**Ε.Α.Π./ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ**

**3η ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ**

**ΑΚΑΔΗΜΑΪΚΟΥ ΕΤΟΥΣ 2022-2023**

**4ος Τόμος**

## Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C

**ΕΝΔΕΙΚΤΙΚΑ ΣΧΕΔΙΑ ΛΥΣΕΩΝ - ΥΠΟΔΕΙΞΕΙΣ**

**04/03/2023**

# ΥΠΟΕΡΓΑΣΙΑ 1 (βαθμοί 20)

#include <stdio.h>

#define MAX\_DURATION 10000 */\* H μέγιστη διάρκεια κλήσης \*/*

#define MAX\_HOURS 2 */\* To μέγιστο πλήθος γεμάτων ωρών κλήσης, το οποίο εξαρτάται από τη μέγιστη διάρκεια κλήσης. Στην περίπτωση των 10000 δευτερολέπτων, είναι το ακέραιο μέρος της διαίρεσης 10000 διά 3600 \*/*

main**(){**

system**("chcp 1253");***/\*Εντολή για ορθή εμφάνιση ελληνικών χαρακτήρων στο παράθυρο εξόδου\*/*

**int** DURATION**;** */\* Μεταβλητή για τη διάρκεια κλήσης \*/*

**int** N**;** */\* Μετρητής πλήθους κλήσεων \*/*

**int** I**;** */\* Βοηθητικός μετρητής \*/*

**int** MIN**,** HOUR**;** */\* Mεταβλητές για τον υπολογισμό λεπτών και ωρών \*/*

**int** FREQ\_CALL\_MINUTES**[**60**];** */\*Ο πίνακας στον οποίο αποθηκεύεται η συχνότητα της διάρκειας των κλήσεων για κάθε λεπτό που συμπληρώθηκε \*/*

**int** FREQ\_CALL\_HOURS**[**MAX\_HOURS**];** */\*Ο πίνακας στον οποίο αποθηκεύεται η συχνότητα της διάρκειας των κλήσεων για κάθε ώρα που συμπληρώθηκε \*/*

*/\* Αρχικοποίηση μεταβλητών \*/*

N**=**0**;**

**for** **(**I**=**0**;**I**<**60**;**I**++){**

FREQ\_CALL\_MINUTES**[**I**]=**0**;**

**}**

**for** **(**I**=**0**;**I**<**MAX\_HOURS**;**I**++){**

FREQ\_CALL\_HOURS **[**I**]=**0**;**

**}**

printf**("Δώσε θετικούς ακεραίους αριθμούς στο διάστημα [1..10000] ή εκτός του διαστήματος για να τερματίσεις: \n");**

scanf**("%d",** **&**DURATION**);** */\* Ανάγνωση αριθμού \*/*

**while** **((**DURATION**>=**1**)** **&&** **(**DURATION**<=** MAX\_DURATION**)){**

*/\* Ελέγχουμε αν ο αριθμός ανήκει στο επιτρεπτό διάστημα τιμών \*/*

N**=**N**+**1**;** */\* Aύξηση μετρητή κλήσεων \*/*

MIN**=** DURATION**/**60**;** */\* Υπολογισμός λεπτών \*/*

**if** **(**MIN **<** 60**)** **{**

FREQ\_CALL\_MINUTES**[**MIN**]=** FREQ\_CALL\_MINUTES**[**MIN**]+**1**;**

*/\* Aύξηση συχνότητας λεπτών, στο πρώτο στοιχείο του πίνακα αποθηκεύεται η ομάδα 0 έως 1 λεπτά \*/*

**}**

**else{**

HOUR**=** MIN**/**60**;** */\* Υπολογισμός ωρών \*/*

FREQ\_CALL\_HOURS**[**HOUR**-**1**]=** FREQ\_CALL\_HOURS**[**HOUR**-**1**]+**1**;**

*/\* Aύξηση συχνότητας ωρών, στο πρώτο στοιχείο του πίνακα αποθηκευεται η ομάδα 1 έως 2 ώρες \*/*

**};**

printf**("Δώσε τον επόμενο θετικό ακέραιο αριθμό στο διάστημα [1..10000] ή εκτός του διαστήματος για να τερματίσεις\n");**

scanf**("%d",** **&**DURATION**);** */\* Ανάγνωση επόμενου αριθμού \*/*

**}**

*/\* Εμφάνιση αποτελεσμάτων \*/*

**if** **(**N**==**0**)** **{**

printf**("Δεν πραγματοποιήθηκε καμία κλήση\n");**

**}**

**else{**

printf**("Πραγματοποιήθηκαν %d κλήσεις\n",** N**);**

**for** **(**I**=**0**;**I**<**60**;**I**++){**

**if** **(**FREQ\_CALL\_MINUTES**[**I**]>**0**)**

printf **("Το πλήθος κλήσεων με διάρκεια %d έως <%d λεπτά είναι: %d\n",** I**,** I**+**1**,** FREQ\_CALL\_MINUTES**[**I**]);**

**}**

**for** **(**I**=**0**;**I**<**MAX\_HOURS**;**I**++){**

**if** **(**FREQ\_CALL\_HOURS**[**I**]>**0**)**

printf**("Το πλήθος κλήσεων με διάρκεια %d έως <%d ώρες είναι: %d\n",** I**+**1**,** I**+**2**,** FREQ\_CALL\_HOURS**[**I**]);**

**}**

**}**

system**("PAUSE");***/\*Εντολή που αποτρέπει το αυτόματο κλείσιμο του παραθύρου εξόδου του compiler\*/*

**}** */\* Τέλος προγράμματος \*/*

# ΥΠΟΕΡΓΑΣΙΑ 2 (βαθμοί 25)

#include <stdio.h>

#define SIGNAL\_LENGTH 10

*/\* Ορισμός πινάκων \*/*

**float** signal\_input**[**SIGNAL\_LENGTH**];**

**float** smoothed\_signal**[**SIGNAL\_LENGTH**];**

*/\* Ορισμός συναρτήσεων \*/*

*/\*2α\*/*

**void** initialize\_signal**(float[]);**

*/\*2β\*/*

**void** print\_signal**(float[]);**

*/\*3α\*/*

**void** get\_signal\_input**();**

*/\*3β\*/*

**void** get\_filter\_size**();**

*/\*4\*/*

**void** smoothing**(float[],** **int,** **float[]);**

*/\* User Input\*/*

**int** filter\_size**;**

*/\*5\*/*

**int** main**()**

**{**

system**("chcp 1253");**

*/\*Αρχικοποίηση πίνακα που θα δεχθεί τις τιμές εισόδου\*/*

initialize\_signal**(**signal\_input**);**

*/\*Αρχικοποίηση πίνακα που θα δεχθεί τις τιμές εξομάλυνσης\*/*

initialize\_signal**(**smoothed\_signal**);**

*/\*Συνάρτηση που ζητά από τον χρήστη να δώσει τις τιμές του σήματος\*/*

get\_signal\_input**();**

*/\*Εκτύπωση των τιμών του σήματος εισόδου\*/*

printf**("%s ",** **"Είσοδος");**

print\_signal**(**signal\_input**);**

*/\*Συνάρτηση που ζητά από τον χρήστη να ορίσει το μέγεθος του φίλτρου\*/*

get\_filter\_size**();**

*/\* Καλούμε τη συνάρτηση εξομάλυνσης για το σήμα εισόδου\*/*

smoothing**(**signal\_input**,** filter\_size**,** smoothed\_signal**);**

*/\* Εκτύπωσε το φίλτρο ανάλογα με το μέγεθος που έδωσε ο χρήστης\*/*

printf**("%s ",** **"Σήμα εξομάλυνσης");**

print\_signal**(**smoothed\_signal**);**

**}**

*/\* Αρχικοποίηση σήματος\*/*

**void** initialize\_signal**(float** **\***signal**)**

**{**

**int** i**;**

**for** **(**i **=** 0**;** i **<** SIGNAL\_LENGTH**;** i**++)**

**{**

signal**[**i**]** **=** 0**;**

**}**

**}**

*/\* Εκτύπωση του σήματος\*/*

**void** print\_signal**(float** **\***signal**)**

**{**

**int** i**;**

**for** **(**i **=** 0**;** i **<** SIGNAL\_LENGTH**;** i**++)**

**{**

printf**("%.2f ",** signal**[**i**]);**

**}**

printf**("\n");**

**}**

*/\*\**

*\* Ζητάμε το μέγεθος του φίλτρου από τον χρήστη.*

*\* Ακολουθούμε αμυντικό προγραμματισμό*

*\*/*

**void** get\_filter\_size**()**

**{**

**do**

**{**

printf**("Εισάγετε το μέγεθος του φίλτρου περιττός θετικός αριθμός μικρότερος του 10: ");**

scanf**("%d",** **&**filter\_size**);**

**}** **while** **(** filter\_size **%** 2 **==** 0 **||** filter\_size **>** SIGNAL\_LENGTH **||** filter\_size **<=**0**);**

**}**

*/\*\*\**

*\* Συνάρτηση που δέχεται τις 10 τιμές του σήματος από τον χρήστη*

*\* Ακολουθούμε αμυντικό προγραμματισμό ώστε κάθε τιμή να είναι θετικός αρθμός*

*\*/*

**void** get\_signal\_input**()**

**{**

**int** i**;**

**for** **(**i **=** 0**;** i **<** SIGNAL\_LENGTH**;** i**++)**

**{**

**do**

**{**

printf**("Δώσε θετικό πραγματικό αριθμό για την τιμή %d του σήματος ",** i **+** 1**);**

scanf**("%f",** **&**signal\_input**[**i**]);**

**}** **while** **(**signal\_input**[**i**]** **<** 0**);**

**}**

**}**

*/\*\**

*\* Smoothing*

*\*/*

**void** smoothing**(float** **\***signal**,** **int** filter\_size**,** **float** **\***smoothed**)**

**{**

**float** value**;**

**int** half\_range **=** filter\_size **/** 2**;**

**float** filter **=** 1.0 **/** filter\_size**;**

**int** i**,** r**;**

**for** **(**i **=** 0**;** i **<** SIGNAL\_LENGTH**;** i**++)**

**{**

value **=** 0.0**;**

**for** **(**r **=** **-**half\_range**;** r **<=** half\_range**;** r**++)**

**{**

*/\* Περίπτωση έξω από τα όρια του σήματος i+r < 0\*/*

**if** **((**i **+** r**)** **<** 0**)**

**{**

value **+=** signal**[**0**]** **\*** filter**;**

**}**

*/\* Περίπτωση έξω από τα όρια του σήματος i+r > n-1\*/*

**else** **if** **(** **(**i**+**r**)** **>** SIGNAL\_LENGTH**-**1 **){**

value **+=** signal**[**SIGNAL\_LENGTH**-**1**]** **\*** filter**;**

**}**

*/\* Περίπτωση μέσα στα όρια του σήματος i+r > n-1\*/*

**else** **{**

value **+=** signal**[**i **+** r**]** **\*** filter**;**

**}**

**}**

smoothed**[**i**]** **=** value**;**

**}**

**}**

# ΥΠΟΕΡΓΑΣΙΑ 3 (βαθμοί 25)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define A1 2.5

#define A2 3

#define A3 1

#define N 5

**struct** intellectual\_property **{**

**int** id**;** *//αύξων αριθμός*

**char** title**[**150**];** *//τίτλος έργου*

**char** name**[**50**];** *//ονοματεπώνυμο δημιουργού*

**int** e1**;** *// Αριθμός επιβεβαιωμένων εκτελέσεων του έργου σε ζωντανή συναυλία*

**int** e2**;** *// Αριθμός επιβεβαιωμένων εκτελέσεων του έργου από ραδιοφωνικούς σταθμούς*

**int** e3**;** *// Αριθμός επιβεβαιωμένων εκτελέσεων του έργου μέσω YouTube*

**};**

**struct** intellectual\_property **\***ptr**=**NULL**;**

**struct** intellectual\_property p **[**N**];**

**float** royalty\_per\_work **(int** id**){** *//συνάρτηση υπολογισμού πνευματικών δικαιωμάτων ανά έργο*

**int** f**=**0**;**

**float** A**;**

ptr**=**p**;**

**while(**f**==**0**){**

**if** **(**ptr**->**id**==**id**){**

A**=** ptr**->**e1**\***A1**+**ptr**->**e2**\***A2**+**ptr**->**e3**\***A3**;**

f**=**1**;**

**}**

ptr**++;**

**}**

**return** A**;**

**}**

**float** royalty\_per\_creator**(char** **\***x**){** *//συνάρτηση υπολογισμού συνολικού ποσού πνευματικών δικαιωμάτων για έναν δημιουργό*

ptr**=**p**;**

**int** i**;**

**float** s**=**0.0**;**

**for(**i**=**0**;**i**<**N**;**i**++){**

**if** **(**strcmp**(**x**,**ptr**->**name**)==**0**){**

s**=**s**+** royalty\_per\_work **(**ptr**->**id**);**

**}**

ptr**++;**

**}**

**return** s**;**

**}**

**int** main**()** **{**

system**("chcp 1253>nul");**

**int** i**;**

**char** ans**;**

**char** creator**[**50**];**

**float** sum**;**

ptr**=**p**;**

**for(**i**=**0**;**i**<**N**;**i**++){** *//εισαγωγή στοιχείων στον πίνακα*

ptr**->**id**=**i**+**1**;**

printf**("Εισαγωγή στοιχείων για τo %d έργο\n",**ptr**->**id**);**

printf**("Δώσε τον τίτλο του έργου\n");**

fflush**(**stdin**);**

gets**(**ptr**->**title**);**

printf**("Δώσε το ονοματεπώνυμο του δημιουργού\n");**

fflush**(**stdin**);**

gets**(**ptr**->**name**);**

printf**("Δώσε τον αριθμό επιβεβαιωμένων εκτελέσεων του έργου σε ζωντανή συναυλία. Η αποδεκτή τιμή είναι αριθμός μεγαλύτερος ή ίσος του μηδενός. \n");**

scanf**("%d",&**ptr**->**e1**);**

**while** **(** ptr**->**e1**<** 0**){** *//αμυντικός προγραμματισμός*

printf**("Λάθος τιμή. Η αποδεκτή τιμή είναι αριθμός μεγαλύτερος ή ίσος του μηδενός. \n");**

printf**("Δώσε ξανά τον αριθμό επιβεβαιωμένων εκτελέσεων του έργου σε ζωντανή συναυλία.\n");**

scanf**("%d",&**ptr**->**e1**);**

**}**

printf**("Δώσε τον αριθμό επιβεβαιωμένων εκτελέσεων του έργου από ραδιοφωνικούς σταθμούς. Η αποδεκτή τιμή είναι αριθμός μεγαλύτερος ή ίσος του μηδενός. \n");**

scanf**("%d",&**ptr**->**e2**);**

**while** **(** ptr**->**e2**<** 0**){** *//αμυντικός προγραμματισμός*

printf**("Λάθος τιμή. Η αποδεκτή τιμή είναι αριθμός μεγαλύτερος ή ίσος του μηδενός. \n");**

printf**("Δώσε ξανά τον αριθμό επιβεβαιωμένων εκτελέσεων του έργου από ραδιοφωνικούς σταθμούς. \n");**

scanf**("%d",&**ptr**->**e2**);**

**}**

printf**("Δώσε τον αριθμό επιβεβαιωμένων εκτελέσεων του έργου μέσω YouTube. Η αποδεκτή τιμή είναι αριθμός μεγαλύτερος ή ίσος του μηδενός. \n");**

scanf**("%d",&**ptr**->**e3**);**

**while** **(** ptr**->**e3**<** 0**){** *//αμυντικός προγραμματισμός*

printf**("Λάθος τιμή. Η αποδεκτή τιμή είναι αριθμός μεγαλύτερος ή ίσος του μηδενός. \n");**

printf**("Δώσε ξανά τον αριθμό επιβεβαιωμένων εκτελέσεων του έργου μέσω YouTube. \n");**

scanf**("%d",&**ptr**->**e3**);**

**}**

ptr**++;**

**}**

**do{**

printf**("Επιθυμείς τον υπολογισμό και την εκτύπωση της συνολικής αποτίμησης από όλα τα έργα ενός συγκεκριμένου δημιουργού; Δώσε Ν (ελληνικός χαρακτήρας) για ΝΑΙ ή Ο (ελληνικός χαρακτήρας)για ΟΧΙ. \n");**

fflush**(**stdin**);**

ans**=**getchar**();**

**while** **(**ans**!=**'Ν' **&&** ans**!=**'Ο'**)** **{** *//αμυντικός προγραμματισμός*

printf**("Λάθος τιμή. Η αποδεκτή τιμή είναι Ν (ελληνικός χαρακτήρας) για ΝΑΙ ή Ο (ελληνικός χαρακτήρας) για ΟΧΙ.\n");**

printf**("Δώσε ξανά απάντηση.\n");**

fflush**(**stdin**);**

ans**=**getchar**();**

**}**

**if** **(**ans**==**'Ν'**)** **{**

printf**("Δώσε το ονοματεπώνυμο του δημιουργού\n");**

fflush**(**stdin**);**

gets**(**creator**);**

ptr**=**p**;** *//ο δείκτης να δείχνει στην αρχή του πίνακα*

**int** found**=**0**;** *//μεταβλητή για την αναζήτηση του δημιουργού*

**int** j**=**0**;**

**while** **(**found **==** 0**&&**j**<**N**){** *//αναζήτηση δημιουργού στον πίνακα*

j**++;**

**if** **(**strcmp**(**creator**,**ptr**->**name**)==**0**)**

found**=**1**;**

ptr**++;**

**}**

**if** **(**found**==**0**)**

printf**("Ο δημιουργός %s δεν έχει καταχωριστεί στο σύστημα\n",** creator**);**

**else{**

sum **=** royalty\_per\_creator **(**creator**);** *//κάλεσμα συνάρτηση για τον υπολογισμό του συνολικού ποσού των πνευματικών δικαιωμάτων*

printf**("Ονοματεπώνυμο δημιουργού: %s\n",** creator**);**

ptr**=**p**;**

**for(**i**=**0**;**i**<**N**;**i**++){** *//εκτύπωση δικαιωμάτων για κάθε έργο*

**if** **(**strcmp**(**creator**,**ptr**->**name**)==**0**){**

printf**("Τίτλος έργου: %s, ",** ptr**->**title**);**

printf**("δικαιώματα έργου: %f\n",** royalty\_per\_work**(**ptr**->**id**));**

**}**

**else**

ptr**++;**

**}**

printf**(" Για τον δημιουργό %s το συνολικό ποσό πνευματικών δικαιωμάτων ανέρχεται σε %f ευρώ\n",** creator**,** sum**);**

**}**

**}**

**}while** **(**ans **==** 'Ν'**);**

**return** 0**;**

**}**

# ΥΠΟΕΡΓΑΣΙΑ 4 (βαθμοί 30)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define CUSTOMER\_NAME\_LENGTH 25

#define ITEM\_NAME\_LENGTH 40

**struct** order**{**

**int** orderNum**;**

**char** customerName**[**CUSTOMER\_NAME\_LENGTH**];**

**char** itemName**[**ITEM\_NAME\_LENGTH**];**

**double** itemValue**;**

**struct** order **\***next**;};**

**typedef** **struct** order **\*** nodepointer**;**

nodepointer get\_node**(int** curOrderNum**,** **char** curCustomerName**[**CUSTOMER\_NAME\_LENGTH**],** **char** curItemName**[**ITEM\_NAME\_LENGTH**],** **double** curItemValue**)**

**{** *// ΕΔΩ ΘΑ ΜΠΕΙ Ο ΚΩΔΙΚΑΣ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ ΠΟΥ ΔΗΜΙΟΥΡΓΕΙ ΕΝΑ ΝΕΟ ΚΟΜΒΟ*

**struct** order **\***node**;**

node **=** **(struct** order**\*)**malloc**(sizeof(struct** order**));**

node **->** orderNum **=** curOrderNum**;**

strcpy **(**node **->** customerName**,** curCustomerName**);**

strcpy **(**node **->** itemName**,** curItemName**);**

node **->** itemValue **=** curItemValue**;**

node **->** next **=** NULL**;**

**return(**node**);**

**}**

**void** createOrderAtTheEnd**(**nodepointer **\*** head**,** **int** orderNumToEnter**)**

**{** *// ΕΔΩ ΘΑ ΜΠΕΙ Ο ΚΩΔΙΚΑΣ ΠΟΥ ΕΙΣΑΓΕΙ ΈΝΑ ΝΕΟ ΚΟΜΒΟ ΣΤΟ ΤΕΛΟΣ ΤΗΣ ΛΙΣΤΑΣ*

*// ΓΙΑ ΤΗΝ ΑΝΑΓΝΩΣΗ ΜΙΑΣ ΣΥΜΒΟΛΟΣΕΙΡΑΣ ΠΟΥ ΠΕΡΙΕΧΕΙ ΚΕΝΑ ΜΠΟΡΕΙ ΝΑ ΧΡΗΣΙΜΟΠΟΙΗΘΕΙ Η ΑΚΟΛΟΥΘΗ ΜΟΡΦΗ*

*// char curCustomerName[CUSTOMER\_NAME\_LENGTH];*

*// scanf("%[^\n]%\*c", curCustomerName);*

**struct** order **\***newNode**,** **\***temp**;**

**char** curCustomerName**[**CUSTOMER\_NAME\_LENGTH**];**

**char** curItemName**[**ITEM\_NAME\_LENGTH**];**

**double** curItemValue**;**

fflush**(**stdin**);**

printf**("Παρακαλώ εισάγετε το Ονοματεπώνυμο του πελάτη: ");**

scanf**("%[^\n]%\*c",** **&**curCustomerName**);**

printf**("Παρακαλώ εισάγετε το όνομα του προϊόντος: ");**

scanf**("%[^\n]%\*c",** **&**curItemName**);**

**do{**

printf**("Παρακαλώ εισάγετε το ποσό πληρωμής για το προϊόν: ");**

scanf**("%lf",** **&**curItemValue**);**

**}while(**curItemValue**<=**0**);**

**if(\***head**==**NULL**)**

**\***head **=** get\_node**(**orderNumToEnter**,** curCustomerName**,** curItemName**,** curItemValue**);**

**else{**

newNode **=** get\_node**(**orderNumToEnter**,** curCustomerName**,** curItemName**,** curItemValue**);**

newNode**->**next **=** NULL**;**

temp **=** **\***head**;**

**while(**temp**!=**NULL **&&** temp**->**next**!=**NULL**)**

temp **=** temp**->**next**;**

temp**->**next **=** newNode**;**

**}**

printf**("Η παραγγελία προστέθηκε στη λίστα");**

**}**

**void** displayList**(**nodepointer node**)**

**{** *// ΕΔΩ ΘΑ ΜΠΕΙ Ο ΚΩΔΙΚΑΣ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ ΠΟΥ ΔΙΑΠΕΡΝΑ ΤΗ ΛΙΣΤΑ ΚΑΙ ΕΜΦΑΝΙΖΕΙ ΤΟΥΣ ΚΟΜΒΟΥΣ ΤΗΣ*

**while** **(**node**!=**NULL**){**

printf**("%4d %25s %30s %10.2lf\n",** node**->**orderNum**,** node**->**customerName**,** node**->**itemName**,** node**->**itemValue**);**

node **=** node**->**next**;**

**}**

printf**("\n");**

**}**

**void** printOrderValue**(**nodepointer node**)**

**{** *// ΕΔΩ ΘΑ ΜΠΕΙ Ο ΚΩΔΙΚΑΣ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ ΠΟΥ ΔΙΑΠΕΡΝΑ ΤΗ ΛΙΣΤΑ ΚΑΙ ΕΜΦΑΝΙΖΕΙ ΤΟΝ ΑΡΙΘΜΟ ΤΩΝ ΠΑΡΑΓΓΕΛΙΩΝ ΠΟΥ ΕΚΚΡΕΜΟΥΝ ΚΑΙ ΤΗ ΣΥΝΟΛΙΚΗ ΤΟΥΣ ΑΞΙΑ*

**int** orderCount**=**0**;**

**double** orderValue**=**0**;**

**while** **(**node**!=**NULL**){**

orderCount**++;**

orderValue **+=** node**->**itemValue**;**

node **=** node**->**next**;**

**}**

**if(**orderCount**==**0**)**

printf**("Δεν εκκρεμεί καμμία παραγγελία");**

**else** **if(**orderCount**==**1**)**

printf**("Μέχρι τώρα εκκρεμεί %d παραγγελία με συνολική αξία %10.2lf\n",** orderCount**,** orderValue**);**

**else**

printf**("Μέχρι τώρα εκκρεμούν %d παραγγελίες με συνολική αξία %10.2lf\n",** orderCount**,** orderValue**);**

**}**

nodepointer serveOrder**(**nodepointer head**,** **int** orderNumToServe**)**

**{**

*// ΕΔΩ ΘΑ ΜΠΕΙ Ο ΚΩΔΙΚΑΣ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ ΠΟΥ ΔΙΑΒΑΖΕΙ ΕΝΑ ΑΡΙΘΜΟ ΠΑΡΑΓΓΕΛΙΑΣ*

*// ΘΑ ΕΜΦΑΝΙΖΕΙ ΤΑ ΣΤΟΙΧΕΙΑ ΤΗΣ ΣΤΗΝ ΟΘΟΝΗ ΚΑΙ ΘΑ ΤΗΝ ΑΦΑΙΡΕΙ ΑΠΟ ΤΗ ΛΙΣΤΑ*

\_Bool found**=**0**;**

**struct** order **\***prev**,** **\***cur**;**

prev **=** NULL**;**

cur **=** head**;**

**while(**cur **!=** NULL**){**

**if(**cur**->**orderNum **==** orderNumToServe**){**

found**=**1**;**

**break;**

**}**

prev **=** cur**;**

cur **=** cur**->**next**;**

**}**

**if** **(**found**){**

printf**("Η παραγγελία που θα εξυπηρετηθεί έχει τα παρακάτω στοιχεία: \n");**

printf**("%4d %25s %30s %10.2lf\n",** cur**->**orderNum**,** cur**->**customerName**,** cur**->**itemName**,** cur**->**itemValue**);**

**if(**prev**!=** NULL**){**

prev**->**next **=** cur**->**next**;**

free**(**cur**);**

**}**

**else{**

prev**=**head**;**

head **=** head**->**next**;**

free**(**prev**);**

**}**

**}**

**else**

printf**("Η παραγγελία που θέλετε να εξυπηρετηθεί δεν υπάρχει στη λίστα των παραγγελιών\n");**

**return(**head**);**

**}**

**int** main**()** **{**

system **("chcp 1253>NULL");**

**struct** order **\***head**;**

**int** menuValue**;**

**int** orderNumToEnter**=**100**;**

**int** orderNumToServe**;**

head **=** NULL**;**

**do{**

printf**("Μενού επιλογών.\n");**

printf**("1. Εισαγωγή νέας παραγγελίας\n");**

printf**("2. Εμφάνιση παραγγελιών\n");**

printf**("3. Εξυπηρέτηση παραγγελίας\n");**

printf**("4. Αξία παραγγελιών που εκκρεμούν\n");**

printf**("5. Έξοδος\n");**

printf**("Εισάγετε επιλογή: ");** scanf**("%d",&**menuValue**);**

**switch** **(**menuValue**){**

**case** 1**:**

createOrderAtTheEnd**(&**head**,** orderNumToEnter**);**

orderNumToEnter**++;** *// ΚΑΘΕ ΝΕΑ ΠΑΡΑΓΓΕΛΙΑ ΕΧΕΙ ΑΡΙΘΜΟ ΑΥΞΗΜΕΝΟ ΚΑΤΑ 1*

**break;**

**case** 2**:**

**if** **(**head**!=**NULL**)**

displayList**(**head**);**

**else**

printf**("Η λίστα των παραγγελιών είναι άδεια");**

**break;**

**case** 3**:**

**if** **(**head**!=**NULL**){**

printf**("Εισάγετε τον αριθμό της παραγγελίας που θέλετε να εξυπηρετηθεί: ");**

scanf**("%d",** **&**orderNumToServe**);**

head **=** serveOrder**(**head**,** orderNumToServe**);**

**}**

**else**

printf**("Η λίστα των παραγγελιών είναι άδεια \n");**

**break;**

**case** 4**:**

**if** **(**head**!=**NULL**)**

printOrderValue**(**head**);**

**else**

printf**("Η λίστα των παραγγελιών είναι άδεια \n");**

**break;**

**case** 5**:**

**break;**

**default:**

**break;**

**}**

printf**("\n");**

**}** **while** **(**menuValue**!=**5**);**

**}**