

ΕΡΓΑΣΤΗΡΙΟ 3 - ΣΗΜΕΙΩΣΕΙΣ

Συγχώνευση Πινάκων

Η συγχώνευση μπορεί να υλοποιηθεί μέσω της αντιγραφής των στοιχείων δύο πινάκων σε ένα τρίτο πίνακα ο οποίος χωρά τους δύο πρώτους. Συνεπώς, ο τρίτος πίνακας θα περιλαμβάνει τα στοιχεία του πρώτου ακολουθούμενα από τα στοιχεία του δεύτερου.

Code 1

(συγχώνευση μη ταξινομημένων πινάκων)

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int arr1[10], arr2[10], arr3[20];
    int i, n1, n2, m, index=0;
    clrscr();
    printf("\n Enter the number of elements in array1 : ");
    scanf("%d", &n1);
    printf("\n\n Enter the elements of the first array");
    for(i=0;i<n1;i++)
    {
        printf("\n arr1[%d] = ", i);
        scanf("%d", &arr1[i]);
    }
    printf("\n Enter the number of elements in array2 : ");
    scanf("%d", &n2);
    printf("\n\n Enter the elements of the second array");
    for(i=0;i<n2;i++)
    {
        printf("\n arr2[%d] = ", i);
        scanf("%d", &arr2[i]);
    }
    m = n1+n2;
    for(i=0;i<n1;i++)
    {
        arr3[index] = arr1[i];
        index++;
    }
    for(i=0;i<n2;i++)
    {
        arr3[index] = arr2[i];
        index++;
    }
    printf("\n\n The merged array is");
    for(i=0;i<m;i++)
```

```

        printf("\n arr[%d] = %d", i, arr3[i]);
    getch();
    return 0;
}

```

Όταν οι πίνακες προς συγχώνευση είναι ταξινομημένοι τότε και ο τρίτος πίνακας πρέπει να είναι και αυτός. Οι λύσεις είναι δύο: είτε συγχωνεύουμε όπως έχουμε πει πιο πάνω και έπειτα ταξινομούμε ή κατά τη συγχώνευση διατηρούμε ταξινομημένα του τρίτου πίνακα. Στον ακόλουθο κώδικα υλοποιούμε τη δεύτερη λύση.

Code 2

(συγχώνευση ταξινομημένων πινάκων)

```

#include <stdio.h>
#include <conio.h>
int main()
{
    int arr1[10], arr2[10], arr3[20];
    int i, n1, n2, m, index=0;
    int index_first = 0, index_second = 0;
    clrscr();
    printf("\n Enter the number of elements in array1 : ");
    scanf("%d", &n1);
    printf("\n\n Enter the elements of the first array");
    for(i=0;i<n1;i++)
    {
        printf("\n arr1[%d] = ", i);
        scanf("%d", &arr1[i]);
    }
    printf("\n Enter the number of elements in array2 : ");
    scanf("%d", &n2);
    printf("\n\n Enter the elements of the second array");
    for(i=0;i<n2;i++)
    {
        printf("\n arr2[%d] = ", i);
        scanf("%d", &arr2[i]);
    }
    m = n1+n2;
    while(index_first < n1 && index_second < n2)
    {
        if(arr1[index_first]<arr2[index_second])
        {
            arr3[index] = arr1[index_first];
            index_first++;
        }
        else
        {
            arr3[index] = arr2[index_second];
            index_second++;
        }
        index++;
    }
    // if elements of the first array are over and the second array has some elements

```

```

if(index_first == n1)
{
    while(index_second < n2)
    {
        arr3[index] = arr2[index_second];
        index_second++;
        index++;
    }
}
// if elements of the second array are over and the first array has some elements
else if(index_second == n2)
{
    while(index_first < n1)
    {
        arr3[index] = arr1[index_first];
        index_first++;
        index++;
    }
}
printf("\n\n The merged array is");
for(i=0; i<m; i++)
    printf("\n arr[%d] = %d", i, arr3[i]);
getch();
return 0;
}

```

Ταξινόμηση Πινάκων

Ταξινόμηση είναι η αναδιάταξη των στοιχείων ενός πίνακα έτσι ώστε τα στοιχεία να τοποθετηθούν σε αύξουσα/φθίνουσα σειρά. Στη βιβλιογραφία έχουν προταθεί αρκετοί αλγόριθμοι ταξινόμησης. Ακολουθεί η κωδικοποίηση δύο από αυτούς.

Code 3

(ταξινόμηση πίνακα με τον αλγόριθμο selection sort)

```

#include <stdio.h>
int main()
{
    int array[100], n, c, d, position, swap;
    printf("Enter number of elements\n");
    scanf("%d", &n);
    printf("Enter %d integers\n", n);
    for ( c = 0 ; c < n ; c++ )
        scanf("%d", &array[c]);
    for ( c = 0 ; c < ( n - 1 ) ; c++ )
    {
        position = c;
        for ( d = c + 1 ; d < n ; d++ )
        {
            if ( array[position] > array[d] )
                position = d;
        }
        if ( position != c )

```

```

    {
        swap = array[c];
        array[c] = array[position];
        array[position] = swap;
    }
}
printf("Sorted list in ascending order:\n");
for ( c = 0 ; c < n ; c++ )
    printf("%d\n", array[c]);

return 0;
}

```

Code 4

(ταξινόμηση πίνακα με τον αλγόριθμο insertion sort)

```

#include <stdio.h>

int main()
{
    int n, array[1000], c, d, t;
    printf("Enter number of elements\n");
    scanf("%d", &n);
    printf("Enter %d integers\n", n);
    for (c = 0; c < n; c++) {
        scanf("%d", &array[c]);
    }
    for (c = 1 ; c <= n - 1; c++) {
        d = c;
        while ( d > 0 && array[d] < array[d-1]) {
            t = array[d];
            array[d] = array[d-1];
            array[d-1] = t;
            d--;
        }
    }
    printf("Sorted list in ascending order:\n");
    for (c = 0; c <= n - 1; c++) {
        printf("%d\n", array[c]);
    }
    return 0;
}

```

Αναζήτηση Στοιχείων

Η αναζήτηση περιλαμβάνει την εύρεση ή μη ενός στοιχείου σε ένα πίνακα. Ακολουθεί η κωδικοποίηση της σειριακής και της δυαδικής αναζήτησης.

Code 5

(σειριακή αναζήτηση – sequential search)

```

#include <stdio.h>
int main()
{

```

```

int array[100], search, c, n;

printf("Enter the number of elements in array\n");
scanf("%d",&n);

printf("Enter %d integer(s)\n", n);

for (c = 0; c < n; c++)
    scanf("%d", &array[c]);

printf("Enter the number to search\n");
scanf("%d", &search);

for (c = 0; c < n; c++)
{
    if (array[c] == search) /* if required element found */
    {
        printf("%d is present at location %d.\n", search, c+1);
        break;
    }
}
if (c == n)
    printf("%d is not present in array.\n", search);

return 0;
}

```

Code 6

(δυναδική αναζήτηση - binary search, εφαρμόζεται σε ταξινομημένο πίνακα)

```

#include <stdio.h>

int main()
{
    int c, first, last, middle, n, search, array[100];

    printf("Enter number of elements\n");
    scanf("%d",&n);

    printf("Enter %d integers\n", n);

    for (c = 0; c < n; c++)
        scanf("%d",&array[c]);

    printf("Enter value to find\n");
    scanf("%d", &search);

    first = 0;
    last = n - 1;
    middle = (first+last)/2;

    while (first <= last) {
        if (array[middle] < search)

```

```

    first = middle + 1;
else if (array[middle] == search) {
    printf("%d found at location %d.\n", search, middle+1);
    break;
}
else
    last = middle - 1;

middle = (first + last)/2;
}
if (first > last)
    printf("Not found! %d is not present in the list.\n", search);

return 0;
}

```

Πίνακες Πολλαπλών Διαστάσεων

Η δήλωση των διδιάστατων πινάκων γίνεται με παραπλήσιο τρόπο όπως και με τους μονοδιάστατους. Η γενική σύνταξη έχει ως εξής:

data_type array_name[row_size][column_size];

Ακολουθεί ένα παράδειγμα ενός διδιάστατου πίνακα.

Rows Columns	Col 0	Col 1	Col 2	Col 3	Col 4
Row 0	marks[0][0]	marks[0][1]	marks[0][2]	marks[0][3]	marks[0][4]
Row 1	marks[1][0]	marks[1][1]	marks[1][2]	marks[1][3]	marks[1][4]
Row 2	marks[2][0]	marks[2][1]	marks[2][2]	marks[2][3]	marks[2][4]

Ουσιαστικά, ένας διδιάστατος πίνακας είναι μια συλλογή πολλαπλών μονοδιάστατων πινάκων.

marks[0] -	marks[0]	marks[1]	marks[2]	marks[3]	marks[4]
marks[1] -	marks[0]	marks[1]	marks[2]	marks[3]	marks[4]
marks[2] -	marks[0]	marks[1]	marks[2]	marks[3]	marks[4]

Οι διευθύνσεις μνήμης ενός διδιάστατου πίνακα καθορίζονται με τους ακόλουθους τύπους:
 $Address(A[I][J]) = Base_Address + w\{M (J - 1) + (I - 1)\}$ (αν έχει αποθηκευτεί ως προς τις στήλες)
 $Address(A[I][J]) = Base_Address + w\{N (I - 1) + (J - 1)\}$ (αν έχει αποθηκευτεί ως προς τις γραμμές)

Το w είναι το μέγεθος σε bytes που καταλαμβάνει το κάθε στοιχείο.

Code 7

(δημιουργεί το τρίγωνο του Pascal - https://en.wikipedia.org/wiki/Pascal%27s_triangle)

```

#include <stdio.h>
#include <conio.h>
int main()
{

```

```

int arr[7][7]={0};
int row=2, col, i, j;
arr[0][0] = arr[1][0] = arr[1][1] = 1;
while(row <= 7)
{
    arr[row][0] = 1;
    for(col = 1; col <= row; col++)
        arr[row][col] = arr[row-1][col-1] + arr[row-1][col];
    row++;
}
for(i=0; i<7; i++)
{
    printf("\n");
    for(j=0; j<=i; j++)
        printf("\t %d", arr[i][j]);
}
getch();
return 0;
}

```

Code 8

(διαβάζει τις πωλήσεις διαφόρων πωλητών και υπολογίζει τις συνολικές πωλήσεις για κάθε πωλητή και για κάθε αντικείμενο)

```

#include <stdio.h>
#include <conio.h>
int main()
{
    int sales[5][3], i, j, total_sales=0;
    //INPUT DATA
    printf("\n ENTER THE DATA");
    printf("\n *****");
    for(i=0; i<5; i++)
    {
        printf("\n Enter the sales of 3 items sold by salesman %d: ", i+1) ;
        for(j=0; j<3; j++)
            scanf("%d", &sales[i][j]);
    }
    // PRINT TOTAL SALES BY EACH SALESMAN
    for(i=0; i<5; i++)
    {
        total_sales = 0;
        for(j=0; j<3; j++)
            total_sales += sales[i][j];
        printf("\n Total Sales By Salesman %d = %d", i+1, total_sales);
    }
    // TOTAL SALES OF EACH ITEM
    for(i=0; i<3; i++) // for each item
    {
        total_sales=0;
        for(j=0; j<5; j++) // for each salesman
            total_sales += sales[j][i];
    }
}

```

```

        printf("\n Total sales of item %d = %d", i+1, total_sales);
    }
    getch();
    return 0;
}

```

Οι δισδιάστατοι πίνακες πολλές φορές υιοθετούνται για να πραγματοποιήσουν την μαθηματική υλοποίηση διαφόρων υπολογισμών (π.χ., πολλαπλασιασμός πινάκων).

Code 9

(μετατόπιση ενός πίνακα)

```

#include <stdio.h>
#include <conio.h>
int main()
{
    int i, j, mat[3][3], transposed_mat[3][3];
    printf("\n Enter the elements of the matrix ");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d", &mat[i][j]);
        }
    }
    printf("\n The elements of the matrix are ");
    for(i=0;i<3;i++)
    {
        printf("\n");
        for(j=0;j<3;j++)
            printf("\t %d", mat[i][j]);
    }
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
            transposed_mat[i][j] = mat[j][i];
    }
    printf("\n The elements of the transposed matrix are ");
    for(i=0;i<3;i++)
    {
        printf("\n");
        for(j=0;j<3;j++)
            printf("\t %d",transposed_mat[i][j]);
    }
    return 0;
}

```

Code 10

(άθροισμα πινάκων)

```

#include <stdio.h>
#include <conio.h>

```



```

int main()
{
    int i, j;
    int rows1, cols1, rows2, cols2, rows_sum, cols_sum;
    int mat1[5][5], mat2[5][5], sum[5][5];
    printf("\n Enter the number of rows in the first matrix : ");
    scanf("%d",&rows1);
    printf("\n Enter the number of columns in the first matrix : ");
    scanf("%d",&cols1);
    printf("\n Enter the number of rows in the second matrix : ");
    scanf("%d",&rows2);
    printf("\n Enter the number of columns in the second matrix : ");
    scanf("%d",&cols2);
    if(rows1 != rows2 || cols1 != cols2)
    {
        printf("\n Number of rows and columns of both matrices must be equal");
        getch();
        exit();
    }
    rows_sum = rows1;
    cols_sum = cols1;
    printf("\n Enter the elements of the first matrix ");
    for(i=0;i<rows1;i++)
    {
        for(j=0;j<cols1;j++)
        {
            scanf("%d",&mat1[i][j]);
        }
    }
    printf("\n Enter the elements of the second matrix ");
    for(i=0;i<rows2;i++)
    {
        for(j=0;j<cols2;j++)
        {
            scanf("%d",&mat2[i][j]);
        }
    }
    for(i=0;i<rows_sum;i++)
    {
        for(j=0;j<cols_sum;j++)
            sum[i][j] = mat1[i][j] + mat2[i][j];
    }
    printf("\n The elements of the resultant matrix are ");
    for(i=0;i<rows_sum;i++)
    {
        printf("\n");
        for(j=0;j<cols_sum;j++)
            printf("\t %d", sum[i][j]);
    }
    return 0;
}

```

Code 11

(πολλαπλασιασμός πινάκων)

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i, j, k;
    int rows1, cols1, rows2, cols2, res_rows, res_cols;
    int mat1[5][5], mat2[5][5], res[5][5];
    printf("\n Enter the number of rows in the first matrix : ");
    scanf("%d",&rows1);
    printf("\n Enter the number of columns in the first matrix : ");
    scanf("%d",&cols1);
    printf("\n Enter the number of rows in the second matrix : ");
    scanf("%d",&rows2);
    printf("\n Enter the number of columns in the second matrix : ");
    scanf("%d",&cols2);
    if(cols1 != rows2)
    {
        printf("\n The number of columns in the first matrix must be equal
        to the number of rows in the second matrix");
        getch();
        exit();
    }
    res_rows = rows1;
    res_cols = cols2;
    printf("\n Enter the elements of the first matrix ");
    for(i=0;i<rows1;i++)
    {
        for(j=0;j<cols1;j++)
        {
            scanf("%d",&mat1[i][j]);
        }
    }
    printf("\n Enter the elements of the second matrix ");
    for(i=0;i<rows2;i++)
    {
        for(j=0;j<cols2;j++)
        {
            scanf("%d",&mat2[i][j]);
        }
    }
    for(i=0;i<res_rows;i++)
    {
        for(j=0;j<res_cols;j++)
        {
            res[i][j]=0;
            for(k=0; k<res_cols;k++)
                res[i][j] += mat1[i][k] * mat2[k][j];
        }
    }
    printf("\n The elements of the product matrix are ");
```

```

    for(i=0;i<res_rows;i++)
    {
        printf("\n");
        for(j=0;j<res_cols;j++)
            printf("\t %d",res[i][j]);
    }
    return 0;
}

```

Για το πέρασμα δισδιάστατων πινάκων σε συναρτήσεις υπάρχουν τρεις τρόποι:

- Πέρασμα ενός μόνο στοιχείου
- Πέρασμα μιας γραμμής
- Πέρασμα όλου του πίνακα

Παράδειγμα περάσματος μιας γραμμής:

Calling function	Called function
<pre> main() { int arr[2][3] = ({1, 2, 3}, {4, 5, 6}); func(arr[1]); } </pre>	<pre> void func(int arr[]) { int i; for(i=0;i<3;i++) printf("%d", arr[i] * 10); } </pre>

Code 12

(πέραςμα ολόκληρου πίνακα σε συνάρτηση)

```

#include <stdio.h>
#include <conio.h>
void read_matrix(int mat[5][5], int);
void display_matrix(int mat[5][5], int);
int main()
{
    int row;
    int mat1[5][5];
    printf("\n Enter the number of rows and columns of the matrix:");
    scanf("%d", &row);
    read_matrix(mat1, row);
    display_matrix(mat1, row);
    getch();
    return 0;
}

void read_matrix(int mat[5][5], int r)
{
    int i, j;
    for(i=0; i<r; i++)
    {
        for(j=0; j<r; j++)
        {
            if(i==j)
                mat[i][j] = 0;
            else if(i>j)
                mat[i][j] = -1;
        }
    }
}

```

```

        else
            mat[i][j] = 1;
    }
}

void display_matrix(int mat[5][5], int r)
{
    int i, j;
    for(i=0; i<r; i++)
    {
        printf("\n");
        for(j=0; j<r; j++)
            printf("\t %d", mat[i][j]);
    }
}

```

Code 13

(πολυδιάστατοι πίνακες)

```

#include <stdio.h>
#include <conio.h>
int main()
{
    int array[2][2][2], i, j, k;
    printf("\n Enter the elements of the matrix");
    for(i=0; i<2; i++)
    {
        for(j=0; j<2; j++)
        {
            for(k=0; k<2; k++)
            {
                scanf("%d", &array[i][j][k]);
            }
        }
    }
    printf("\n The matrix is : ");
    for(i=0; i<2; i++)
    {
        printf("\n");
        for(j=0; j<2; j++)
        {
            printf("\n");
            for(k=0; k<2; k++)
                printf("\t array[%d][%d][%d] = %d", i, j, k, array[i][j][k]);
        }
    }
    getch();
    return 0;
}

```