A MATCHING SCHEME FOR AQUACULTURE; A GRAPHING CALCULATOR APPROACH

I. K. Dontwi[1], F. O. Boateng[2] and E. Owusu-Ansah[1]
[1]Department of Mathematics, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana.
[2]Department of Information Technology Education, University of Education Winneba-Kumasi Campus, Ghana

ABSTRACT
Graphing Calculator Technology possesses a lot of capabilities in solving a variety of scientific and non-scientific problems. Its programming abilities make it very flexible to use. Unfortunately, it has been underutilised. In this paper, we develop an application with the graphing calculator to handle the processes in a Matching Scheme for Assigning Fish Types to Ponds. The application gives an inclusion matrix which is used as the basis of the matching and finally gives summary of the assignment.

Keywords: *Utility function, Inclusion matrix*, *Total matching, Partial matching.*

INTRODUCTION

A matching scheme for assigning fish types to ponds is a mathematical model that has been developed to match fish types (or species) on the basis of fish survivorship in relation to type of ponds and the water quality variables (Dontwi *et al*., 2005). This mathematical tool has been developed at a time the Government is exploiting all avenues to help boost the production of fish through marine fishing and fish pond culture.

Practically, it is tedious to go through the matching process described by the model manually. One is prone to numerical errors and more so a little oversight will render the whole process invalid. It is therefore imperative to look at ways that will help enhance the effective use of this model.

The use of computer software and graphing calculator technology are two techniques used in this direction. Graphing calculator in this case is preferred to the computer due to the fact that the graphing calculator:

i) is very portable since it can easily be carried around any part of the country irrespective of the weather condition.
ii) runs on dry cells and therefore will not need electricity.
iii) is simple to use.

Though, some effort has been made to exploit the capabilities of the graphing calculator technology (Dunham, 2000; Kutzler, 2000; Benjamin, 1997), we demonstrate how the TI 89 Graphing Calculator could be used for this task.

The Matching Scheme

We quickly go through the model.

Preliminaries and Notations

- P the set of ponds (|P| - the number of ponds);
- F the set of fish types (|F| - the number of fish types);
- NE the number of ordered list of elements or variables;
- W measure of importance, measured through weightings on elements for fish types;
- H the utility function for fish pond elements;
- S the survivorship of fish in ponds;
- PE the elements in ponds; and
- FE the elements in fish.

Crude Measure per Element

Measurement will be based on whether a match is totally satisfied or partially satisfied or not satisfied. For a given fish, i, the element, k, is matched to pond, j, of the corresponding element.

If the outcome is a total matching, then it means the element range of the pond, PE^k_j falls completely in the element range of the fish, FE^k_j. That is $PE^k_j \subseteq FE^k_j$. We shall refer to an outcome as partially matching if matching is within an acceptable range of violation in respect to the ideal standard for an element, k. The matching function will be represented by the function

where V_{ij}^k is the acceptable violation for element, k of fish, i based on pond, j qualities, that is the threshold.

Utility Function

In order to measure the extent of contribution or the importance of any particular element, k, we give it a weighting, w_i^k, that is the importance of element, k for fish, i. We are, therefore, in the position to have a weighted measure per element which we denote as the Utility Function, h_{ij}^k, for fish, i, pond, j and element, k; such that

$$h^k_{ij} = w^k_i \cdot g^k_{ij}. \quad \text{….. (1)}$$

The Inclusion Matrix C(i, j)

We shall consider the cumulative effect of the matching. This then could form the basis for the formation of an inclusion matrix. This matrix will tell us the types of fish, i going into pond j.

Let $$S_{ij} = \sum_{k}^{NE(i)} w^k_i \cdot g^k_{ij} \quad \text{….. (2)}$$

If our weightings are independent of i then $w^k_i = w^k$
In the case if total matching, this reduces
to

$$S_{ij} = \sum_{k}^{NE(i)} w^k_i \cdot 1 = \tilde{w}_i, \text{ where } \tilde{w}_i \equiv NE(i), i \in F. \quad \text{…. (3)}$$

If $S_{ij} = \tilde{w}_i$ then fish, i can go into pond j (that is totally satisfying). What happens if $S_{ij} < \tilde{w}_i$?

In this case i might go into j. It is, therefore, necessary to examine, for example, the number of

$$g\left(FE^k_i, PE^k_j\right) = g^k_{ij} = \begin{cases} 1, & \text{for total matching, that is, } PE^k_j \subseteq FE^k_i; \\ \Delta^k_{ij}, & \text{for partial matching, that is } \left|FE^k_i - PE^k_j\right| \le V^k_{ij}; \\ 0, & \text{for no matching} \end{cases}$$

elements not fully satisfied and the extent to which it missed. This, therefore, brings to bare a threshold value, say, $\hat{w}_i$ below which a fish is to be rejected or accepted into a particular pond.

THE GRAPHING CALCULATOR CONCEPT

Implementation

We demonstrate the program by matching three (3) fish species to four (4) ponds on the basis of their water quality variables. The water quality variables considered are:

1. Temperature (°C)
2. pH
3. Salinity (ppt)
4. Dissolved Oxygen, DO (ppm)

The necessary data are given as follows:

Table 1: Fish Types and their respective Fish Element Values

Fish Type	Fish No.	Fish Element (FE_i^k) Temp. (°C)	pH	Salinity (ppt)	DO (ppm)	Hardness (mg/L)
Clarias gariepinus African catfish	F1	12 - 32	6.5–8	2 – 11	2 – 13	32 – 65
Sarotherodon melanotheron melanotheron Blackchin tilapia	F2	23 - 25	7-8	0 – 5	3 – 10	25 – 80
Cyprinus carpio carpio Common carp	F3	8 - 30	7-7.5	0 – 9	2 -16	25 – 80

Source: *Froese and Pauly, 2005.*

Table 2: Ponds and their respective Pond Element Values

Town/Name of Farmer	Pond No.	Pond Element (PE_j^k) Temp. (°C)	pH	Salinity (ppt)	DO (ppm)	Hardness (mg/L)
Anwomaso-Domiabra/ Nana Siaw	P1	27	8.1	1.5	11	54
	P2	27	7.0	1.5	13.2	52
	P3	27	7.2	1.5	10	34
	P4	27	8.0	1.4	12	40

Source: *Dontwi, 2005.*

Table 3: Values of the Acceptable Range of Violation (V_{ij}^k) for the Fish Types

Fish No.	Temp. (°C)	pH	Salinity (ppt)	DO (ppm)	Hardness (mg/L)
F1	4	0.2	0.5	0.2	5
F2	2	0.2	0.1	0.5	5
F3	5	0.1	0.2	1	5
F4	5	0.2	0.5	0.5	2
F5	2	0.3	0.1	1	5

Source: *Froese and Pauly, 2005.*

Table 4: Weightings of the Fish Elements

Fish Elements	Weightings, w^k
Temperature (°C)	0.10
pH	0.25
Salinity (ppt)	0.40
Dissolved Oxygen (ppm)	0.15
Hardness (mg/L)	0.10

Source: *Boateng, 2006*

HOW THE PROGRAM WORKS

This program basically formulates a matrix called the inclusion matrix, which represents the matching. The program also goes further to give the summary of the matching based on the inclusion matrix. A complete program source code is found in the Appendix.

Data Input and Editing

It is important to note that before running the program, data to be used should be entered carefully. The data for fish types are entered as matrices with two columns. Since the data for fish elements (Table 1) are in ranges, the first column takes the low limit of the range and the second column takes the upper limit. On the other hand data for the ponds (Table 2) are entered as vectors. Data is entered with the aid of matrix editor in the calculator. The dimension of the rows depends on the number of fish/pond elements used.

The naming convention for the matrices of fish types are f1, f2, f3,..., fn corresponding to fish type 1, fish type 2, fish type 3 and etc. Similarly, we have p1, p2, p3,…, pn corresponding to pond 1, pond 2, pond 3, and etc (Fig. 1a, 1b, 2a and 2b).

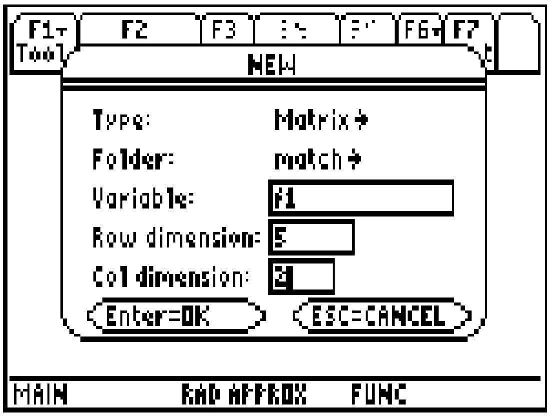

Fig. 1a: Creating matrix f1for fish type 1.

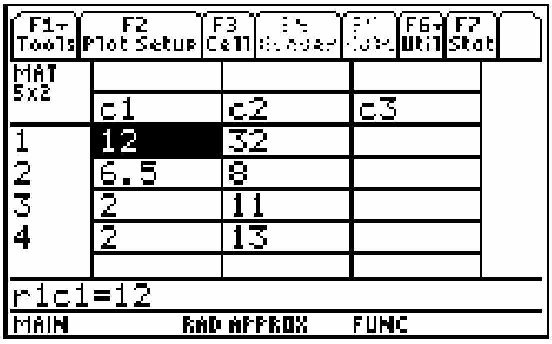

Fig. 1b: Matrix f1 values

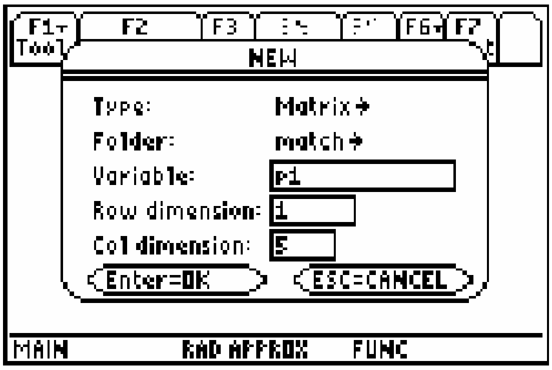

Fig. 2a: Creating vector p1 for pond 1.

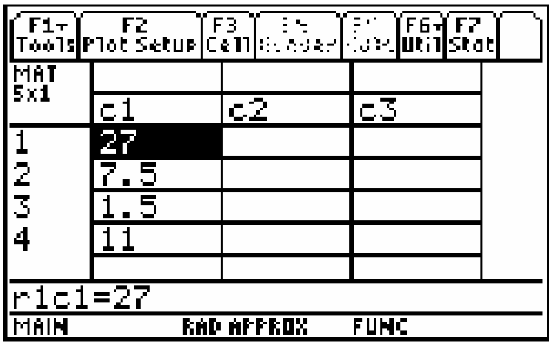

Fig. 2b: Vector p1 values.

The Weighting values (Table 4) are entered as a vector, w. The number of rows also depends on the elements in the WQVs (Fig. 3a and 3b).

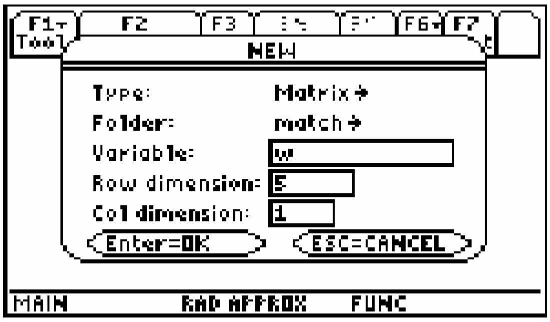

Fig. 3a: Creating a vector, w for weightings

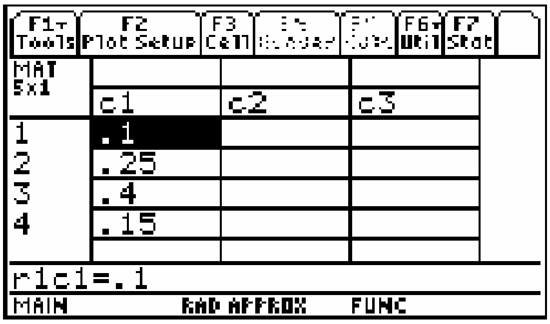

Fig. 3b: Vector, w values

Finally, the Acceptable Range of Violation values (Table 3) are entered as matrix, v, with the row and column dimension being equal to number fish elements and number of fish types respectively (Fig. 4a and 4b) .

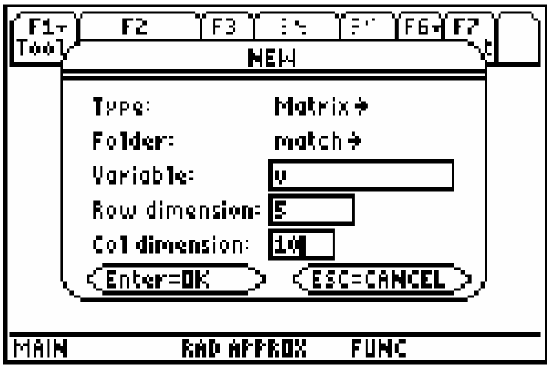

Fig. 4a: Creating a matrix, v

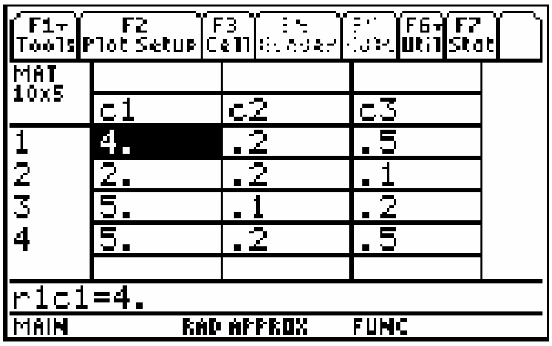

Fig. 4b: Matrix, v values

In the same vain, data entered can be edited. This is done by opening the matrix with the aid of the matrix editor and effecting the necessary changes.

Remark
All data should be stored in the directory of the program.

The Matching Process
To use the program enter its name (we called our program match) on the entry line of the Home Screen of the Calculator as shown in Fig. 5. The name of the program could be typed on the entry line or retrieved from a directory. It is advisable to run program from the directory of the program. Always remember to close the bracket or parenthesis in front of the name to avoid syntax error.

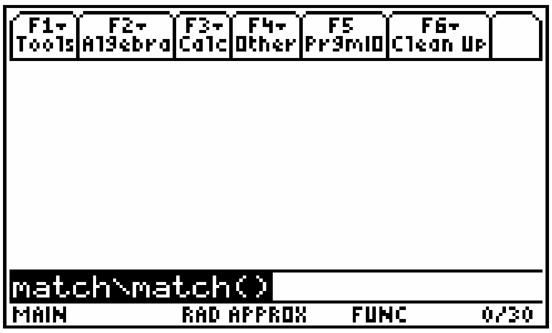

Fig. 5: Home Screen

The program is launched by pressing the ENTER key. A dialog box shown in Fig. 6 is displayed requesting information needed to do the matching.

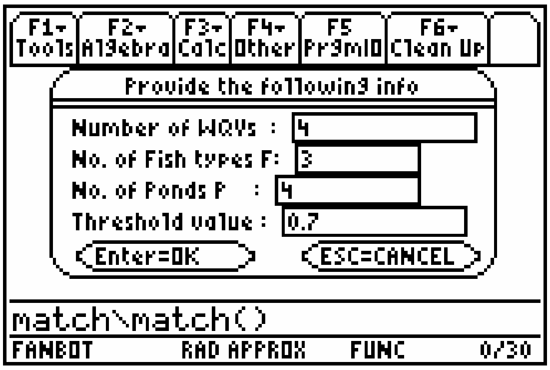

Fig. 6: Dialog box requesting inputs

The following describe the steps to be followed in Fig. 6:

1. Press the alpha key to remove the current cursor position from alphabet mode to allow entry of positive integers required.
2. Enter the number of elements in the WQVs.
3. Enter the number of fish types needed for the matching.
4. Enter the number of ponds needed for the matching.
5. Enter the threshold value. This is the value below which a match is rejected.

Finally, the enter key is press twice to continue the process.
After the dialog box, Fig. 7 is displayed.

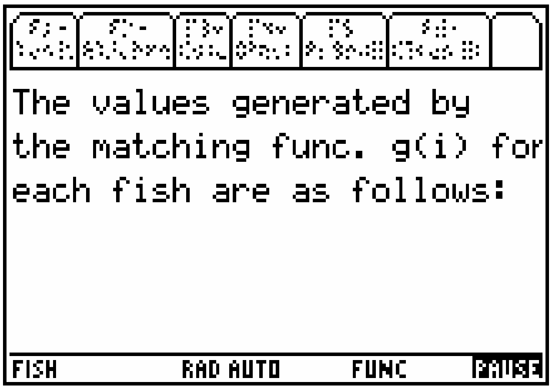

Fig. 7: Information about next screen

- To continue the process from one stage to the other, the ENTER key is pressed.

The values generated by the matching function are in a matrix form. The fish elements for a particular fish type are matched with the corresponding pond elements of the various ponds and the outcome is given by the matrix. This is done for all the fish types. Some of the matching function values are shown in Fig. 8a and 8b.

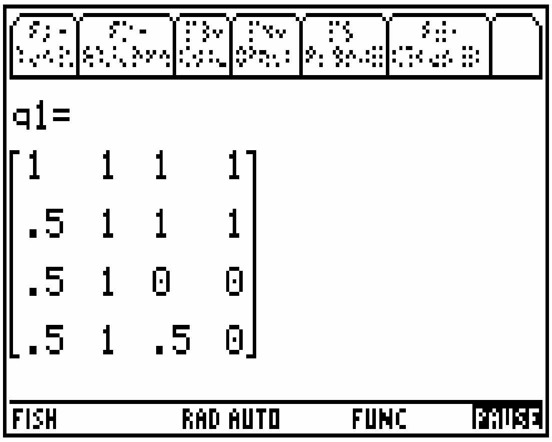

Fig. 8a: Matching function values for Fish type 1

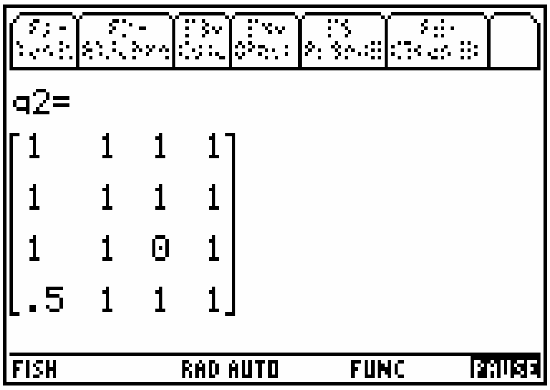

Fig. 8b: Matching function values for Fish type 2.

The cumulative effect of the matching together with the weightings is given by the matrix in Fig.9 representing the survivorship of fish types in the various ponds.

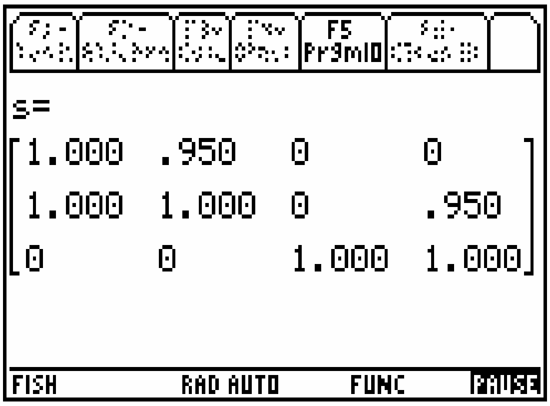

Fig. 9: Matrix from the cumulative values of the matching

Based on the threshold value entered, the inclusion matrix is formulated. The rows and columns of this matrix represent the fish types and ponds respectively. The values in the matrix show whether a fish type can go into a particular pond or not as explained in the theory. The inclusion matrix is shown in Fig. 10.

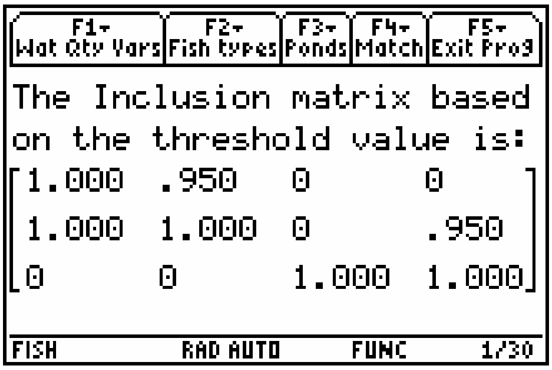

Fig. 10: Inclusion matrix

Summary of Matching

Summary of the matching has been categorised into two. Thus, total and partial matching. The total matching is the ones whose values in the inclusion matrix are ones (1), and between 0 and 1 for partial matching. The summary is shown in the Fig. 11a and 11b.

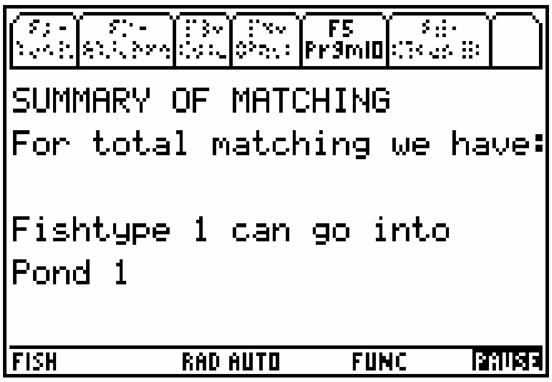

Fig. 11a: Summary of matching for total matchingl

Fig. 11b: Summary of matching for partial matching

After the summary, a message is displayed indicating the end of the matching process. This is shown in the Fig. 12.

Fig. 12: Message indicating end process

CONCLUSION

The program presented has been made very simple to use and can be carried to field to help in the stocking of fish in fish pond culture. The zeros (0) in the inclusion matrix indicate mismatching. This implies that those fish types will not survive in the respective ponds. It can be modified to solve similar matching problems.

This program would run on the TI 92 and TI 92 plus with little or no modification. However the same can not be said about the other models of TI graphing calculator due to language incompatibility. Graphing Calculation Technology has immense potential that needs to be utilised.

REFERENCES

Benjamin, N. L. (1997). Graphing Technology Guide: Houghton Mifflin Company: Boston, New York, pp 94-113.

Boateng, F. O. (2006). A Matching Technique and Its Application to Aquaculture. M.Sc. Thesis, Department of Mathematics, Kwame Nkrumah University of Science and Technology, Kumasi, pp 44-47.

Dontwi, I. K., Sahli, S. and Dontwi, J. (2005). “A Matching Scheme for Assigning Fish Types to Ponds. A Water Management Approach.” *European Journal of Scientific Reseach*, 11(1): 13-20.

Dontwi, J. (2005). Effective Monitoring and Surveillance of Fish Diseases in the Country. Report on Aquaculture. Presented to the Ministry of Food and Agriculture, Ghana.

Dunham, P. H. (2000). *Hand-held Calculators in Mathematics Education: A Research Perspective*. Hand-held Technology in Mathematics and Science Education. A Collection of Papers: The Ohio State University.

Froese, R. and Pauly, D. (2005). FishBase. World Wide Web electronic publication. www.fishbase.org, version (11/2005), (accessed 2005 December 17).

Kutzler, B. (2000). *The Algebraic Calculator as a Pedagogical Tool for Teaching Mathematics*. Hand-held Technology in Mathematics and Science Education. A Collection of Papers: The Ohio State University.

APPENDIX

Source Code of the Program

```
Prgm
ClrIO
Try
NewFold fanbot
Else
setFold(fanbot)
EndTry
Dialog
Title "Provide the following info"
Request "Number of WQVs",r
Request "No. of Fish types F",n
Request "No. Of Ponds P",m
Request "Threshold value",u
EndDlog
expr(r)→r:expr(n)→n
expr(m)→m:expr(u)→u
newMat(r,2)→f
newMat(r,2)→p
newMat(m,r)→g
newMat(n,m)→s
For i,1,n
newMat(m,r)→#("g"&string(i))
EndFor
For i,1,n
#("f"&string(i))→f
For j,1,m
#("p"&string(j))→p
For k,1,r
If f[k,1]≤p[k,1] and p[k,2]≤f[k,2] Then
1→g[j,k]
ElseIf abs(f[k,1]-p[k,1])≤v[k,1] or abs(f[k,2]-p
[k,2])≤v[k,1] Then
.5→g[j,k]
Else
0→g[j,k]
EndIf
EndFor
EndFor
g→#("g"&string(i))
EndFor
Disp "The values generated by the matching
func."
Disp " g(i) for each fish are as follows:"
```

```
Pause
For i,1,n
Disp "g"&string(i)
Disp #("g"&string(i))
Pause
ClrIO
EndFor
For i,1,n
#("g"&string(i))→g
For j,1,m
If Π(g[j,k],k,1,r)=0 Then
0→s[i,j]
Else
Σ(w[k,1]*g[j,k],k,1,r)→s[i,j]
EndIf
EndFor
EndFor
Disp "s =",s
For i,1,n
For j,1,m
If s[i,j]<u Then
0→s[i,j]
EndIf
EndFor
EndFor
s→c
Pause
ClrIO
Disp "The Inclusion Matrix based"
Disp " on the threshold value is"
Pause
Disp c
Pause
ClrIO
Disp "SUMMARY OF MATCHING"
Disp "For total matching we have:"
Pause
For i,1,n
0→t
For j,1,m
If c[i,j]=1 and t=0 Then
Disp "Fish type"&string(i)&" can go into "
Disp "Pond"&string(j)
1→t
ElseIf c[i,j]=1 and t=1 Then
Disp "Pond"&string(j)
1→t
Else
0→t
EndIf
EndFor
Pause
EndFor
Pause
Disp
Disp "For partial matching"
Disp "we have:"
For i,1,n
0→t
For j,1,m
If c[i,j]>0 and c[i,j]<1 Then
If t=0 Then
Disp "Fishtype "&string(i)&" may go into "
Disp "Pond"&string(j)
1→t
ElseIf t=1 Then
Disp "Pond"&string(j)
1→t
Else
0→t
EndIf
EndIf
EndFor
Pause
EndFor
Text "THANK YOU. THE PROCESS HAS
ENDED."
For i,1,n
DelVar #("g"&string(i))
EndFor
DelVar f,p,g,i,j,k,n,m,r,t
setFold(main)
EndPrgm
```