ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών
Πανεπιστημιούπολη Σερρών

Προγραμματισμός Ι (Θ)

Δρ. Δημήτρης Βαρσάμης
Αναπληρωτής Καθηγητής

Οκτώβριος 2019

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι (Θ)

Contents

Δομές δεδομένων - Πίνακες

- Οι πίνακες είναι μια στατική δομή δεδομένων
- Ο πίνακας δίνει τη δυνατότητα στον προγραμματιστή να δημιουργήσει μια δομή δεδομένων στην οποία μπορεί να αποθηκεύσει (καταχωρίσει) πολλές τιμές.
- Ο προγραμματιστής αναφέρεται και διαχειρίζεται τα δεδομένα τα οποία έχουν αποθηκευτεί σε πίνακα με τη βοήθεια ενός ονόματος και των δεικτών του πίνακα.
- Οι δείκτες του πίνακα σε C ξεκινούν από το μηδέν και είναι μια αύξουσα ακολουθία. Η τιμή του δείκτη μας δείχνει την θέση του πίνακα.

Δομές δεδομένων - Πίνακες

- Οι πίνακες μπορούν να έχουν πολλές διαστάσεις
  - Οι μονοδιάστατοι πίνακες οι οποίοι έχουν ένα δείκτη
  - Οι δισδιάστατοι πίνακες οι οποίοι έχουν δυο δείκτες
  - Οι πολυδιάστατοι πίνακες οι οποίοι έχουν αριθμό δεικτών αντίστοιχο με τις διαστάσεις του πίνακα.
- Η διαχείριση και επεξεργασία των στοιχείων ενός πίνακα γίνεται με την βοήθεια επαναληπτικών εντολών
- Η πιο κατάλληλη επαναληπτική εντολή για την επεξεργασία πινάκων είναι η εντολή **`for`**. (είναι προφανές ότι για την επεξεργασία πινάκων μπορούμε να χρησιμοποιήσουμε όποια εντολή επανάληψης θέλουμε)

Δομές δεδομένων - Πίνακες

- Τα στοιχεία ενός πίνακα πρέπει να είναι ιδίου τύπου
- Ένας πίνακας μπορεί να δηλωθεί σύμφωνα με τους γνωστούς τύπους δεδομένων της C, δηλαδή, **`int`**, **`float`**, **`char`** κ.α.
- Οι πιο συνηθισμένοι τύποι πινάκων που χρησιμοποιούνται είναι οι αριθμητικοί (arrays) και οι αλφαριθμητικοί ή συμβολοσειρές (strings)

Contents

Δισδιάστατοι Πίνακες

	Column 1	Column 2	Column 3	Column 4
Row 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Row 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Row 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

Δισδιάστατοι Πίνακες - Δήλωση

Δήλωση δισδιάστατου πίνακα

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(){
4     int A[100][20];
5     return 0;
6 }
```

- Η δήλωση του πίνακα γίνεται όπως των απλών μεταβλητών με την προσθήκη του μεγέθους του πινάκα το οποίο περικλείεται σε αγκύλες
- Για τον δισδιάστατο πίνακα δηλώνουμε δυο μεγέθη, το πλήθος των γραμμών και το πλήθος των στηλών

Δισδιάστατοι Πίνακες - Δήλωση

Δήλωση δισδιάστατου πίνακα

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define N 10
4 #define M 20
5 int main(){
6     int A[N][M];
7     return 0;
8 }
```

- Ένας ορθός τρόπος να δηλώνουμε τα μεγέθη του πίνακα είναι με την οδηγία #define με την οποία ορίζουμε μια σταθερά

Δισδιάστατοι Πίνακες - Αρχικοποίηση Ι

Αρχικοποίηση και εμφάνιση στοιχείων μονοδιάστατου πίνακα

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(){
4     int A[2][3]={{1,2,3},{4,5,6}};
5     int B[][3]={{1,2,3},{4,5,6}};
6     int B[2][3]={1,2,3,4,5,6};
7     return 0;
8 }
```

- Στην πρώτη περίπτωση δηλώνουμε το πλήθος των γραμμών (2) και το πλήθος των στηλών (3) και αρχικοποιούμε τον πίνακα με τιμές οι οποίες είναι χωρισμένες σε τριάδες (πλήθος στοιχείων γραμμής)

Δισδιάστατοι Πίνακες - Αρχικοποίηση II

- Στην δεύτερη περίπτωση δηλώνουμε το πλήθος των στηλών (3) και αρχικοποιούμε τον πίνακα με τιμές οι οποίες είναι χωρισμένες σε τριάδες (πλήθος στοιχείων γραμμής). Το πλήθος των γραμμών θα το υπολογίσει ο μεταγλωττιστής.
- Στην τρίτη περίπτωση δηλώνουμε το πλήθος των γραμμών (2) και το πλήθος των στηλών (3) και αρχικοποιούμε τον πίνακα με τιμές σε πλήθος όσες και οι συνολικές θέσεις του δισδιάστατου πίνακα. Τα στοιχεία θα τα τοποθετήσει στις θέσεις του πίνακα ο μεταγλωττιστής.

Δισδιάστατοι Πίνακες - Εμφάνιση (α) Ι

Εμφάνιση στοιχείων δισδιάστατου πίνακα

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4      int i, j, A[2][3]={{10,15,5},{1,2,3}};
5      for(i=0;i<2;i=i+1){
6          for(j=0;j<3;j=j+1){
7              printf("\t%d",A[i][j]);
8          }
9      }
10     printf("\n");
11     return 0;
12 }
```

Δισδιάστατοι Πίνακες - Εμφάνιση (α) ΙΙ

```
10 15 5 1 2 3
```

- Εμφάνιση των στοιχείων του πίνακα σε μια γραμμή (απλή προγραμματιστική υλοποίηση)
- Η επεξεργασία δισδιάστατου πίνακα απαιτεί την χρήση εμφωλευμένων εντολών επανάληψης.

Δισδιάστατοι Πίνακες - Εμφάνιση (β) Ι

Εμφάνιση στοιχείων δισδιάστατου πίνακα

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4      int i, j, A[2][3]={{10,15,5},{1,2,3}};
5      for(i=0;i<2;i=i+1){
6          for(j=0;j<3;j=j+1){
7              printf("\t%d\n",A[i][j]);
8          }
9      }
10     printf("\n");
11     return 0;
12 }
```

Δισδιάστατοι Πίνακες - Εμφάνιση (β) II

```
10
15
5
1
2
3
```

- Εμφάνιση των στοιχείων του πίνακα σε μια στήλη (απλή προγραμματιστική υλοποίηση)

Δισδιάστατοι Πίνακες - Εμφάνιση (γ) Ι

Εμφάνιση στοιχείων δισδιάστατου πίνακα

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4      int i, j, A[2][3]={{10,15,5},{1,2,3}};
5      for(i=0;i<2;i=i+1){
6          for(j=0;j<3;j=j+1){
7              printf("A[%d][%d]=%d\n",i,j,A[i][j]);
8          }
9      }
10     printf("\n");
11     return 0;
12 }
```

Δισδιάστατοι Πίνακες - Εμφάνιση (γ) II

```
A[0][0]=10
A[0][1]=15
A[0][2]=5
A[1][0]=1
A[1][1]=2
A[1][2]=3
```

- Εμφάνιση των στοιχείων και των θέσεων του πίνακα σε μια στήλη (ορθή προγραμματιστική υλοποίηση)

Δισδιάστατοι Πίνακες - Εμφάνιση (δ) Ι

Εμφάνιση στοιχείων δισδιάστατου πίνακα

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4      int i, j, A[2][3]={{10,15,5},{1,2,3}};
5      for(i=0;i<2;i=i+1){
6          for(j=0;j<3;j=j+1){
7              printf("\t%d",A[i][j]);
8          }
9          printf("\n");
10     }
11     printf("\n");
12     return 0;
13 }
```

Δισδιάστατοι Πίνακες - Εμφάνιση (δ) II

```
10      15      5
1       2       3
```

- Εμφάνιση των στοιχείων του πίνακα σε μαθηματική μορφή

Δισδιάστατοι Πίνακες - Εκχώρηση - Επεξεργασία Ι

Εκχώρηση τιμών και επεξεργασία στοιχείων δισδιάστατου πίνακα

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4      int i, j, A[2][3], sumR, maxA=0;
5      for(i=0;i<2;i=i+1){
6          for(j=0;j<3;j=j+1){
7              printf("A[%d][%d]=",i,j);
8              scanf("%d",&A[i][j]);
9          }
10     }
11     for(i=0;i<2;i=i+1){
12         for(j=0;j<3;j=j+1){
```

Δισδιάστατοι Πίνακες - Εκχώρηση - Επεξεργασία II

```
13                if(A[i][j]>maxA||(i==0&&j==0)){
14                    maxA=A[i][j];
15                }
16            }
17        }
18        printf("The maximum element of A is %d\n",
              maxA);
19        for(i=0;i<2;i=i+1){
20            sumR=0;
21            for(j=0;j<3;j=j+1){
22                sumR=sumR+A[i][j];
23            }
24            printf("The sum of row %d is %d\n", i,
                  sumR);
```

Δισδιάστατοι Πίνακες - Εκχώρηση - Επεξεργασία III

```
25        }
26        return 0;
27    }
```

Δισδιάστατοι Πίνακες - Εκχώρηση - Επεξεργασία IV

```
A[0][0]=1
A[0][1]=2
A[0][2]=3
A[1][0]=4
A[1][1]=5
A[1][2]=6
The maximum element of A is 6
The sum of row 0 is 6
The sum of row 1 is 15
```

Contents

Δισδιάστατοι Πίνακες - Άσκηση 1

Να γράψετε ένα πρόγραμμα το οποίο να δημιουργεί και να εμφανίζει τον πίνακα A με διαστάσεις 10×10.

- Τα στοιχεία του πίνακα A δίνονται από τον τύπο

$$A[i][j] = \frac{j+1}{i+1}$$

Δισδιάστατοι Πίνακες - Άσκηση 2

Να γράψετε ένα πρόγραμμα το οποίο

- να δημιουργεί (με αυτοματοποιημένη διαδικασία) και να εμφανίζει τον διαγώνιο πίνακα

$$D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

- να δημιουργεί (με αυτοματοποιημένη διαδικασία) και να εμφανίζει τον πίνακα

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}$$

- να υπολογίζει και να εμφανίζει την παράσταση

$$2 \cdot D + 3 \cdot B$$

Δισδιάστατοι Πίνακες - Άσκηση 3

Να γράψετε ένα πρόγραμμα το οποίο να δημιουργεί και να εμφανίζει τον πίνακα A με διαστάσεις 4×3.

- Τα στοιχεία του πίνακα A δίνονται από τον τύπο

$$A[i][j] = \frac{2j+1}{2i+1}$$

- Να εμφανίζει τον ανάστροφο του πίνακα A (οι γραμμές στήλες και οι στήλες γραμμές)
- Να εμφανίζει τον πίνακα A κανονικοποιημένο (θα πρέπει να διαιρέσουμε όλα τα στοιχεία του πίνακα με το μεγαλύτερο στοιχείο του πίνακα
- Να εμφανίζει τα αθροίσματα των γραμμών και το μέγιστο αυτών των αθροισμάτων
- Να εμφανίζει τα μέγιστα των στηλών και το άθροισμα αυτών των μεγίστων

Δισδιάστατοι Πίνακες - Άσκηση 4

Να γράψετε ένα πρόγραμμα το οποίο να δέχεται τον πίνακα A με διαστάσεις 3×3 και να ελέγχει αν

- ο πίνακας A είναι τριγωνικός άνω (τα στοιχεία κάτω από την διαγώνιο να είναι μηδέν)
- ο πίνακας A είναι τριγωνικός κάτω (τα στοιχεία πάνω από την διαγώνιο να είναι μηδέν)
- ο πίνακας A είναι διαγώνιος (τα στοιχεία κάτω και πάνω από την διαγώνιο να είναι μηδέν)

Contents

Προτεινόμενη λύση της Άσκησης 1 1

```
1  int main(){
2      int i,j;
3      float A[10][10];
4      for (i=0;i<10;i=i+1){
5          for (j=0;j<10;j=j+1){
6              A[i][j]=(float)(j+1)/(i+1);
7          }
8      }
9      for (i=0;i<10;i=i+1){
10         for (j=0;j<10;j=j+1){
11             printf("\t%.2f",A[i][j]);
12         }
13         printf("\n");
14     }
15     return 0;
```

Προτεινόμενη λύση της Άσκησης 1 II

```
16 || }
```

Προτεινόμενη λύση της Άσκησης 2 Ι

```
 1 int main(){
 2     int A[3][3],B[3][3],C[3][3],i,j;
 3     for (i=0;i<3;i=i+1){
 4         for (j=0;j<3;j=j+1){
 5             if(i==j){
 6                 A[i][j]=3;
 7             }
 8             else{
 9                 A[i][j]=0;
10             }
11         }
12     }
13     for (i=0;i<3;i=i+1){
14         for (j=0;j<3;j=j+1){
15             B[i][j]=i+1;
```

Προτεινόμενη λύση της Άσκησης 2 II

```
16          }
17      }
18      for (i=0;i<3;i=i+1){
19          for (j=0;j<3;j=j+1){
20              C[i][j]=2*A[i][j]+3*B[i][j];
21          }
22      }
23      printf("\n\nA=\n");
24      for (i=0;i<3;i=i+1){
25          for (j=0;j<3;j=j+1){
26              printf("\t%d",A[i][j]);
27          }
28          printf("\n");
29      }
30      printf("\n\nB=\n");
```

Προτεινόμενη λύση της Άσκησης 2 III

```
31      for (i=0;i<3;i=i+1){
32          for (j=0;j<3;j=j+1){
33              printf("\t%d",B[i][j]);
34          }
35          printf("\n");
36      }
37      printf("\n\n2*A+3*B=\n");
38      for (i=0;i<3;i=i+1){
39          for (j=0;j<3;j=j+1){
40              printf("\t%d",C[i][j]);
41          }
42          printf("\n");
43      }
44      return 0;
45  }
```

Προτεινόμενη λύση της Άσκησης 3 Ι

```
1  int main(){
2      int i,j;
3      float A[4][3], max=0, sumR, maxS=0, maxC=0,
           sumMaxC=0;
4      for (i=0;i<4;i=i+1){
5          for (j=0;j<3;j=j+1){
6              A[i][j]=(float)(2*j+1)/(2*i+1);
7          }
8      }
9      printf("\n\nA=\n");
10     for (i=0;i<4;i=i+1){
11         for (j=0;j<3;j=j+1){
12             printf("\t%.3f",A[i][j]);
13         }
14         printf("\n");
```

Προτεινόμενη λύση της Άσκησης 3 II

```
15          }
16          printf("\n\nTranspose Matrix=\n");
17          for (j=0;j<3;j=j+1){
18              for (i=0;i<4;i=i+1){
19                  printf("\t%.3f",A[i][j]);
20              }
21              printf("\n");
22          }
23          for (i=0;i<4;i=i+1){
24              for (j=0;j<3;j=j+1){
25                  if(A[i][j]>max||(i==0&&j==0)){
26                      max=A[i][j];
27                  }
28              }
29          }
```

Προτεινόμενη λύση της Άσκησης 3 III

```
30        printf("\n\nNormalize Matrix=\n");
31        for (i=0;i<4;i=i+1){
32            for (j=0;j<3;j=j+1){
33                printf("\t%.3f",A[i][j]/max);
34            }
35            printf("\n");
36        }
37        printf("\n\nMAX SUM ROW\n");
38        for (i=0;i<4;i=i+1){
39            sumR=0;
40            for (j=0;j<3;j=j+1){
41                sumR=sumR+A[i][j];
42            }
43            if(sumR>maxS||i==0){
44                maxS=sumR;
```

Προτεινόμενη λύση της Άσκησης 3 IV

```
45          }
46          printf("Sum of Row %d is %f\n",i,sumR);
47      }
48      printf("\nMAX SUM ROW= %f\n",maxS);
49      printf("\n\nSUM MAX COLUMN\n");
50      for (i=0;i<4;i=i+1){
51          for (j=0;j<3;j=j+1){
52              if(A[i][j]>maxC||j==0){
53                  maxC=A[i][j];
54              }
55          }
56          sumMaxC=sumMaxC+maxC;
57          printf("Max of Column %d is %f\n",j,maxC)
                ;
58      }
```

Προτεινόμενη λύση της Άσκησης 3 V

```
59        printf("\nSUM MAX COLUMN= %f\n",sumMaxC);
60        return 0;
61    }
```

Προτεινόμενη λύση της Άσκησης 4 Ι

```
1  int main(){
2      int A[3][3],i,j,count1=0,count2=0;
3      for(i=0;i<3;i=i+1){
4          for(j=0;j<3;j=j+1){
5              printf("A[%d][%d]=",i,j);
6              scanf("%d",&A[i][j]);
7          }
8      }
9      for (i=0;i<3;i=i+1){
10         for (j=0;j<3;j=j+1){
11             printf("\t%d",A[i][j]);
12         }
13         printf("\n");
14     }
15     for (i=0;i<3;i=i+1){
```

Προτεινόμενη λύση της Άσκησης 4 II

```
16              for (j=0;j<3;j=j+1){
17                  if(i>j&&A[i][j]==0){
18                      count1=count1+1;
19                  }
20                  if(i<j&&A[i][j]==0){
21                      count2=count2+1;
22                  }
23              }
24          }
25          if(count1==3 && count2==3){
26              printf("The matrix is Diagonal\n");
27          }
28          else if(count1==3){
29              printf("The matrix is Upper Triangular\n"
                    );
```

Προτεινόμενη λύση της Άσκησης 4 III

```
30          }
31          else if(count2==3){
32              printf("The matrix is Low Triangular\n");
33          }
34          else{
35              printf("The matrix is Random\n");
36          }
37          return 0;
38  }
```