 0.0152 | 0.0033 | 0.0096 | 0.0794 | 0.0020 | 40 |
| 0.446 | 3.189 | 2.1 | 0.23 | 0.348 | 0.318 | 0.009 | 0.0044 | 0.0069 | 0.0571 | 0.0039 | 23 |
| 0.399 | 3.173 | 1.89 | 0.36 | 0.345 | 0.279 | 0.0084 | 0.0083 | 0.0032 | 0.0646 | 0.0058 | 5 |
| 0.461 | 2.837 | 2.02 | 0.4 | 0.359 | 0.809 | 0.0187 | 0.0060 | 0.0066 | 0.0763 | 0.0019 | 29 |
| 0.62 | 2.93 | 2.3 | 0.42 | 0.285 | 0.269 | 0.0112 | 0.0055 | 0.0034 | 0.0977 | 0.0039 | 25 |
| 0.356 | 2.884 | 1.88 | 0.38 | 0.328 | 0.547 | 0.0092 | 0.0051 | 0.0058 | 0.0589 | 0.0015 | 38 |
| 0.709 | 2.503 | 1.96 | 0.36 | 0.322 | 0.51 | 0.0081 | 0.0090 | 0.0036 | 0.0775 | 0.0022 | 33 |
| 0.62 | 2.611 | 2.13 | 0.28 | 0.335 | 0.354 | 0.0155 | 0.0041 | 0.0031 | 0.0635 | 0.0027 | 28 |
| 0.727 | 2.584 | 1.91 | 0.36 | 0.308 | 0.497 | 0.0111 | 0.0046 | 0.0049 | 0.0713 | 0.0051 | 3 |
| 1.458 | 2.688 | 2.05 | 0.17 | 0.342 | 0.474 | 0.0156 | 0.0056 | 0.0052 | 0.0674 | 0.0128 | 9 |
| 0.397 | 2.572 | 1.71 | 0.34 | 0.276 | 0.365 | 0.0152 | 0.0039 | 0.0022 | 0.0730 | 0.0034 | 20 |
| 0.723 | 2.669 | 2.32 | 0.5 | 0.302 | 0.665 | 0.0099 | 0.0047 | 0.0041 | 0.0708 | 0.0072 | 19 |
| 0.547 | 2.88 | 1.91 | 0.3 | 0.280 | 0.343 | 0.0182 | 0.0108 | 0.0055 | 0.0638 | 0.0014 | 1 |
| 0.87 | 2.853 | 1.87 | 0.22 | 0.291 | 0.544 | 0.0116 | 0.0042 | 0.0049 | 0.0762 | 0.0036 | 24 |
| 0.701 | 2.884 | 1.47 | 0.48 | 0.355 | 0.697 | 0.0085 | 0.0065 | 0.0047 | 0.0952 | 0.0059 | 6 |
| 0.786 | 2.816 | 1.39 | 0.24 | 0.283 | 0.449 | 0.0093 | 0.0081 | 0.0050 | 0.0416 | 0.0065 | 8 |
| 0.897 | 2.775 | 1.76 | 0.33 | 0.291 | 0.381 | 0.0214 | 0.0047 | 0.0045 | 0.0564 | 0.0027 | 36 |
| 0.759 | 2.423 | 1.61 | 0.27 | 0.275 | 0.813 | 0.0149 | 0.0082 | 0.0039 | 0.0527 | 0.0051 | 18 |
| 0.564 | 2.521 | 1.57 | 0.37 | 0.293 | 0.469 | 0.0234 | 0.0023 | 0.0050 | 0.0540 | 0.0029 | 45 |
| 0.446 | 2.532 | 1.43 | 0.4 | 0.276 | 0.686 | 0.0112 | 0.0078 | 0.0042 | 0.0476 | 0.0021 | 39 |
| 0.363 | 2.578 | 1.72 | 0.39 | 0.276 | 0.333 | 0.0136 | 0.0052 | 0.0044 | 0.0428 | 0.0040 | 22 |
| 0.227 | 2.412 | 1.55 | 0.28 | 0.296 | 0.35 | 0.0199 | 0.0054 | 0.0040 | 0.0417 | 0.0016 | 46 |
| 0.877 | 2.398 | 1.7 | 0.27 | 0.269 | 0.35 | 0.021 | 0.0035 | 0.0056 | 0.0540 | 0.0025 | 37 |

| | | |
|---|---|---|
| 22.49 | 1.689 | |
| 22.26 | 1.786 | |
| 22.26 | 2.352 | |
| 21.95 | 1.719 | |
| 21.41 | 1.786 | |
| 21.09 | 1.645 | |
| 20.46 | 1.810 | |
| 19.91 | 1.654 | |
| 19.78 | 1.659 | |
| 19.38 | 1.838 | |
| 17.67 | 1.595 | |

**Check that the database holds the yield in the front (column A) and the survey lot no. in the last column.**
**To get the overall nutrient indices for the survey area, the mean of the survey data should be appended to the last row in the database as last record (it should be added to the no of samples)**

Microsoft Excel - re

File Edit View In

Arial

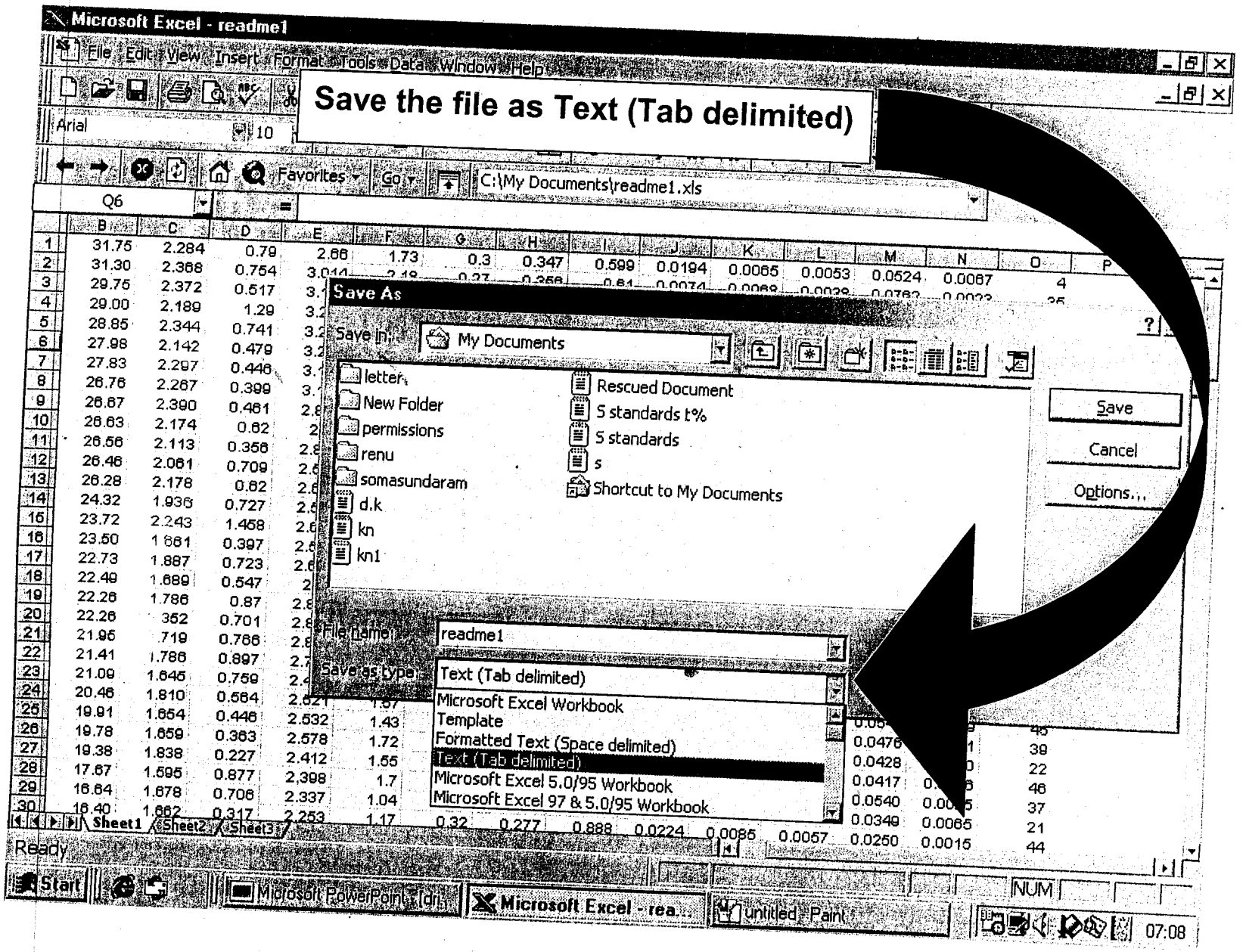
Q6

| | B | D | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 31.75 | 2.284 | 0.79 | 2.66 | 1.73 | 0.3 | 0.347 | 0.599 | 0.0194 | 0.0065 | 0.0053 | 0.0524 | 0.0067 | 4 | | |
| 2 | 31.30 | 2.368 | 0.754 | 3.014 | 2.18 | 0.27 | 0.356 | 0.61 | 0.0074 | 0.0068 | 0.0038 | 0.0762 | 0.0023 | 35 | | |
| 3 | 29.75 | 2.372 | 0.517 | 3.112 | 2.4 | 0.28 | 0.357 | 0.796 | 0.0146 | 0.0079 | 0.0026 | 0.0857 | 0.0022 | 27 | | |
| 4 | 29.00 | 2.189 | 1.29 | 3.257 | 1.61 | 0.28 | 0.336 | 0.485 | 0.0087 | 0.0038 | 0.0065 | 0.0635 | 0.0045 | 34 | | |
| 5 | 28.85 | 2.344 | 0.741 | 3.262 | 2.13 | 0.18 | 0.354 | 0.41 | 0.0131 | 0.0046 | 0.0037 | 0.0825 | 0.0028 | 31 | | |
| 6 | 27.98 | 2.142 | 0.479 | 3.218 | 2.34 | 0.39 | 0.331 | 0.737 | 0.0152 | 0.0033 | 0.0096 | 0.0794 | 0.0020 | 40 | | |
| 7 | 27.83 | 2.297 | 0.446 | 3.189 | 2.1 | 0.23 | 0.348 | 0.318 | 0.009 | 0.0044 | 0.0069 | 0.0571 | 0.0039 | 23 | | |
| 8 | 26.76 | 2.267 | 0.399 | 3.173 | 1.89 | 0.36 | 0.345 | 0.279 | 0.0084 | 0.0083 | 0.0032 | 0.0646 | 0.0058 | 5 | | |
| 9 | 26.67 | 2.390 | 0.461 | 2.837 | 2.02 | 0.4 | 0.359 | 0.809 | 0.0187 | 0.0060 | 0.0066 | 0.0763 | 0.0019 | 29 | | |
| 10 | 26.63 | 2.174 | 0.62 | 2.93 | 2.3 | 0.42 | 0.285 | 0.269 | 0.0112 | 0.0055 | 0.0034 | 0.0977 | 0.0039 | 25 | | |
| 11 | 26.56 | 2.113 | 0.356 | 2.884 | 1.88 | 0.38 | 0.328 | 0.547 | 0.0092 | 0.0051 | 0.0058 | 0.0589 | 0.0015 | 38 | | |
| 12 | 26.46 | 2.061 | 0.709 | 2.503 | 1.96 | 0.36 | 0.322 | 0.51 | 0.0081 | 0.0090 | 0.0036 | 0.0775 | 0.0022 | 33 | | |
| 13 | 26.28 | 2.178 | 0.62 | 2.611 | 2.13 | 0.28 | 0.335 | 0.354 | 0.0155 | 0.0041 | 0.0031 | 0.0635 | 0.0027 | 28 | | |
| 14 | 24.32 | 1.936 | 0.727 | 2.584 | 1.91 | 0.36 | 0.308 | 0.497 | 0.0111 | 0.0046 | 0.0049 | 0.0713 | 0.0051 | 3 | | |
| 15 | 23.72 | 2.243 | 1.458 | 2.688 | 2.05 | 0.17 | 0.342 | 0.474 | 0.0156 | 0.0056 | 0.0052 | 0.0674 | 0.0128 | 9 | | |
| 16 | 23.50 | 1.661 | 0.397 | 2.572 | 1.71 | 0.34 | 0.276 | 0.365 | 0.0152 | 0.0039 | 0.0022 | 0.0730 | 0.0034 | 20 | | |
| 17 | 22.73 | 1.887 | 0.723 | 2.669 | 2.32 | 0.5 | 0.302 | 0.665 | 0.0099 | 0.0047 | 0.0041 | 0.0708 | 0.0072 | 19 | | |
| 18 | 22.49 | 1.689 | 0.547 | 2.88 | 1.91 | 0.3 | 0.280 | 0.343 | 0.0182 | 0.0108 | 0.0055 | 0.0638 | 0.0014 | 1 | | |
| 19 | 22.26 | 1.786 | 0.87 | 2.853 | 1.87 | 0.22 | 0.291 | 0.544 | 0.0116 | 0.0042 | 0.0049 | 0.0762 | 0.0036 | 24 | | |
| 20 | 22.26 | 2.352 | 0.701 | 2.884 | 1.47 | 0.48 | 0.355 | 0.697 | 0.0085 | 0.0065 | 0.0047 | 0.0952 | 0.0059 | 6 | | |
| 21 | 21.95 | 1.719 | 0.766 | 2.816 | 1.39 | 0.24 | 0.283 | 0.449 | 0.0093 | 0.0081 | 0.0050 | 0.0416 | 0.0065 | 8 | | |
| 22 | 21.41 | 1.786 | 0.897 | 2.775 | 1.76 | 0.33 | 0.291 | 0.381 | 0.0214 | 0.0047 | 0.0045 | 0.0564 | 0.0027 | 36 | | |
| 23 | 21.09 | 1.645 | 0.759 | 2.423 | 1.61 | 0.27 | 0.275 | 0.813 | 0.0149 | 0.0082 | 0.0039 | 0.0527 | 0.0051 | 18 | | |
| 24 | 20.46 | 1.810 | 0.564 | 2.521 | 1.57 | 0.37 | 0.293 | 0.469 | 0.0234 | 0.0023 | 0.0050 | 0.0540 | 0.0029 | 45 | | |
| 25 | 19.91 | 1.654 | 0.446 | 2.532 | 1.43 | 0.4 | 0.276 | 0.686 | 0.0112 | 0.0078 | 0.0042 | 0.0476 | 0.0021 | 39 | | |
| 26 | 19.78 | 1.659 | 0.363 | 2.578 | 1.72 | 0.39 | 0.276 | 0.333 | 0.0136 | 0.0052 | 0.0044 | 0.0428 | 0.0040 | 22 | | |
| 27 | 19.38 | 1.838 | 0.227 | 2.412 | 1.55 | 0.28 | 0.296 | 0.35 | 0.0199 | 0.0054 | 0.0040 | 0.0417 | 0.0016 | 46 | | |
| 28 | 17.67 | 1.595 | 0.877 | 2.398 | 1.7 | 0.27 | 0.269 | 0.35 | 0.021 | 0.0035 | 0.0056 | 0.0540 | 0.0025 | 37 | | |
| 29 | 16.64 | 1.678 | 0.706 | 2.337 | 1.04 | 0.22 | 0.278 | 0.649 | 0.0133 | 0.0059 | 0.0051 | 0.0349 | 0.0065 | 21 | | |
| 30 | 16.40 | 1.662 | 0.317 | 2.253 | 1.17 | 0.32 | 0.277 | 0.888 | 0.0224 | 0.0085 | 0.0057 | 0.0250 | 0.0015 | 44 | | |

Sheet1 Sheet2 Sheet3

Ready NUM

Start Microsoft PowerPoint - [dri... Microsoft Excel - rea... untitled - Paint 07:06


Save the file as Text (Tab delimited)
Microsoft Excel - readme1
File Edit View Insert Format Tools Data Window Help
Arial
10
Favorites
Go
C:\My Documents\readme1.xls
Q6
Save As
Save in:
My Documents
letter
New Folder
permissions
renu
somasundaram
d.k
kn
kn1
Rescued Document
S standards t%
S standards
s
Shortcut to My Documents
Save
Cancel
Options...
File name:
readme1
Save as type:
Text (Tab delimited)
Microsoft Excel Workbook
Template
Formatted Text (Space delimited)
Text (Tab delimited)
Microsoft Excel 5.0/95 Workbook
Microsoft Excel 97 & 5.0/95 Workbook
Sheet1
Sheet2
Sheet3
Ready
NUM
Start
Microsoft PowerPoint - [dr...
Microsoft Excel - rea...
untitled - Paint
07:08

The data file will be a text file with the order of data as given below (but without the header row)

data file - Notepad

File Edit Search Help

| yield | N | P | K | Ca | Mg | Na | S | B | Zn | Cu | Fe | Mn |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 22.49 | 1.69 | 0.55 | 2.88 | 1.91 | 0.30 | 0.28 | [illegible] | 0.0182 | [illegible] | | | |
| 12.79 | 1.37 | 0.53 | 2.04 | 1.39 | 0.24 | [illegible] | [illegible] | | | | | |
| 24.32 | 1.94 | 0.73 | 2.58 | 1.91 | 0.36 | [illegible] | | | [illegible] | 0.0055 | 0.0638 | 0. |
| 31.75 | 2.28 | 0.79 | 2.66 | 1.73 | [illegible] | | | [illegible] | 0.0096 | 0.0024 | 0.0361 | 0. |
| 26.76 | 2.27 | [illegible] | 3.17 | 1.89 | [illegible] | | [illegible] | 0.0111 | 0.0046 | 0.0049 | 0.0713 | 0. |
| 22.26 | 2.35 | [illegible] | 2.88 | 1.47 | | | 0.60 | 0.0194 | 0.0065 | 0.0053 | 0.0524 | 0. |
| | | | 2.01 | | | 0.34 | 0.28 | 0.0084 | 0.0083 | 0.0032 | 0.0646 | 0. |
| | | | [illegible] | | 0.48 | 0.35 | 0.70 | 0.0085 | 0.0065 | 0.0047 | 0.0952 | 0. |
| | | | | [illegible] | 0.20 | 0.23 | 0.55 | 0.0140 | 0.0058 | 0.0020 | 0.0254 | 0. |
| | | | [illegible] | 1.39 | 0.24 | 0.28 | 0.45 | 0.0093 | 0.0081 | 0.0050 | 0.0416 | 0. |
| | | | 2.69 | 2.05 | 0.17 | 0.34 | 0.47 | 0.0156 | 0.0056 | 0.0052 | 0.0674 | 0. |
| | | | 2.06 | 1.05 | 0.40 | 0.25 | 0.63 | 0.0139 | 0.0076 | 0.0035 | 0.0227 | 0. |
| [illegible] | [illegible] | 1.06 | 2.27 | 1.22 | 0.39 | 0.26 | 0.54 | 0.0141 | 0.0086 | 0.0046 | 0.0487 | 0. |
| [illegible] | [illegible] | 1.01 | 2.04 | 1.41 | 0.24 | 0.24 | 0.33 | 0.0144 | 0.0052 | 0.0029 | 0.0328 | 0. |
| [illegible] | 1.29 | 0.47 | 2.08 | 1.74 | 0.38 | 0.23 | 0.67 | 0.0114 | 0.0077 | 0.0045 | 0.0526 | 0. |
| [illegible] | 1.44 | 0.45 | 2.10 | 1.81 | 0.38 | 0.25 | 0.71 | 0.0212 | 0.0057 | 0.0028 | 0.0690 | 0. |
| [illegible] | 1.50 | 0.43 | 2.16 | 1.06 | 0.22 | 0.26 | 0.69 | 0.0142 | 0.0052 | 0.0033 | 0.0738 | 0. |
| [illegible] | 1.27 | 0.58 | 2.01 | 1.95 | 0.41 | 0.23 | 0.19 | 0.0193 | 0.0057 | 0.0022 | 0.0237 | 0. |
| 15.90 | 1.58 | 0.73 | 2.25 | 1.42 | 0.32 | 0.27 | 0.58 | 0.0211 | 0.0065 | 0.0039 | 0.0935 | 0. |
| [illegible] | 1.64 | [illegible] | 2.42 | 1.61 | 0.27 | 0.27 | 0.81 | 0.0149 | 0.0082 | 0.0039 | 0.0527 | 0. |
| [illegible] | 1.89 | [illegible] | 2.67 | 2.32 | 0.50 | 0.30 | 0.67 | 0.0099 | 0.0047 | 0.0041 | 0.0708 | 0. |
| 23.50 | 1.66 | 0.40 | 2.57 | 1.71 | 0.34 | 0.28 | 0.37 | 0.0152 | 0.0039 | 0.0022 | 0.0730 | 0. |
| 16.64 | 1.68 | 0.71 | 2.34 | 1.04 | 0.22 | 0.28 | 0.65 | 0.0133 | 0.0059 | 0.0051 | 0.0349 | 0. |
| 19.78 | 1.66 | 0.36 | 2.58 | 1.72 | 0.39 | 0.28 | 0.33 | 0.0136 | 0.0052 | 0.0044 | 0.0428 | 0. |
| 27.83 | 2.30 | 0.45 | 3.19 | 2.1 | 0.23 | 0.35 | 0.32 | 0.0090 | 0.0044 | 0.0069 | 0.0571 | 0. |
| 22.26 | 1.79 | 0.87 | 2.85 | 1.87 | 0.22 | 0.29 | 0.54 | 0.0116 | 0.0042 | 0.0049 | 0.0762 | 0. |
| 26.63 | 2.17 | 0.62 | 2.93 | 2.3 | 0.42 | 0.29 | 0.27 | 0.0112 | 0.0055 | 0.0034 | 0.0977 | 0. |
| 15.26 | 1.60 | 0.61 | 2.19 | 0.99 | 0.31 | 0.27 | 0.66 | 0.0232 | 0.0055 | 0.0024 | 0.0377 | 0. |
| 29.75 | 2.37 | 0.52 | 3.11 | 2.4 | 0.28 | 0.36 | 0.80 | 0.0146 | 0.0079 | 0.0026 | 0.0857 | 0. |
| 26.28 | 2.18 | 0.62 | 2.61 | 2.13 | 0.28 | 0.33 | 0.35 | 0.0155 | 0.0041 | 0.0031 | 0.0635 | 0. |
| 26.67 | 2.39 | 0.46 | 2.84 | 2.02 | 0.40 | 0.36 | 0.81 | 0.0187 | 0.0060 | 0.0066 | 0.0763 | 0. |
| 15.97 | 1.54 | 0.69 | 2.26 | 1.17 | 0.26 | 0.26 | 0.54 | 0.0108 | 0.0056 | 0.0038 | 0.0825 | 0. |
| 28.85 | 2.34 | 0.74 | 3.26 | 2.13 | 0.18 | 0.35 | 0.41 | 0.0131 | 0.0046 | 0.0037 | 0.0825 | 0. |
| 15.01 | 1.31 | 0.55 | 2.14 | 1.38 | 0.37 | 0.24 | 0.55 | 0.0182 | 0.0063 | 0.0019 | 0.0454 | 0. |
| 26.46 | 2.06 | 0.71 | 2.50 | 1.96 | 0.36 | 0.32 | 0.51 | 0.0081 | 0.0090 | 0.0036 | 0.0775 | 0. |

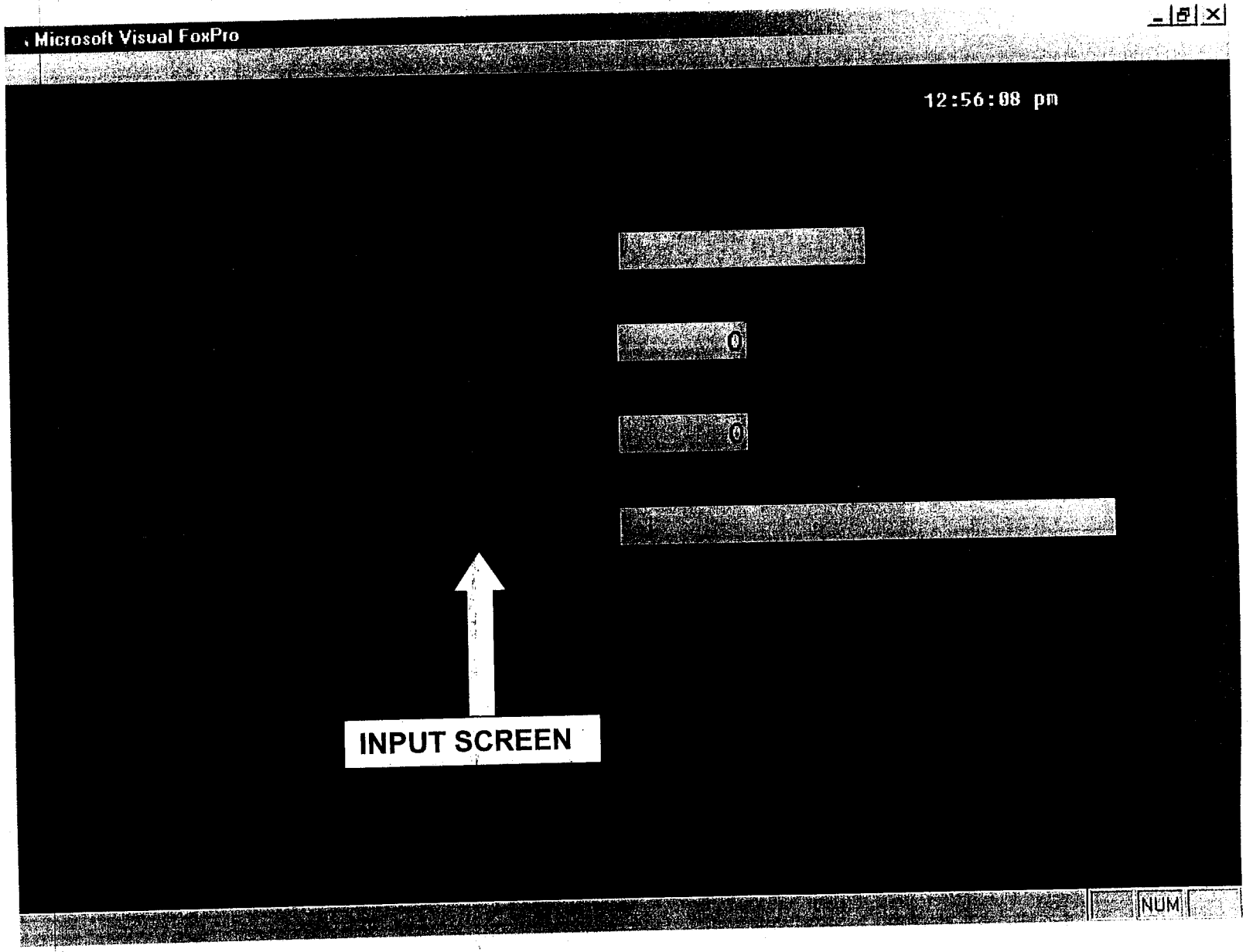
Note: Yield val[illegible] column

data file - Notepad

File Edit Search Help

| P | K | Ca | Mg | Na | S | B | Zn | Cu | Fe | Mn | Field no |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.55 | 2.88 | 1.91 | 0.30 | 0.28 | [illegible] | 0.0182 | 0.0108 | 0.0055 | 0.0638 | 0.0014 | 1 |
| 0.53 | 2.04 | 1.39 | 0.24 | 0.24 | [illegible] | 0.0158 | 0.0096 | 0.0024 | 0.0361 | 0.0022 | 2 |
| 0.73 | 2.58 | 1.91 | 0.36 | [illegible] | 0.50 | 0.0111 | 0.0046 | 0.0049 | 0.0713 | 0.0051 | 3 |
| 0.79 | 2.66 | 1.73 | 0.30 | [illegible] | 0.60 | 0.0194 | 0.0065 | 0.0053 | 0.0524 | 0.0067 | 4 |
| 0.40 | 3.17 | 1.89 | [illegible] | [illegible] | 0.28 | 0.0084 | 0.0083 | 0.0032 | 0.0646 | 0.0058 | 5 |
| 0.70 | 2.88 | 1.47 | [illegible] | 0.35 | 0.70 | 0.0085 | 0.0065 | 0.0047 | 0.0952 | 0.0059 | 6 |
| 0.78 | 2.01 | [illegible] | 0.20 | 0.23 | 0.55 | 0.0140 | 0.0058 | 0.0020 | 0.0254 | 0.0058 | 7 |
| 0.77 | 2.82 | [illegible] | 0.24 | 0.28 | 0.45 | 0.0093 | 0.0081 | 0.0050 | 0.0416 | 0.0065 | 8 |
| 1.46 | 2.62 | 2.05 | 0.17 | 0.34 | 0.47 | 0.0156 | 0.0056 | [illegible] | [illegible] | [illegible] | [illegible] |
| 0.70 | [illegible] | 1.05 | 0.40 | 0.25 | 0.63 | 0.0139 | 0.0076 | [illegible] | | | |
| 1.06 | [illegible] | 1.22 | 0.39 | 0.26 | 0.54 | 0.0141 | 0.0086 | [illegible] | | | |
| 1.01 | 2.04 | [illegible] | 0.24 | 0.24 | 0.33 | 0.0144 | 0.0052 | [illegible] | | | |
| [illegible] | [illegible] | 1.74 | 0.38 | 0.23 | 0.67 | 0.0114 | 0.0077 | [illegible] | | | |
| [illegible] | 2.10 | 1.81 | 0.38 | 0.25 | 0.71 | 0.0212 | 0.0057 | 0.0028 | 0.0690 | 0.0070 | 14 |
| [illegible] | 2.16 | 1.06 | 0.22 | 0.26 | 0.69 | 0.0142 | 0.0052 | 0.0033 | 0.0738 | 0.0036 | 15 |
| [illegible] | 2.01 | 1.95 | 0.41 | 0.23 | 0.19 | 0.0193 | 0.0057 | 0.0022 | 0.0237 | 0.0022 | 16 |
| 0.73 | 2.25 | 1.42 | 0.32 | 0.27 | 0.58 | 0.0211 | 0.0065 | 0.0039 | 0.0935 | 0.0014 | 17 |
| 0.76 | 2.42 | 1.61 | 0.27 | 0.27 | 0.81 | 0.0149 | 0.0082 | 0.0039 | 0.0527 | 0.0051 | 18 |
| 0.72 | 2.67 | 2.32 | 0.50 | 0.30 | 0.67 | 0.0099 | 0.0047 | 0.0041 | 0.0708 | 0.0072 | 19 |
| 0.40 | 2.57 | 1.71 | 0.34 | 0.28 | 0.37 | 0.0152 | 0.0039 | 0.0022 | 0.0730 | 0.0034 | 20 |
| 0.71 | 2.34 | 1.04 | 0.22 | 0.28 | 0.65 | 0.0133 | 0.0059 | 0.0051 | 0.0349 | 0.0065 | 21 |
| 0.36 | 2.58 | 1.72 | 0.39 | 0.28 | 0.33 | 0.0136 | 0.0052 | 0.0044 | 0.0428 | 0.0040 | 22 |
| 0.45 | 3.19 | 2.1 | 0.23 | 0.35 | 0.32 | 0.0090 | 0.0044 | 0.0069 | 0.0571 | 0.0039 | 23 |
| 0.87 | 2.85 | 1.87 | 0.22 | 0.29 | 0.54 | 0.0116 | 0.0042 | 0.0049 | 0.0762 | 0.0036 | 24 |
| 0.62 | 2.93 | 2.3 | 0.42 | 0.29 | 0.27 | 0.0112 | 0.0055 | 0.0034 | 0.0977 | 0.0039 | 25 |
| 0.61 | 2.19 | 0.99 | 0.31 | 0.27 | 0.66 | 0.0232 | 0.0055 | 0.0024 | 0.0377 | 0.0079 | 26 |
| 0.52 | 3.11 | 2.4 | 0.28 | 0.36 | 0.80 | 0.0146 | 0.0079 | 0.0026 | 0.0857 | 0.0022 | 27 |
| 0.62 | 2.61 | 2.13 | 0.28 | 0.33 | 0.35 | 0.0155 | 0.0041 | 0.0031 | 0.0635 | 0.0027 | 28 |
| 0.46 | 2.84 | 2.02 | 0.40 | 0.36 | 0.81 | 0.0187 | 0.0060 | 0.0066 | 0.0763 | 0.0019 | 29 |
| 0.69 | 2.26 | 1.17 | 0.26 | 0.26 | 0.54 | 0.0108 | 0.0056 | 0.0038 | 0.0825 | 0.0028 | 30 |
| 0.74 | 3.26 | 2.13 | 0.18 | 0.35 | 0.41 | 0.0131 | 0.0046 | 0.0037 | 0.0825 | 0.0028 | 31 |
| 0.55 | 2.14 | 1.38 | 0.37 | 0.24 | 0.55 | 0.0182 | 0.0063 | 0.0019 | 0.0454 | 0.0017 | 32 |
| 0.71 | 2.50 | 1.96 | 0.36 | 0.32 | 0.51 | 0.0081 | 0.0090 | 0.0036 | 0.0775 | 0.0022 | 33 |

**Note: Field no. (sample no.) at last of all variable**


Microsoft Visual FoxPro
12:56:08 pm
0
0
NUM
INPUT SCREEN

Microsoft Visual FoxPro

**COUNT INCLUDES THE AVERAGE SURVEY DATA**

trail

46

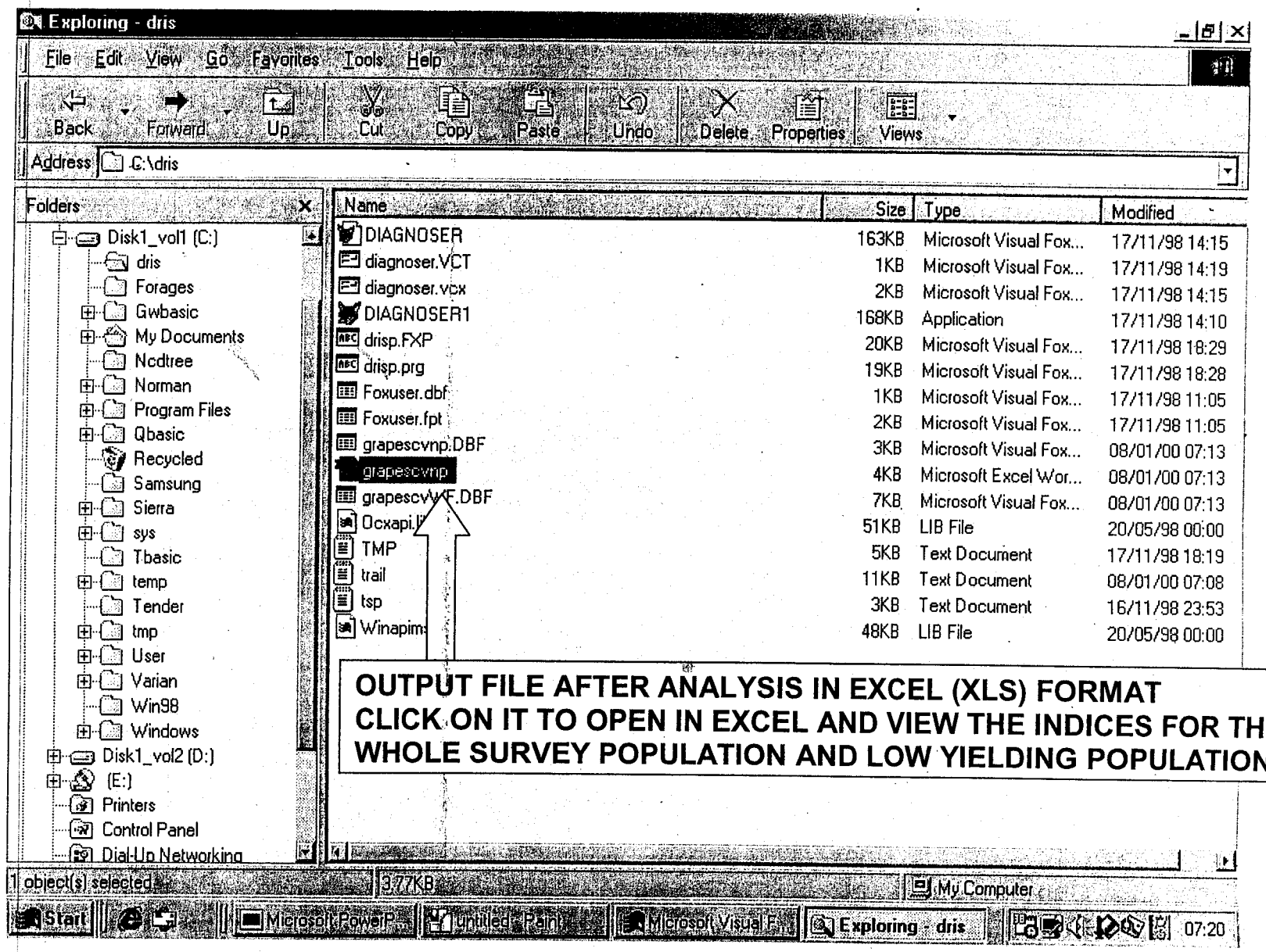
12

grapes

* ONLY THE NUTRIENT PARAMETERS TO BE COUNTED AND ENTERED
* THE YIELD AND FIELD NO WILL BE ACCOUNTED AUTOMATICALLY

NUM

Start | Microsoft PowerPoint - [Di... | untitled - Paint | Microsoft Visual FoxPro | 07:12


Exploring - dris
File Edit View Go Favorites Tools Help
Back Forward Up Cut Copy Paste Undo Delete Properties Views
Address C:\dris
Folders
Disk1_vol1 (C:)
dris
Forages
Gwbasic
My Documents
Ncdtree
Norman
Program Files
Qbasic
Recycled
Samsung
Sierra
sys
Tbasic
temp
Tender
tmp
User
Varian
Win98
Windows
Disk1_vol2 (D:)
(E:)
Printers
Control Panel
Dial-Up Networking
Name Size Type Modified
DIAGNOSER 163KB Microsoft Visual Fox... 17/11/98 14:15
diagnoser.VCT 1KB Microsoft Visual Fox... 17/11/98 14:19
diagnoser.vcx 2KB Microsoft Visual Fox... 17/11/98 14:15
DIAGNOSER1 168KB Application 17/11/98 14:10
drisp.FXP 20KB Microsoft Visual Fox... 17/11/98 18:29
drisp.prg 19KB Microsoft Visual Fox... 17/11/98 18:28
Foxuser.dbf 1KB Microsoft Visual Fox... 17/11/98 11:05
Foxuser.fpt 2KB Microsoft Visual Fox... 17/11/98 11:05
grapescvnp.DBF 3KB Microsoft Visual Fox... 08/01/00 07:13
grapescvnp 4KB Microsoft Excel Wor... 08/01/00 07:13
grapescvWF.DBF 7KB Microsoft Visual Fox... 08/01/00 07:13
Ocxapi.lib 51KB LIB File 20/05/98 00:00
TMP 5KB Text Document 17/11/98 18:19
trail 11KB Text Document 08/01/00 07:08
tsp 3KB Text Document 16/11/98 23:53
Winapim 48KB LIB File 20/05/98 00:00
OUTPUT FILE AFTER ANALYSIS IN EXCEL (XLS) FORMAT
CLICK ON IT TO OPEN IN EXCEL AND VIEW THE INDICES FOR THE
WHOLE SURVEY POPULATION AND LOW YIELDING POPULATION
1 object(s) selected
3.77KB
My Computer
Start
Microsoft PowerP...
untitled - Paint
Microsoft Visual F...
Exploring - dris
07:20

CHAPTER - VI

# REFERENCES

Beaufils, E.R. 1973. Diagnosis and Recommendation Integrated System (DRIS). Soil Sci. Bull. Vol. 1, p. 1-132

Cate, R.B. and Nelson, L.A. 1971. A simple statistical procedure for partitioning soil test correlation data into two classes. **Soil Sci. Soc. Amer. Proc. 35:** 658-660.

Microsoft. 1997. Microsoft Excel 97 and Microsoft Visual Foxpro 97, Incline Village, NV.

Parent, L. E. and M. Dafir. 1992. A Theoretical concept of Compositional Nutrient Diagnosis. J. Amer. Soc. Hort. Sci., 117(2): 239-242.

Walworth, J.L. and M.E. Sumner. 1987. The Diagnosis and recommendation Integrated System (DRIS). **In:** Ad. Soil Sci., Vol 6. Ed. B.A. Stewart. pp 149-188.