

```

/* Χειρισμός αρχείου εγγραφών σαν αρχείο κειμενου
(λ.χ. .csv και .json αρχεια αν και εκεί η επεξεργασία
θα είναι λίγο διαφορετική λόγω διαφορετικής
τυποποίησης του περιεχομένου) και σαν δυαδικό αρχείο
*/

#include <stdio.h>
#define N 100

typedef struct order
{
    char name[100];
    float price;
    int id;
} ORDER;

ORDER products[N]; // Πίνακας εγγραφών

/* ΠΡΩΤΟΣ ΤΡΟΠΟΣ ΧΕΙΡΙΣΜΟΥ ΠΙΝΑΚΑ ΕΓΓΡΑΦΩΝ ΣΑΝ ΑΡΧΕΙΟ
ΧΕΙΡΙΣΜΟΣ ΜΕ ΑΡΧΕΙΟ ΚΕΙΜΕΝΟΥ*/

int main()
{
    /*
    Χειρισμός με αρχείο κειμένου

    Έναρξη

    FILE * fp=fopen(argv[1], "r");

    for (i=0; i<N; i++)
        fscanf(fp, "%s%f%d", products[i].name, &products[i].price,
        &products[i].id);
        fclose(fp);

    */

    /* κώδικας επεξεργασίας του πίνακα products στην κύρια μνήμη

    .....
    .....
    .....
    */

```

```

        /*          Λήξη Χρήσης

FILE * fp=fopen(argv[1], "w");

    for (i=0; i<N; i++)
        fprintf(fp, "%s %f %d ", products[i].name, products[i].price,
products[i].is);
        fclose(fp);

    Προσέξτε το κενά που βάζουμε στην printf() ώστε να έχουμε τα
δεδομένα στην κατάλληλη μορφή για fprintf()
    */

}

```

```

/*
ΔΕΥΤΕΡΟΣ ΤΡΟΠΟΣ ΧΕΙΡΙΣΜΟΥ ΠΙΝΑΚΑ ΕΓΓΡΑΦΩΝ ΣΑΝ ΑΡΧΕΙΟ
ΧΕΙΡΙΣΜΟΣ ΜΕ ΔΥΑΔΙΚΟ ΑΡΧΕΙΟ */

```

```

    int main()
{
    /* Έναρξη

Χειρισμός με δυαδικό αρχείο

FILE * fp=fopen(argv[1], "rb");

fread(products, sizeof(ORDER), N, fp);
fclose(fp);

    */

/* κώδικας επεξεργασίας του πίνακα products
.....
.....
.....

    */

    /*
        Λήξη Χρήσης

FILE * fp=fopen(argv[1], "w");
fwrite(products, sizeof(ORDER), N, fp);
fclose(fp);
    */ }

```