

Αλγόριθμοι σε C

Μάθημα 4:

Αλγεβρικές Πράξεις Πινάκων

Δημήτρης Ψούνης



www.psounis.gr



Περιεχόμενα Μαθήματος

A. Θεωρία

1. Μονοδιάστατοι Πίνακες

1. Αντιγραφή Πινάκων
2. Πρόσθεση Πινάκων
3. Εσωτερικό Γινόμενο Πινάκων

2. Διδιάστατοι Πίνακες

1. Διαπέραση Πίνακα
2. Πράξεις πίνακα κατά γραμμές (ή στήλες)
3. Πρόσθεση Πινάκων
4. Γινόμενο Πινάκων
5. Ανάστροφος Πίνακας

B. Ασκήσεις



A. Θεωρία

1. Μονοδιάστατοι Πίνακες

1. Αντιγραφή Πινάκων

- Το ακόλουθο τμήμα κώδικα αντιγράφει το περιεχόμενο του πίνακα B στον A.

```
for (i=0; i<N; i++)  
{  
    A[i]=B[i];  
}
```

Άσκηση 1: Αφού μελετήσετε το πρόγραμμα 1d-copy.c , κατασκευάστε συνάρτηση με όνομα copy_1d_table(int *a, int *b, int n) που αντιγράφει το περιεχόμενο του πίνακα b στον πίνακα a. Μετατρέψτε το πρόγραμμα ώστε να χρησιμοποιεί τη συνάρτηση που ορίσατε.



A. Θεωρία

1. Μονοδιάστατοι Πίνακες

2. Άθροισμα Πινάκων

- Το ακόλουθο τμήμα κώδικα προσθέτει το περιεχόμενο δύο πινάκων A και B και αποθηκεύει το αποτέλεσμα στον πίνακα C.

```
for (i=0; i<N; i++)  
{  
    C[i]=A[i]+B[i];  
}
```

- που αντιστοιχεί στην πρόσθεση διανυσμάτων (π.χ. για N=3):

$$\begin{bmatrix} 5 \\ 8 \\ 1 \end{bmatrix} + \begin{bmatrix} 2 \\ 11 \\ 3 \end{bmatrix} = \begin{bmatrix} 7 \\ 19 \\ 4 \end{bmatrix}$$

Άσκηση 2: Αφού μελετήσετε το πρόγραμμα 1d-sum.c , κατασκευάστε συνάρτηση με όνομα `sum_1d_table(int *c, int *a, int *b, int n)` που εκτελεί την πράξη πινάκων: `c=a+b`. Μετατρέψτε το πρόγραμμα ώστε να χρησιμοποιεί τη συνάρτηση που ορίσατε.



A. Θεωρία

1. Μονοδιάστατοι Πίνακες

3. Εσωτερικό Γινόμενο Πινάκων

- Το ακόλουθο τμήμα κώδικα πραγματοποιεί το εσωτερικό γινόμενο των πινάκων A και B και αποθηκεύει το αποτέλεσμα στη μεταβλητή c.

```
c=0;
for (i=0; i<N; i++)
{
    c=c+A[i]*B[i];
}
```

- που αντιστοιχεί στην πρόσθεση διανυσμάτων (π.χ. για N=3):

Άσκηση 3: Αφού μελετήσετε το πρόγραμμα 1d-product.c , κατασκευάστε συνάρτηση με όνομα `product_1d_table(int *a, int *b, int n)` που υπολογίζει και επιστρέφει το εσωτερικό γινόμενο των διανυσμάτων a και b. Μετατρέψτε το πρόγραμμα ώστε να χρησιμοποιεί τη συνάρτηση που ορίσατε.



A. Θεωρία

2. Διδιάστατοι Πίνακες

1. Διαπέραση Πίνακα

- Το ακόλουθο τμήμα κώδικα σαρώνει τον πίνακα κατά γραμμές και τυπώνει τα στοιχεία του.

```
for (i=0; i<M; i++)  
{  
    for (j=0; j<N; j++)  
    {  
        printf("%2d ", pinakas[i][j]);  
    }  
    printf("\n");  
}
```

Άσκηση 4: Αφού μελετήσετε το πρόγραμμα 2d_traversal_rows.c, τροποποιήστε το ώστε να τροποποιεί τον πίνακα και να τυπώνει τα τετράγωνα των στοιχείων του πίνακα.

A. Θεωρία

2. Διδιάστατοι Πίνακες

1. Διαπέραση Πίνακα

- Το ακόλουθο τμήμα κώδικα σαρώνει τον πίνακα κατά στήλες και τυπώνει τα στοιχεία του.

```
for (j=0; j<N; j++)  
{  
    for (i=0; i<M; i++)  
    {  
        printf("%2d ", pinakas[i][j]);  
    }  
    printf("\n");  
}
```

Άσκηση 5: Αφού μελετήσετε το πρόγραμμα 2d_traversal_cols.c, τροποποιήστε το ώστε να τροποποιεί τον πίνακα και να τυπώνει τους κύβους των στοιχείων του πίνακα.

Άσκηση 6: Κατασκευάστε ένα πρόγραμμα ώστε να γίνεται η διαπέραση του πίνακα κατά γραμμές, αλλά να τυπώνονται αντίστροφα τα στοιχεία κάθε γραμμής

Άσκηση 7: Κατασκευάστε ένα πρόγραμμα ώστε να γίνεται η διαπέραση του πίνακα κατά στήλες, αλλά να τυπώνονται αντίστροφα τα στοιχεία κάθε γραμμής



A. Θεωρία

2. Διδιάστατοι Πίνακες

2. Πράξεις πίνακα κατά γραμμές (ή στήλες)

- Το ακόλουθο τμήμα κώδικα υπολογίζει και τυπώνει το άθροισμα κάθε γραμμής

```
for (i=0; i<M; i++)  
{  
    sum=0;  
    for (j=0; j<N; j++)  
    {  
        sum+=pinakas[i][j];  
    }  
    printf("%2d ", sum);  
}
```

Άσκηση 8: Αφού μελετήσετε το πρόγραμμα sum_rows.c, τροποποιήστε το ώστε να αποθηκεύει τα επιμέρους αθροίσματα σε έναν νέο πίνακα sums και να τυπώνει στο τέλος τα αθροίσματα.

Άσκηση 9: Κατασκευάστε πρόγραμμα το οποίο να εμφανίζει το άθροισμα των στοιχείων κάθε στήλης του διδιάστατου πίνακα.



A. Θεωρία

2. Διδιάστατοι Πίνακες

2. Πράξεις πίνακα κατά γραμμές (ή στήλες)

Άσκηση 10: Κατασκευάστε πρόγραμμα που να μορφοποιεί την εκτύπωση των αθροισμάτων γραμμών-στηλών σύμφωνα με το παράδειγμα:

$$\begin{array}{ccc|c} 4 & 2 & 1 & 7 \\ 4 & 1 & 4 & 9 \\ 1 & 5 & 2 & 8 \\ \hline 9 & 8 & 7 & \end{array}$$

Άσκηση 11: Κατασκευάστε πρόγραμμα το οποίο να εμφανίζει τους μέσους όρους κάθε γραμμής και κάθε στήλης του πίνακα



A. Θεωρία

2. Διδιάστατοι Πίνακες

3. Πρόσθεση Πινάκων

- Το ακόλουθο τμήμα κώδικα υπολογίζει και τυπώνει το άθροισμα δύο πινάκων:

```
for (i=0; i<M; i++)  
{  
    for (j=0; j<N; j++)  
    {  
        C[i][j]=A[i][j]+B[i][j];  
    }  
}
```

Άσκηση 12: Κατασκευάστε πρόγραμμα το οποίο να αρχικοποιεί τυχαία δύο $M \times N$ πίνακες και έπειτα να υπολογίζει και να τυπώνει το άθροισμά τους.



A. Θεωρία

2. Διδιάστατοι Πίνακες

4. Γινόμενο Πινάκων

- Το ακόλουθο τμήμα κώδικα υπολογίζει και τυπώνει το γινόμενο δύο πινάκων:

```
for (i=0; i<M; i++)
    for (j=0; j<N; j++)
    {
        C[i][j]=0;
        for (k=0; k<K; k++)
            C[i][j]+=A[i][k]*B[k][j];
    }
```

- Παράδειγμα:

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} =$$
$$= \begin{bmatrix} 1 \cdot 1 + 2 \cdot 0 + 3 \cdot 1 & 1 \cdot 2 + 2 \cdot 1 + 3 \cdot 0 & 1 \cdot 0 + 2 \cdot 1 + 3 \cdot 1 & 1 \cdot 1 + 2 \cdot 0 + 3 \cdot 1 \\ 3 \cdot 1 + 2 \cdot 0 + 2 \cdot 1 & 3 \cdot 2 + 2 \cdot 1 + 2 \cdot 0 & 3 \cdot 0 + 2 \cdot 1 + 2 \cdot 1 & 3 \cdot 1 + 2 \cdot 0 + 2 \cdot 1 \end{bmatrix}$$
$$= \begin{bmatrix} 4 & 4 & 5 & 4 \\ 5 & 8 & 4 & 5 \end{bmatrix}$$

Άσκηση 13: Κατασκευάστε πρόγραμμα το οποίο να αρχικοποιεί τυχαία δύο $M \times N$ πίνακες και έπειτα να υπολογίζει και να τυπώνει το γινόμενο τους.



A. Θεωρία

2. Διδιάστατοι Πίνακες

5. Ανάστροφος πίνακας

- Το ακόλουθο τμήμα κώδικα δημιουργεί τον ανάστροφο ενός πίνακα A:

```
for (i=0; i<M; i++)  
    for (j=0; j<N; j++)  
        B[j][i]=A[i][j];
```

- Παράδειγμα:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 2 \end{bmatrix}, \quad B = A^T = \begin{bmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 2 \end{bmatrix}$$

Άσκηση 14: Κατασκευάστε πρόγραμμα το οποίο να αρχικοποιεί τυχαία ένα $M \times N$ πίνακα και έπειτα να υπολογίζει και να τυπώνει τον ανάστροφό του.



B. Ασκήσεις

Εφαρμογή 1

- Τροποποιήστε το πρόγραμμα της άσκησης 12 (άθροισμα διδιάστατων πινάκων) ώστε να διαχειρίζεται τη μνήμη δυναμικά. Θα χρειαστεί να ορίσετε:
 - Μία συνάρτηση `create_array` που να δεσμεύει δυναμικά το χώρο για το διδιάστατο πίνακα
 - Μία συνάρτηση `init_array` η οποία να αρχικοποιεί με τυχαίο τρόπο τα στοιχεία του πίνακα
 - Μία συνάρτηση `sum_arrays` η οποία να αθροίζει δύο διδιάστατους πίνακες
 - Μία συνάρτηση `print_array` η οποία να τυπώνει ένα διδιάστατο πίνακα
 - Μία συνάρτηση `destroy_array` η οποία να αποδεσμεύει τη μνήμη που έχει δεσμεύσει ο διδιάστατος πίνακας.