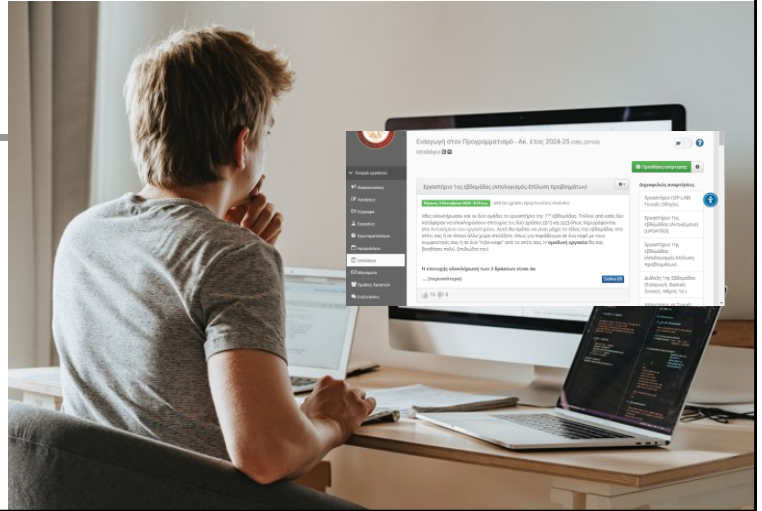


Εισαγωγή στον Προγραμματισμό

(CEID_NY131)

Οργάνωση Εβδομαδιαίων Δραστηριοτήτων Μαθήματος

Kleanthis Thramboulidis
Prof. of Software and System Engineering
University of Patras
<https://sites.google.com/site/thramboulidiskleanthis/>



Week 1 - Αντικείμενο

- Σημασία, Στόχος και Οργάνωση του Μαθήματος
 - Εργαστήριο (Το βασικότερο συνθετικό του Μαθήματος)
- Βασικές έννοιες Διαδικαστικού Προστακτικού Προγραμματισμού
 - με τη χρήση παραδειγμάτων από την καθημερινή ζωή
- Αφαιρετικότητα στις Διεργασίες (procedural abstraction)
 - Όνομα – Περιγραφή Διεργασίας
- Διαδικασία Ανάπτυξης Προγράμματος
 - Βασικά Βήματα (Λεκτική Περιγραφή, Η μετατροπή σε Πρόγραμμα)
 - Ο ρόλος του IDE στην διαδικασία αυτή
 - Οργάνωση αποθηκευτικού χώρου
 - Το πρόγραμμα HelloWorld

Week 1 - Το βασικό Εργαλείο (ΕΑ-1)

Εργαστήριο 1ης εβδομάδας (Αντικείμενο) [UPDATED]

Δευτέρα, 30 Σεπτεμβρίου 2024 - 6:14 μ.μ.

από τον χρήστη Θραμπουλίδης Κλεάνθης

Αντικείμενο 1^{ου} Εργαστηρίου:

Εισαγωγή στο περιβάλλον του εργαστηρίου

Το 1ο εργαστήριο έχει στόχο τη γνωριμία του χώρου διεξαγωγής καθώς και του υπολογιστικού περιβάλλοντος που θα χρησιμοποιούμε κατά τη διάρκεια του εξαμήνου.

Εισαγωγή στο περιβάλλον του εργαστηρίου

Δράσεις της ΕΑ-1

■ Αξιοποίηση του Νέφους ως αποθηκευτικού χώρου

■ Οργάνωση αποθηκευτικού χώρου

■ Εγκατάσταση DevCpp

■ Διαδικασία Ανάπτυξης Προγράμματος με τη χρήση IDE

■ Η περίπτωση του DevCpp

■ Το πρόγραμμα HelloWorld

Κλεάνθης Θραμπουλίδης

Week Agenda

3

Week 1 - Παραδοτέο ΕΑ-1

Τίτλος

Αξιολόγηση 1ης Εβδομάδας

Ανάθεση σε: Συγκεκριμένες Ομάδες Χρηστών

Βαθμολογήστε από 0 (καθόλου) έως 5 (πάρα πολύ) τα παρακάτω ως προς το πόσο σας βοήθησαν για την επιτυχή εκτέλεση του 1^{ου} Εργαστηρίου.

Δράσεις 1ης Εβδομάδας

Ανάθεση σε: Συγκεκριμένες Ομάδες Χρηστών

Το ερωτηματολόγιο είναι το παραδοτέο σας για το εργαστήριο της 1^{ης} εβδομάδας και ως εκ τούτου η συμπλήρωσή του είναι υποχρεωτική.

Αποτελέσματα Ερωτηματολογίου

Κλεάνθης Θραμπουλίδης

Week Agenda

4

Week 2 - Agenda 1/3 (Add2Numbers Version1)

Ένα πιο σύνθετο πρόγραμμα (**Add2Numbers Version1**)

- Μετατροπή Λεκτικής περιγραφής σε πηγαίο κώδικα
- Δημιουργία Project Add2Numbers
- Είσοδος δεδομένων (η συνάρτηση scanf)
- Η σημασία της Μεταβλητής
- Η σημασία των τύπων δεδομένων
- Οι Τελεστές ως βασικές διεργασίες
- Έκφραση
- Δομή ενός C προγράμματος

Week 2 - Agenda 2/3 (Add2Numbers Version2)

Η δική μας πρώτη συνάρτηση (**getNumber()**)

- **Δήλωση** Συνάρτησης
- **Ορισμός** Συνάρτησης
- **Κλήση** Συνάρτησης

Μια πιο σύνθετη Συνάρτηση (**sum()**)

- Η έννοια της επαναχρησιμοποίησης (reusability)
- Ορίσματα συνάρτησης
- Επιστρεφόμενη τιμή

Week 2 - Agenda 3/3 (HS - Fibonacci numbers)

Ανάθεση έργου στη Μηχανή – Παραδείγματα

- **Δύο πιο σύνθετα προβλήματα**
 - Fibonacci numbers
 - Hailstone sequence
- **Hailstone sequence** (Λεκτική περιγραφή – 3 εκδοχές)
- Η επικοινωνία με την μηχανή με την χρήση της ακολουθίας αριθμών Fibonacci (Fibonacci numbers)
- Συγγραφή πηγαίου κώδικα

Week 3 - Week 2 Αντικείμενο

- Μετατροπή Λεκτικής περιγραφής σε πηγαίο κώδικα (Add2Numbers)
- Δημιουργία Project Add2Numbers – Οργάνωση αποθηκευτικού χώρου
- Είσοδος δεδομένων (η συνάρτηση scanf)
- Η σημασία της Μεταβλητής
- Η σημασία των τύπων δεδομένων
- Οι Τελεστές ως βασικές διεργασίες
- Έκφραση
- Συνάρτηση ως αυτόνομη επαναχρησιμοποιήσιμη μονάδα κώδικα
- Δομή ενός C προγράμματος

Όλα τα παραπάνω τα είδαμε μέσα από ένα πιο σύνθετο πρόγραμμα (**Add2Numbers Version1, Version2, Version3**)

Week 3 - Week 2 Το βασικό Εργαλείο (EA-2)

Εργαστήριο 1ης εβδομάδας (Αντικείμενο) [UPDATED]

Δευτέρα, 30 Σεπτεμβρίου 2024 - 6:14 μ.μ. - από τον χρήστη Θραμπουλίδης Κλεάνθης

Αντικείμενο 1^ο Εργαστηρίου: Εισαγωγή στο περιβάλλον του εργαστηρίου

Το 1ο εργαστήριο έχει στόχο τη γνωριμία του χώρου διεξαγωγής καθώς και του υπολογιστικού περιβάλλοντος που θα χρησιμοποιούμε κατά τη διάρκεια του εξαμήνου.

add2Numbers

Δράσεις της EA-2

- Αξιοποίηση του Νέφους ως αποθηκευτικού χώρου
 - Οργάνωση αποθηκευτικού χώρου
- Δημιουργία project για κάθε πρόγραμμα
- Διαδικασία Ανάπτυξης Προγράμματος με τη χρήση IDE
 - Η περίπτωση του DevCpp
- Ορισμός δικών μας συναρτήσεων

Κλεάνθης Θραμπουλίδης

Week Agenda

9

Week 3 - Week 2 Παραδοτέο EA-2

1^ο Μέρος

Όνομα Άσκησης	Ρυθμίσεις άσκησης	Αποτελέσματα
Παραδοτέο Εργαστηρίου 2ης Εβδομάδας (Μέρος 1ο) (έχει λήξει) Όλες οι ερωτήσεις της άσκησης αφορούν υλικό που έχουμε δουλέψει μέχρι σήμερα. Πριν από την εκτέλεση της άσκησης διαβάστε προσεκτικά τις Οδηγίες εκτέλεσης άσκησης στο eclass	Έναρξη: 11/10/24, 5:00 μ.μ. Λήξη: 13/10/24, 11:59 μ.μ. Διάρκεια: 20 λεπτά Προσπάθειες: 1 Προσωρινή αποθήκευση: Ναι	Εμφάνιση 85 υποβολές

2^ο Μέρος

Τίτλος	Ημερομηνία	Απαντήσεις
Παραδοτέο Εργαστηρίου 2ης Εβδομάδας (Μέρος 2ο) Ανάθεση σε: Συγκεκριμένες Ομάδες Χρηστών	Από: Παρασκευή, 11 Οκτωβρίου 2024 - 7:00 μ.μ. έως: σήμερα - 2:00 π.μ.	59

Αποτελέσματα Ερωτηματολογίου

Εργασία Ομάδας 2: (Σε 40 παραδοτέα) V3-20, V2-9, V1-4, Fail-7

Κλεάνθης Θραμπουλίδης

Week Agenda

10

Week 3 - Week 3 Αντικείμενο

- Η επικοινωνία με τη μηχανή – Η σημασία της Αφαιρετικότητας
- Προτάσεις Επανάληψης
- Μια πιο αφαιρετική αναπαράσταση Διεργασίας και η σημασία της
- Διαδικασία Ανάπτυξης Προγράμματος (Updated)
- Fibonacci numbers (Ανάπτυξη προγράμματος)
- Εισαγωγή στους Πίνακες (Arrays)
- Η διεργασία ως επαναχρησιμοποιήσιμη μονάδα κώδικα (συνάρτηση)
- Υπολογισμός μέσου όρου (Average) (Ανάπτυξη προγράμματος)
- Πίνακες ως όρισμα σε συνάρτηση
 - Η Συνάρτηση average()

Τι κάνουμε αν έχουμε ερωτήσεις;

- Δείτε πρώτα από όλα αν η ερώτηση σας έχει ήδη απαντηθεί (**FAQs.pdf**)
- Εκφράστε την ερώτηση σας δημόσια.
 - Πιθανότατα αφορά και τους συμφοιτητές σας.
- ~~Για άμεση επικοινωνία χρησιμοποιήστε την ομάδα στο FB.~~
- Εναλλακτικά χρησιμοποιείτε την περιοχή συζητήσεων του eclass
- Αξιοποιήστε την δια ζώσης επικοινωνία στις διαλέξεις, τα διαλείμματα, το φροντιστήριο, το εργαστήριο.
- Αν η ερώτηση σας είναι προσωπική επικοινωνήστε με τους υπευθύνους του μαθήματος.

Παρακαλώ πολύ **ερωτήσεις με email μόνο όταν έχετε εξαντλήσει τους παραπάνω τρόπους**. Στην περίπτωση αυτή βάλτε στον τίτλο του μηνύματος την έκφραση **12P-Urgent** Μόνο σε αυτές τις περιπτώσεις θα περιμένετε απάντηση από εμένα. Οι προσωπικές σας απόψεις για το μάθημα ανήκουν στην κατηγορία αυτή και είναι ευπρόσδεκτες.

Week 4 - Week 3 Παραδοτέο ΕΑ-3

Τίτλος		Υποβλ.				
Παραδοτέο Εργαστηρίου 3ης εβδομάδας (Μέρος 1ο)		79				
Ατομική εργασία		Title	Uploaded	Viewed	Grade	Similarity
Ανάθεση σε: Συγκεκριμένες Ομάδες Χρηστών		hailstone versions	October 21, 2024			81%
		HailstoneSequence (V1.2, ...	October 20, 2024			98%
		askisi ergastiriou	October 20, 2024			38%
		ΠΑΡΑΔΟΤΕΟ.pdf	October 20, 2024			68%
		Hailstone neofitos spyrid...	October 20, 2024			50%

Όνομα Άσκησης	2ο Μέρος	Ρυθμίσεις άσκησης	Αποτελέσματα
Παραδοτέο Εργαστηρίου 3ης Εβδομάδας (Μέρος 2ο) (έχει λήξει) Όλες οι ερωτήσεις της Άσκησης αφορούν υλικό που έχουμε δουλέψει μέχρι σήμερα. Για τον λόγο αυτό εκτελέστε την άσκηση μετά από την προσεκτική μελέτη		Έναρξη: 19/10/24, 10:00 π.μ. Λήξη: 20/10/24, 11:59 μ.μ. Διάρκεια: 35 λεπτά	Εμφάνιση 81 υποβολές

10<20min, 18<25min, 32<30min

Κλεάνθης Θραμπουλίδης

Week Agenda

13

Χαρακτηριστικές Απαντήσεις 1/2

Δώστε την πρόταση για την εμφάνιση στην οθόνη του 3ου στοιχείου του πίνακα που δηλώνεται από την πρόταση int fbh[20].
Απάντηση
printf("Το 3ο στοιχειο του πινακα ειναι:%d\n", fbh[20]);
[Fibonacci Numbers] Ποια από τις δύο προτάσεις ελέγχου ροής προγράμματος for και while θα επιλέγατε και γιατί;
Απάντηση
Θα χρησιμοποιουσα την προταση while επειδη ειναι πιο ευελικτη εφοσον δεν χρειαζεται να γνωριζουμε ακριβως το χρονικο διαστημα για το ποιο θα διασκεσει η εκτελεση του προοναυματος.
Δώστε τη δήλωση μιας συνάρτησης που δέχεται ένα πίνακα αριθμών κινητής υποδιαστολής απλής ακρίβειας και επιστρέφει το άθροισμα των περιττής τάξης στοιχείων του.
Απάντηση
[Hailstone Sequence] Δράση [Δ3]: Τι προσθέσατε στον κώδικα της V1.1 για να πάρετε αυτόν της V1.2;
Απάντηση
Κλεάνθης Θραμπουλίδης
Week Agenda

Χαρακτηριστικές Απαντήσεις 2/2

Δώστε τη δήλωση μιας συνάρτησης που δέχεται ένα πίνακα αριθμών κινητής υποδιαστολής απλής ακρίβειας και επιστρέφει την τάξη του στοιχείου του που έχει την μεγαλύτερη τιμή.

Απάντηση

```
int findMaxIndex(float arr[], int size) {  
    int maxIndex = 0;  
    for (int i = 1; i < size; i++) {  
        if (arr[i] > arr[maxIndex]) {  
            maxIndex = i;  
        }  
    }  
    return maxIndex;  
}
```

Τι κάνει ο υπολογιστής όταν εκτελεί την πρόταση `int fbn[24];`;

Απάντηση Βρίσκει το στοιχείο του πίνακα `fbn` που βρίσκεται στην θέση 24 και το διαβάζει.

Διάρκεια 7 λεπτά
2 Απαντήσεις και αυτές λάθος

```
int max(int nums[]) {  
    int i = 0;  
    int max = 0;  
    for (i=0; i < sizeof(nums)/sizeof(nums[0]); {  
        if (nums[i] > max ) {  
            max=nums[i];  
        }  
    }  
}
```

[Average] Τι ρόλο παίζει η τοποθέτηση του `(float)` πριν από την μεταβλητή `sum` στην έκφραση υπολογισμού της τιμής της μεταβλητής που αναπαριστά τον μέσο όρο;

Απάντηση

Σε περίπτωση που έχουν δοθεί `float` αριθμοί για την πράξη το αποτέλεσμα θα διαφέρει καθώς αυτά τα ψηφία μετά το κόμμα θα ήταν υπό χρήσης `int` 0

Απάντηση

Με την τοποθέτηση `float` πριν την μεταβλητή `sum` επιτρέπουμε στην `sum` να πάρει τιμές ακεραίων και δεκαδικών.

Κλεάνθης Θραμπουλίδης

Week Agenda

HailstoneSequence – [S1, S2] Αφαιρετική αναπαράσταση

Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα δέχεται έναν αριθμό και θα εμφανίζει την hailstone sequence για τον αριθμό αυτό.



- Step 2**
- A.1. Ζήτη ένα θετικό ακέραιο αριθμό
 - A.2. Πάρε τον αριθμό
 - A.3. **Επανάλαβε μέχρι ο αριθμός να γίνει 1**
 - A3.1 Αν ο αριθμός είναι άρτιος
Διαίρεσε τον με το 2
 - Διαφορετικά
Πολλαπλασίασε τον με το 3 και πρόσθεσε το 1
 - A3.2 Τύπωσε τον αριθμό

3^η εκδοχή

Κλεάνθης Θραμπουλίδης

Week Agenda

16

HailstoneSequence – [S3]

1
Πάρε τον αριθμό
Επανάλαβε όσο ο αριθμός είναι διάφορος του 1
Αν ο αριθμός είναι άρτιος
 διαίρεσε με το 2.
Αλλιώς
 πολλαπλασίασε με το 3 και πρόσθεσε το 1
Τύπωσε τον αριθμό

```
while(num != 1){  
    if(num%2==0)  
        num=num/2;  
    else {  
        num = num*3;  
        num = num+1;  
    }  
    printf("%d\n", num);  
}
```

2
Πάρε τον αριθμό
Επανάλαβε όσο ο αριθμός είναι διάφορος του 1
Υπολόγησε επόμενο όρο ακολουθίας
Εμφάνισε τον όρο

```
while(num!=1){  
    num=HSNextTerm(num);  
    printf("%d\n", num);  
}
```

3
Πάρε τον αριθμό
Υπολόγησε και Εμφάνισε Hailstone ακολουθία

 **calcAndDisplayHailstoneSequence(num);**

Αναπτύξτε μια 4^η έκδοση με βάση τις συναρτήσεις *calcHailstoneSequence* και *displayHailstoneSequence*

Week 4 - Week 4 Αντικείμενο

- **Είσοδος – Έξοδος** ([Άσκηση 1 Κεφάλαιο 11](#))
 - Συναρτήσεις Εισόδου/Εξόδου της βασικής βιβλιοθήκης ([C stdio Functions stdio.h](#))
- **Ο τύπος του χαρακτήρα (char)** ([I2P Variables&DataTypes.pdf](#) s12,13,17-21)
 - American Standard Code for Information Interchange (ASCII).
- **Αλφαριθμητικά (String)** ([Άσκηση 2 κεφάλαιο 11](#)) ([I2P Variables&DataTypes.pdf](#), s23)
 - Το αλφαριθμητικό ως πίνακας ([I2P Arrays.pdf](#)) χαρακτήρων
 - Συναρτήσεις διαχείρισης αλφαριθμητικών της βασικής βιβλιοθήκης ([C string Functions string.h](#))
 - Χαρακτηριστικές συναρτήσεις: strlen(), strcat(), strcmp(), strcpy(), strncat(), strncpy(), ...
- **Δείκτες (Pointers)** ([Άσκηση 2 κεφάλαιο 11](#)) ([I2P Pointers.pdf](#))
 - Το όνομα του πίνακα ως διεύθυνση του πρώτου στοιχείου του πίνακα

Week 5 - Οργάνωση

■ Παραδοτέο 4^{ης} εβδομάδας (MinAndMax)

- Σχολιασμός Παραδοτέου
- Εναλλακτικές υλοποιήσεις MinAndMax (modularity)

■ String Handlings (addressBook case study)

- getString based on getchar()
- (Άσκηση 4 – Κεφάλαιο 11)

■ Ασκήσεις 5^{ης} Εβδομάδας

- Άσκηση 3 (Κεφάλαιο 11) – Τέσσερις Πράξεις
- Άσκηση 4 (Κεφάλαιο 11) - Ταξινόμηση Λέξεων

Παραδοτέο 4^{ης} Εβδομάδας

Εργαστήριο 4ης εβδομάδας (Παραδοτέο)

Παρασκευή, 25 Οκτωβρίου 2024 - 10:30 π.μ. - από τον χρήστη Θραμπουλίδης Κλεάνθης

Το παραδοτέο της 4^{ης} εβδομάδας αποτελείται από μια Εργασία η οποία περιλαμβάνει 2 αρχεία pdf:

1. Το ένα θα περιέχει τον πηγαίο κώδικα των εκδόσεων V1 και V2 που έχετε αναπτύξει και τρέχουν σωστά για την Δράση [Δ3] – MinAndMax (αν έχετε αναπτύξει μια στέλνετε μόνο μία). Το pdf που θα παραδώσετε θα το δημιουργήσετε από ένα αρχείο πηγαίου κώδικα στο οποίο θα κάνετε copy-paste τους πηγαίους κώδικες από τα αντίστοιχα αρχεία πηγαίου κώδικα των δύο εκδόσεων. Πριν από τον πηγαίο κώδικα κάθε έκδοσης θα βάλετε σε σχόλιο την έκδοση του, π.χ. // V1, //V2
2. Το άλλο θα περιέχει:

α) την αφαιρετική αναπαράσταση του προγράμματος, και,

β) τις λεκτικές περιγραφές των 2 εκδόσεων που έχετε περιλάβει στο αρχείο πηγαίου κώδικα με την ένδειξη της έκδοσης πριν από κάθε περιγραφή. Οι λεκτικές περιγραφές θα πρέπει να είναι σύμφωνες με τον πηγαίο κώδικα σας.

👍 2 🗨️ 0

Εργαστήριο 4ης εβδομάδας (Αντικείμενο) [Updated]

👍 3 🗨️ 0

Παράδειγμα Παραδοτέου



A) αφαιρετική παράσταση του προγράμματος V1:

Στόχος του προγράμματος είναι να διαβάσει ένα τιμές από τον χρήστη και να υπολογίζει τη μικρότερη και την μεγαλύτερη τιμή.

1. Ορίζω τις μεταβλητές min και max στις οποίες αποθηκεύονται οι άκαιρες τιμές και την μεταβλητή numberOfTerms που ορίζει το πλήθος των αριθμών που θα εισάγει ο χρήστης.
2. Ο χρήστης καλείται να δώσει το πλήθος των τιμών και στην συνέχεια τους αριθμούς έναν.
3. Για κάθε τιμή που εισάγεται αν είναι μεγαλύτερη από την τρέχουσα μεγαλύτερη ενημερώνεται η Max με την νέα τιμή, όμοιος αν η τιμή που εισάγεται είναι μικρότερη από την τρέχουσα μικρότερη ενημερώνεται η min.
4. Τέλος, εμφανίζονται οι τελικές τιμές min και max.

“έκανα την υποβολή των 2 pdf και μ ήρθε μήνυμα στο ακαδημαϊκό email ότι ο βαθμός είναι 0. Είναι ο βαθμός μου στο παραδοτέο ή κάποιο λάθος του συστήματος γιατί **έκανα ακριβώς ότι ζητούσαν οι οδηγίες**. Σας ευχαριστώ για τον χρόνο σας.”



B) Λεκτική περιγραφή του προγράμματος V1:

Μετά την υποβολή σας εμφανίζεται βαθμός 0. Τι θα κάνετε;

1. Το πρόγραμμα ζητείται από τον χρήστη το πλήθος των αριθμών π θα βάλει.
2. ο χρήστης εισάγει τις τιμές μια.
3. το πρόγραμμα έχει μια τιμή min και max.
4. για κάθε αριθμό που εισάγει ο χρήστης ενημερώνεται η min αν η τιμή είναι μικρότερη από την min και η τιμή max αν η νέα τιμή είναι μεγαλύτερη από την max.
5. Τέλος εμφανίζει το μικρότερο και τον μεγαλύτερο αριθμό.

Week 6 - Οργάνωση

■ Παραδοτέο 5^{ης} εβδομάδας

- Σχολιασμός Παραδοτέου
- Προετοιμασία EA-5

■ String Handlings (addressBook case study)

- getString based on getchar() (getStringV2())
- Βιβλιοθήκη Τρίτου κατασκευαστή (third party library) – **i2p library**

■ Menu and Words Handling

- Δόμηση πηγαίου κώδικα προγράμματος
- Αξιοποίηση Αφαιρετικότητας – Αυξητική Ανάπτυξη

Παραδοτέο 5ης Εβδομάδας

[EA-5] Εργαστήριο 5ης και 6ης εβδομάδας (Αντικείμενο)

■ [Δ1] – StringHandling
(Διαχείριση Αλφαριθμητικού)

■ [Δ2] – MinAndMax
(Μέγιστο και ελάχιστο σε Ακολουθία αριθμών)

■ [Δ3] – Τέσσερις Πράξεις (Menu)

■ [Δ4] – Διαχείριση Λέξεων (WordsHandling)

■ Η βιβλιοθήκη i2p

προϋποθέτει την προσεκτική παρακολούθηση διά λέξης και φροντιστηρίου και μελέτη του αντίστοιχου υλικού στο eclass).

Παραδοτέο Εργαστηρίου 5ης Εβδομάδας
Ανάθεση σε: Συγκεκριμένες Ομάδες Χρηστών

Από: Παρασκευή, 1 Νοεμβρίου 2024 - 6:30 μ.μ.
έως: χθες - 11:59 μ.μ.

41

Κλεάνθης Θραμπουλίδης

Week Agenda

23

βαθμός δυσκολίας EA-4

[Δ1] – BasicInputOutput (Βασική Είσοδος/Εξοδος)

[Δ2] – StringHandling (Διαχείριση Αλφαριθμητικού)

[Δ3] – MinAndMax (Μέγιστο και ελάχιστο σε Ακολουθία αριθμών)

1- Πολύ εύκολη

2 - Εύκολη

3 - Βατή

4 - Δύσκολη

5 - Πολύ δύσκολη

EA-4

Level	Count
1	10
2	12
3	19
4	12
5	2

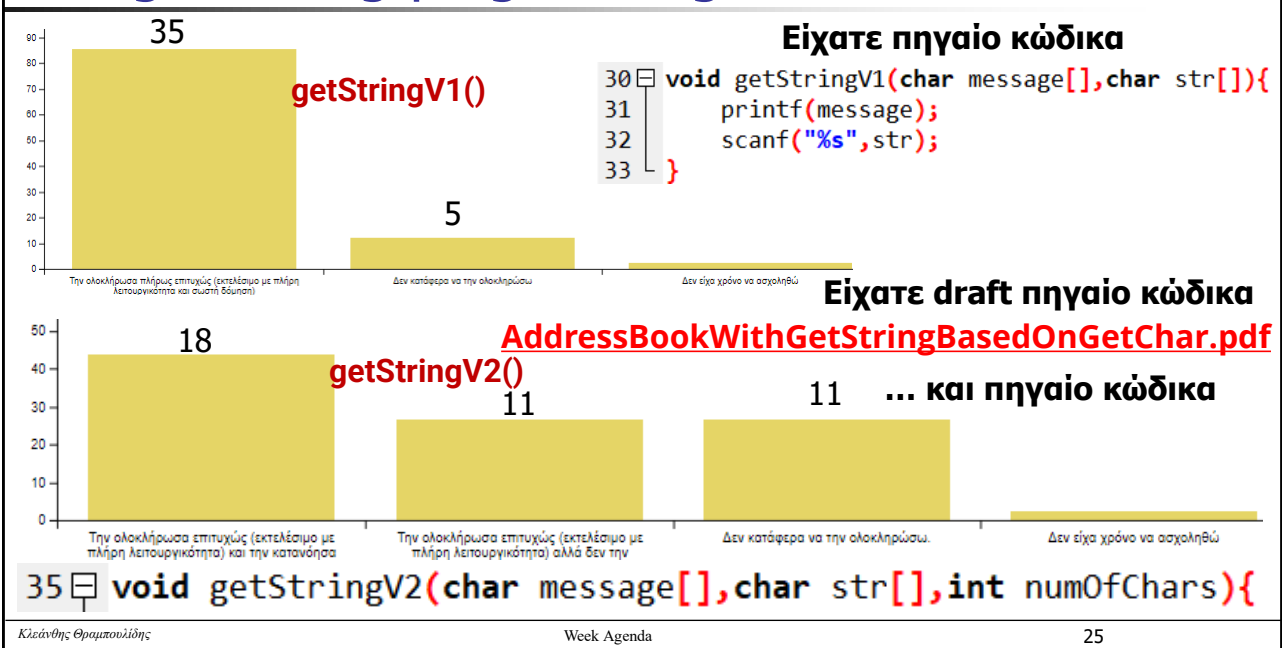
<https://eclass.upatras.gr/modules/blog/index.php?course=CEID1456&action=showPost&pId=1716>

Κλεάνθης Θραμπουλίδης

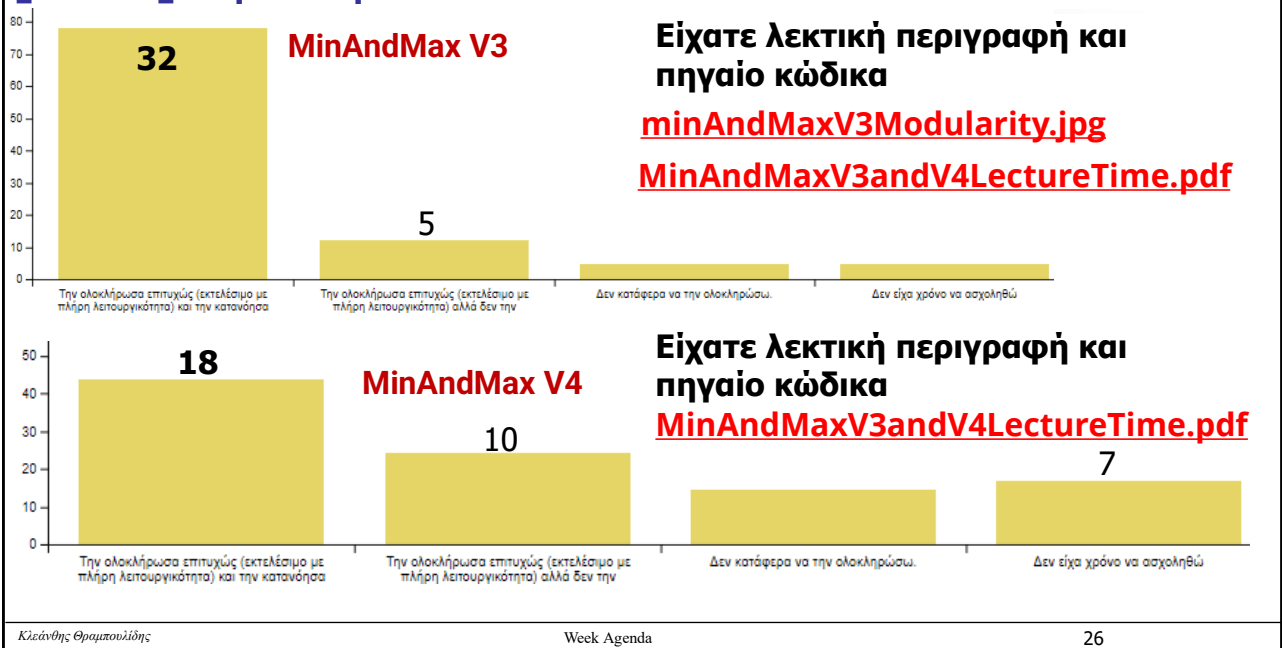
Week Agenda

24

StringHandling με getStringV1 και V2



[EA-5] Δράση Δ2 – MinAndMax V3 and V4



βαθμός δυσκολίας EA-5

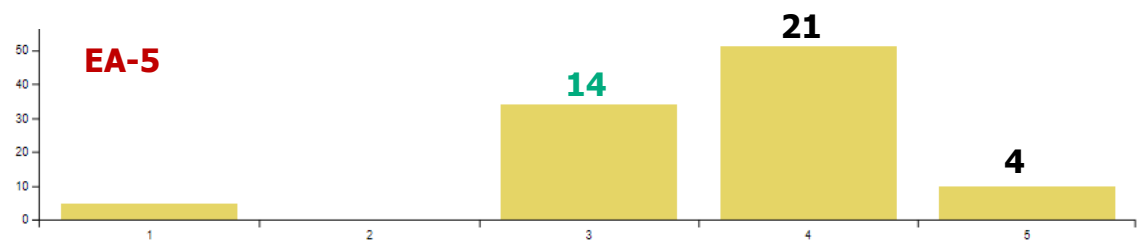
- 1 - Πολύ εύκολη
- 2 - Εύκολη
- 3 - Βατή
- 4 - Δύσκολη
- 5 - Πολύ δύσκολη

[Δ1] – StringHandling (Διαχείριση Αλφαριθμητικού)

[Δ2] – MinAndMax (Μέγιστο και ελάχιστο σε Ακολουθία αριθμών)

[Δ3] – Τέσσερις Πράξεις (Menu)

[Δ4] – Διαχείριση Λέξεων (WordsHandling)



<https://eclass.upatras.gr/modules/blog/index.php?course=CEID1456&action=showPost&pId=1726>

Κλεάνθης Θραμπουλίδης

Week Agenda

27

[EA-5] Δράση Δ3 – Menu



Δώσε λεκτική περιγραφή

Λεκτική Περιγραφή

Εμφάνισε μενού (λειτουργίες προγράμματος)

Πάρε επιλογή χρήστη

Όσο η επιλογή δεν είναι τερματισμός προγράμματος

Εκτέλεσε τη ζητούμενη λειτουργία

Εμφάνισε μενού και πάρε την επιλογή χρήστη

Κλεάνθης Θραμπουλίδης

Week Agenda

28

Βιβλιοθήκη I2P

Οδηγίες χρήσης της βιβλιοθήκης I2P

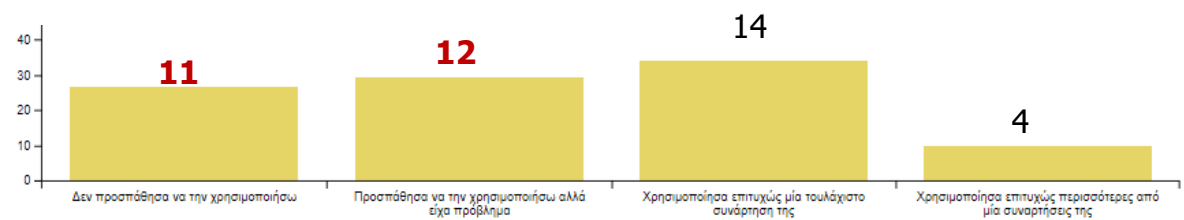
Οδηγίες χρήσης της της βιβλιοθήκης I2P θα βρείτε στο [i2pLibraryV4part.pdf](#)

Είχατε οδηγίες Χρήσης
[i2pLibraryV4part.pdf](#)

Μπορείτε να χρησιμοποιήσετε και τις συναρτήσεις `getInt()` και `getArrayOfInts()` των οποίων οι δηλώσεις είναι οι παρακάτω:

```
int getArrayOfInts(int ar[],int maxItems);
```

```
int getInt(char message[]);
```



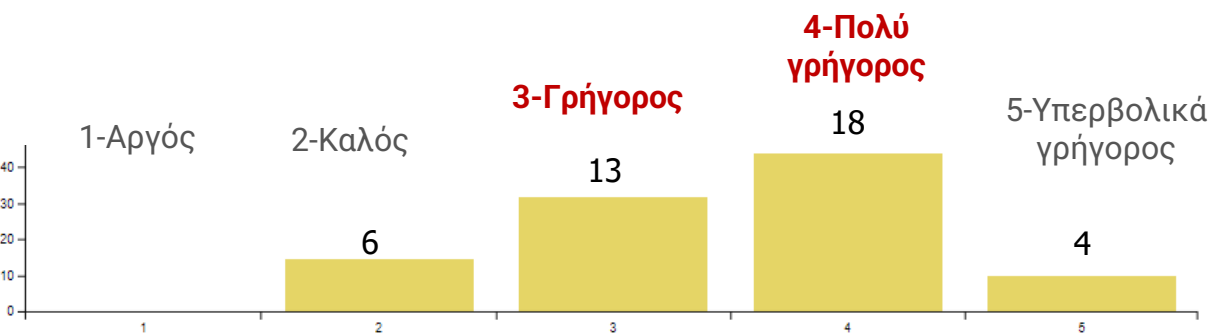
Κλεάνθης Θραμπουλίδης

Week Agenda

29

ρυθμός μαθήματος συνολικά

Σημειώστε πως χαρακτηρίζετε τον ρυθμό με τον οποίο προχωράμε στο μάθημα συνολικά 1-Αργό, 2-Καλό, 3-Γρήγορο, 4-Πολύ γρήγορο, 5-Υπερβολικά γρήγορο



Κλεάνθης Θραμπουλίδης

Week Agenda

30

Λεκτική περιγραφή της getStringV2

1.Ενημέρωσε χρήστη να δώσει όνομα 2.Πάρε όνομα 3.Ενημέρωσε χρήστη να δώσει επίθετο 4.Πάρε επίθετο 5.Ενημέρωσε χρήστη να δώσει οδό 6.Πάρε οδό 7.Ενημέρωσε χρήστη να δώσει νούμερο 8.Πάρε νούμερο 9.Εμφάνισε όλα τα στοιχεία στη σειρά

εκτύπωσε το μήνυμα που ζητάει τον αριθμό των χαρακτήρων δώσε τον αριθμό των χαρακτήρων διάβασε τον πρώτο χαρακτήρα αν ο πρώτος χαρακτήρας είναι το κενό τότε βάλε στην θέση 0 ότι τερματίζει η συνάρτηση μέχρι να φτάσεις τον αριθμό των χαρακτήρων-1 ή να βρεις το κενό , θα παίρνεις τον εκάστοτε χαρακτήρα και θα τον αποθήκευσε στην αντίστοιχη θέση βάλε στην επόμενη θέση τον τερματικό χαρακτήρα

ΕΜΦΑΝΙΣΕ ΜΗΝΥΜΑ ΓΙΑ ΟΣΟ ΕΙΝΑΙ Ο ΑΡΙΘΜΟΣ ΤΩΝ ΧΑΡΑΚΤΗΡΩΝ ΕΠΑΝΕΛΑΒΕ ΠΑΡΕ ΧΑΡΑΚΤΗΡΑ ΟΡΙΣΕ ΤΟ ΕΚΑΣΤΟΤΕ ΣΤΟΙΧΕΙΟ ΤΟΥ ΑΛΦΑΡΙΘΜΗΤΙΚΟΥ ΩΣ ΤΟΝ ΧΑΡΑΚΤΗΡΑ ΠΟΥ ΔΙΑΒΑΣΕΣ ΑΝ Ο ΧΑΡΑΚΤΗΡΑΣ ΕΙΝΑΙ ΤΟ ENTER ΕΠΕΣΤΡΕΨΕ ΟΡΙΣΕ ΤΟ ΕΠΟΜΕΝΟ ΣΤΟΙΧΕΙΟΥ ΤΟΥ ΑΛΦΑΡΙΘΜΗΤΙΚΟΥ ΙΣΟ ΜΕ ΤΟ \0

Κυριε Θραμπουλιδη, δεν μπορεσα να ερθω στο εργαστηριο της 5ης βδομαδας, ομως τα προσπαθηα στο σπιτι μου και σας στελνω feedback απο το σπιτι μου.

Διάβασε αλφαριθμητικό Διάβασε μόνο όσους χαρακτήρες μπορεί να αποθηκεύσει η αντίστοιχη μεταβλητή

Ζήτα Λέξη από τον Χρήστη Σε περίπτωση λάθους εμφάνισε του μήνυμα κινδύνου

1) Τύπωσε το μήνυμα 2) Μέχρι να συμπληρωθεί ο μέγιστος αριθμός χαρακτήρων ή μέχρι να φτάσει χαρακτήρα "\n" επανέλαβε: 2.1) Διάβασε χαρακτήρα 2.2) Αποθήκευσε χαρακτήρα στον πίνακα 2.3) Α χαρακτήρας είναι "\n", ολοκλήρωσε το αλφαριθμητικό 3) Ολοκλήρωσε το αλφαριθμητικό

Παρε τη συμβολοσειρα με οριο χαρακτηρων Αποθηκευσε την με ασφαλη τερματισμο

Τύπωσε ενα μήνυμα. Πάρε χαρακτήρες.

Δώσε οδό Δώσε όνομα Δώσε επίθετο Για κάθε λέξη μικρότερη του μέγιστου αριθμού χαρακτήρων Πάρε λέξη Αποθήκευσε την λέξη

Ζήτα όνομαεπίθετοοδό Αν είναι κενό τερμάτισε Επανάλαβε ζήτα όνομαεπίθετοοδό αποθήκευσε την πληροφορία(όνομα,ιδιεύθυνση,οοδ) διαδοχικά σε μικρά κομμάτια Τερμάτισε

Πάρε 4 χαρακτήρες για την οδό Πάρε 3 χαρακτήρες για το επώνυμο Πάρε 5 χαρακτήρες για το όνομα

Η συνάρτηση getStringV2 λαμβάνει από τον χρήστη μια σειρά αλφαριθμών τα οποία διαβάσει ξεχωριστά το καθένα. Ελέγχει τον αριθμν των αλφάριθμων που δόθηκαν από τον χρήστη σε σχέση με το επιτρεπόμενο αριθμό από το πρόγραμμα. Εξάγει στον χρήστη ή την σειρά αλφαριθμών ή μήνυμα που τον ενημερώνει για την παραβίαση του μέγιστου αριθμού αλφαριθμών εάν έχει παραβιαστεί.

1. Εκτύπωσε μήνυμα 2. Διάβασε τον 1ο χαρακτήρα 3. Έλεγξε αν είναι κενό 4. Διάβασε χαρακτήρες και αποθήκευσε τους σε πίνακα 5. Πρόσθεσε \0 στο τέλος του πίνακα

1.Εκτύπωσε μηνυμα 2.Διαβασε 1ο χαρακτηρα 3.Αν enter τερμάτισε Αλλιως Μεχρι να τελειωσουν οι χαρακτες η να συναντησεις enter -διαβαζε καθε χαρακτηρα -Αποθηκευσε τον σε θεση του string

Λεκτική περιγραφή της getStringV2

Κώδικας που είχαμε δώσει πρόχειρα

```
34 // 1st darft version - to be updated
35 void getStringV2(char message[],char str
36     char ch;
37     int i=0;
38
39     printf(message);
40     ch=getchar();
41     if(ch=='\n'){
42         str[0]='\0';
43         return;
44     }
45     for(i=0;i<numOfChars&&ch!='\n';i++){
46         ch=getchar();
47         str[i]=getchar();
48     }
49     str[i]='\0';
50     return;
51 }
```

Λεκτική Περιγραφή

Εμφάνισε μήνυμα
Για όσους χαρακτήρες ζητείται
Πάρε χαρακτήρα
Αν ο χαρακτήρας είναι Enter
τερμάτισε την επανάληψη
Βάλε τον χαρακτήρα στο αλφαριθμητικό
Βάλε τον χαρακτήρα τερματισμού μετά τον τελευταίο χαρακτήρα
Καθάρισε βασική είσοδο

Μετατρέψτε την Λεκτική Περιγραφή σε πηγαίο κώδικα

Words Handling

word1, word2, ... word5 είναι πέντε μεταβλητές ίδιου τύπου

```
char word1[20];  
char word2[20];  
char word3[20];  
char word4[20];  
char word5[20];
```



char word[5][20];

word[0] ↔ word1

word[1] ↔ word2

word[2] ↔ word3

word[3] ↔ word4

word[4] ↔ word5



```
scanf("%s", word1);  
printf("%s\n", word1);
```

```
scanf("%s", word[0]);  
printf("%s\n", word[0]);
```

```
scanf("%s", word[i]);  
printf("%s\n", word[i]);
```



Βασικά Στοιχεία της C

Το αντικείμενο του μαθήματος δεν είναι η C. Αποφύγετε στην φάση αυτή λεπτομέρειες της C! Ότι χρειάζεται από την C για το μάθημα Εισαγωγή στον Προγραμματισμό θα βρείτε στους παραπλεύρως συνδέσμους από την κατηγορία συνδέσμους από την κατηγορία Βασικά στοιχεία της Γλώσσας C.

Ομάδες Χρηστών

Συζητήσεις

Σύνδεσμοι

Συνομιλία

Κατηγορίες συνδέσμων

Βασικά στοιχεία της Γλώσσας C

Γλώσσα Προγραμματισμού C

Κατηγορίες συνδέσμων

Βασικά στοιχεία της Γλώσσας C

I2P_LanguageOverview.pdf

Συναρτήσεις (I2P_Functions.pdf)

Μεταβλητές και Τύποι Δεδομένων (I2P_Variables&DataTypes.pdf)

Τελεστές και Εκφράσεις (I2P_Expressions&Operators.pdf)

Προτάσεις Ελέγχου Ροής (I2P_ControlFlowStatements.pdf)

Πίνακες (I2P_Arrays.pdf)

Δείκτες (I2P_Pointers.pdf)

I2P_StatementsInBrief.pdf

Δομές (I2P_Struct.pdf)

<https://eclass.upatras.gr/modules/link/?course=CEID1456>

Week 7 - Οργάνωση

[EA-6] Εργαστήριο 7ης εβδομάδας (Αντικείμενο)

- **[EA-5] Σχολιασμός**
- **Παραδοτέο 6ης εβδομάδας**
 - Σχολιασμός Παραδοτέου
- **Προετοιμασία EA-6**
 - **MostFrequentNumber**
 - Divisors
 - AddNumbers

Δράσεις

Η EA-5 αποτελείται από 3 δράσεις που δίνονται παρακάτω.

Σε όλες τις δράσεις θα ακολουθήσετε τα βήματα της Διαδικασίας Ανάπτυξης Προγράμματος (Updated)) αξιοποιώντας ότι υλικό abstraction) και αυξητική ανάπτυξη (incremental development)

[Δ1] – MostFrequentNumber

Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα εμφανίστηκε στην ακολουθία περισσότερες συνεχόμενες φορές

[Δ2] – Divisors

Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα είναι πρώτος θα εμφανίζει τους διαιρέτες του και το πλήθος τ

[Δ3] – AddNumbers

Αναπτύξτε ένα πρόγραμμα με βάση το οποίο η μηχανή θα εκτε [I2P_BasicConceptsPartC.pdf](#).

👍 1 🗨 0

<https://eclass.upatras.gr/modules/blog/index.php?course=CEID1456&action=showPost&pId=1735>

Week 7 - EA-5 Επιλεγμένα θέματα

- **[Δ1] StringHandling**
 - Μετατροπή λεκτικής περιγραφής σε πηγαίο κώδικα
- **[Δ2] MinAndMax**
 - Χρήση της i2p library (sortIntArrayInc) [Η Βιβλιοθήκη i2p](#)
- **[Δ3] MenuV3** [menuV3.pdf](#)
 - Μετατρέψτε τις συναρτήσεις add(), sub(), mul() κ.λ.π. σε επαναχρησιμοποιήσιμες. [Συναρτήσεις \(I2P_Functions.pdf\)](#) [Βασικά στοιχεία της Γλώσσας C](#)
- **[Δ4] WordsHandling** [wordsHandlingV5.pdf](#)
 - Λεκτική περιγραφή των συναρτήσεων wordStats() και charStats()

Παραδοτέο 6ης Εβδομάδας

[EA-5] Εργαστήριο 5ης και 6ης εβδομάδας (Αντικείμενο)

■ Αναφέρεται στην Δράση [Δ4] (WordsHandling) της [EA-5]

Στο Μέρος 1° θα παραδώσετε ένα αρχείο, με όνομα <Ζώνη-Θέση>_SourceCode.pdf.

Το αρχείο θα περιέχει τον πηγαίο κώδικα της τελευταίας λειτουργούσας έκδοσης που αναπτύξατε για τη Δράση [Δ4] - WordsHandling.

Στο Μέρος 2° θα παραδώσετε ένα αρχείο, με όνομα <Ζώνη-Θέση>_VD.pdf, το οποίο θα περιέχει τις λεκτικές περιγραφές για τα παρακάτω:

- 1. Λεκτική περιγραφή της main() της έκδοσης της wordsHandling της οποίας παραδώσατε τον πηγαίο κώδικα.
- 2. Λεκτική περιγραφή της charStats(), ή όπως αλλιώς έχετε ονομάσει την συνάρτηση που υλοποιεί την λειτουργία στατιστικών χαρακτήρα). Η λεκτική περιγραφή θα πρέπει να είναι σύμφωνη με τον πηγαίο

προϋποθέτει την προσεκτική παρακολούθηση διά λέξης και φροντιστηρίου και μελέτη του αντίστοιχου υλικού στο eclass.

Παραδοτέο Εργαστηρίου 6ης εβδομάδας (Μέρος 1ο)	Turnitin	72
Ατομική εργασία		
Ανάθεση σε: Συγκεκριμένες Ομάδες Χρηστών		
Παραδοτέο Εργαστηρίου 6ης εβδομάδας (Μέρος 2ο)	Turnitin	67
Ατομική εργασία		

Ενδεικτικά παραδοτέα – main()

- Παρουσίασε μενού και πάρε επιλογή από χρήστη
- Ανάλογα την επιλογή του χρήστη κάνε αντίστοιχη δράση
- Ζήτη πάλι την επιλογή του χρήστη



Δείξε τον τίτλο προγράμματος
Οσο δεν έχει επιλεχθεί ο τερματισμός του προγράμματος:
Ενάλλαξε μεταξύ των λειτουργιών:

Εισήγαγε λέξεις
Τέλος
Πρόβαλε λέξεις
Τέλος
Ταξινομήσε λέξεις σε αύξουσα σειρά
Τέλος



1. ΕΜΦΑΝΙΣΕ ΣΤΟΝ ΧΡΗΣΤΗ ΤΗΣ ΕΠΙΛΟΓΕΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ
2. ΠΑΡΕ ΑΡΙΘΜΟ ΑΠΟ ΤΟΝ ΧΡΗΣΤΗ ΓΙΑ ΤΗΝ ΕΠΙΛΟΓΗ ΠΡΟΓΡΑΜΜΑΤΟΣ
3. ΑΝΑΖΗΤΗΣΕ ΜΕ ΒΑΣΗ ΤΟΝ ΑΡΙΘΜΟ ΤΟ ΠΡΟΓΡΑΜΜΑ
4. ΑΝ ΔΕΝ ΥΠΑΡΧΕΙ ΕΜΦΑΝΙΣΕ ΑΝΑΛΟΓΟ ΜΗΝΥΜΑ
5. ΕΠΑΝΕΛΑΒΕ ΜΕΧΡΙ ΤΟΝ ΤΕΡΜΑΤΙΣΜΟ ΑΠΟ ΤΟΝ ΧΡΗΣΤΗ
6. ΣΤΟΝ ΤΕΡΜΑΤΙΣΜΟ ΕΜΦΑΝΙΣΕ ΑΝΑΛΟΓΟ ΜΗΝΥΜΑ



Περιγραφή της main()



-Το πρόγραμμα ξεκινά με την εκτύπωση ενός εισαγωγικού μηνύματος (

Α) Λεκτική Περιγραφή της main()



Η συνάρτηση main() είναι η κύρια συνάρτηση τ την εκτέλεση του μενού και την επιλογή των δι της είναι η εξής:



Εμφάνιση Μενού:

Λεκτική Περιγραφή main():

- 1) Όρισε μεταβλητή ch
- 2) Όρισε μεταβλητή max
- 3) Όρισε μεταβλητή min



Όρισε την ακαίρεα μεταβλητή Operation
Τύπωσε "WordHandling"
Τρέξε την συνάρτηση menu



Ενδεικτικά παραδοτέα – charStats()

Ζήτη χαρακτήρα από τον χρήστη

Έλεγε κάθε χαρακτήρα των λέξεων που ο χρήστης έχει προσθέσει αν είναι ίδιος με τον χαρακτήρα που έδωσε

Τύπωσε πόσες φορές εμφανίζεται ο χαρακτήρας συνολικά



Έλεξε σε κάθε λέξη ποσες φορές υπάρχει ο κάθε χαρακτήρας .



Τοποθέτησε τον κάθε αριθμό χαρακτήρων σε έναν πίνακα.

Τύπωσε τον κάθε αριθμό.



ΛΕΚΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΗΣ WORDSSTATS

1. ΑΝ ΔΕΝ ΕΧΕΙ ΔΩΣΕΙ Ο ΧΡΗΣΤΗΣ ΛΕΞΕΙΣ ΕΜΦΑ
2. ΟΡΙΣΕ ΩΣ ΜΕΓΙΣΤΟ,ΕΛΑΧΙΣΤΟ ΚΑΙ ΜΕΣΟ ΜΗΚ
3. ΑΝΑΖΗΤΗΣΕ ΣΤΑΤΙΣΤΙΚΑ ΤΟΥ ΜΗΚΟΥΣ ΤΗΣ ΛΙ

1. Θέσε μεταβλητές « j , biggest , smallest, NumSum ,avg»
2. Θέσε την τιμή biggest ως το μήκος της word[0]
3. Θέσε την τιμή smallest ως το μήκος της word[0]
4. Θέσε την NumSum ως μηδέν
5. Εμφάνισε «word statistics»



6. Επανάλαβε όσο το j είναι μικρότερο του i αυξάνοντας το κατά 1

MostFrequentNumber

Συχνότερα

εμφανιζόμενος αριθμός

- Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα διαβάζει μία ακολουθία από n αριθμούς και θα εμφανίζει:
 - α) τον αριθμό που εμφανίστηκε στην ακολουθία περισσότερες συνεχόμενες φορές, και,
 - β) το πλήθος των συνεχόμενων εμφανίσεων του.

Διαδικασία Ανάπτυξης Προγράμματος (Updated)

- **[Step1-(S1)]** Δώστε την **Αφαιρετική αναπαράσταση** της Διεργασίας.
Η Αφαιρετική αναπαράσταση **καταγράφει τη διεργασία και α) τις εισόδους** που αυτή απαιτεί για την εκτέλεση του έργου **και β) τις εξόδους της**.
 - Τα βήματα της διεργασίας θα περιγράφει η Λεκτική περιγραφή.
- **[S2]** Δώστε τη **λεκτική περιγραφή** της διεργασίας **αξιοποιώντας αφαιρετικότητα στις διεργασίες**.
 - **Ελέγξτε τη Λεκτική περιγραφή**
Ελέγξτε αν εσείς, εκτελώντας τις ενέργειες που η λεκτική περιγραφή ορίζει, πετυχαίνετε τον στόχο της διεργασίας, για παράδειγμα να δημιουργήσετε την ακολουθία Fibonacci.
- **[S3]** Μετασχηματίστε τη λεκτική περιγραφή σε **C κώδικα**.
Δηλώστε τις μεταβλητές που αντιστοιχούν σε εισόδους και εξόδους. Προσδιορίστε ποιες από τις εισόδους είναι μεταβλητές και ποιες σταθερές. Συνεχίστε με τις προτάσεις της Λ.Π.
- **[S4]** Μετασχηματίστε τον πηγαίο κώδικα σε εκτελέσιμο.
- **[S5]** Εκτελέστε και ελέγξτε το πρόγραμμα.

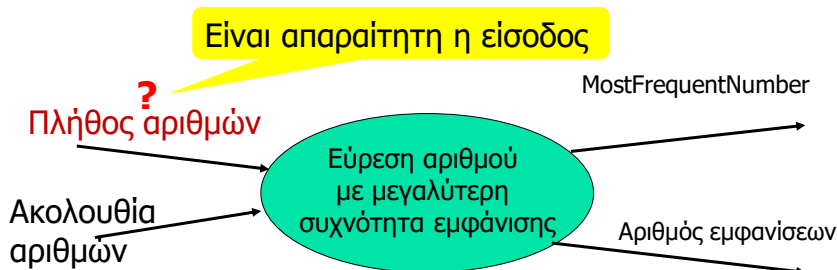
Ο αναλυτικός έλεγχος μπορεί να αναβληθεί εν μέρη για το S5

[I2P_BasicConceptsPartC.pdf](#)

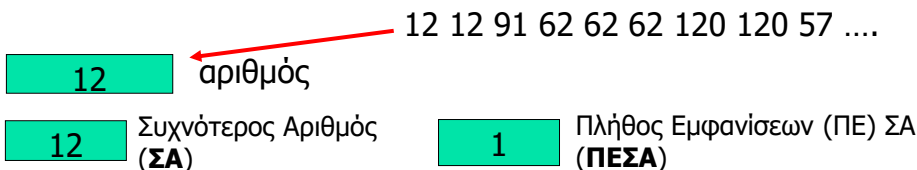
MostFrequentNumber – S1 (Αφαιρετική Αναπαράσταση)

Συχνότερα εμφανιζόμενος αριθμός

- Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα διαβάσει μία ακολουθία από n αριθμούς και θα εμφανίζει:
 - α) τον αριθμό που εμφανίστηκε στην ακολουθία περισσότερες συνεχόμενες φορές, και,
 - β) το πλήθος των συνεχόμενων εμφανίσεων του.



MostFrequentNumber – S2 (Λεκτική Περιγραφή 1/3)



Πάρτε τον πρώτο αριθμό
Θεώρησε τον αριθμό ΣΑ με ΠΕ 1
Για κάθε επόμενο αριθμό της ακολουθίας

- **Αν** αυτός είναι ίσος με τον ΣΑ

- Αύξησε κατά 1 το ΠΕΣΑ

- **Αλλιώς**

- Αντικατέστησε τον ΣΑ με τον αριθμό

Εμφάνισε Συχνότερο Αριθμό και Πλήθος Εμφανίσεων

1η εκδοχή

Αναγνωσιμότητα
(Readability)
Απλότητα
(Simplicity)

Συνεχίστε τον έλεγχο
για διάφορα σενάρια
ακολουθίας

MostFrequentNumber – S2 (Λεκτική Περιγραφή 2/3)

62

αριθμός

12 12 91 62 62 62 120 120 57

62

Υποψήφιος Συχνότερος Αριθμός (ΥΣΑ)

3

Πλήθος Εμφανίσεων ΥΣΑ (ΠΕΥΣΑ)

12

Συχνότερος Αριθμός (ΣΑ)

2

Πλήθος Εμφανίσεων ΣΑ (ΠΕΣΑ)

Πάρε τον πρώτο αριθμό
Θεώρησε τον αριθμό ΥΣΑ με ΠΕ 1
Για κάθε επόμενο αριθμό της ακολουθίας

- **Αν** αυτός είναι ίσος με τον ΥΣΑ
 - Αύξησε κατά 1 το ΠΕΥΣΑ
 - **Αν** ο ΠΕΥΣΑ είναι μεγαλύτερος από το ΠΕΣΑ
 - Αντικατέστησε τον ΣΑ με τον ΥΣΑ
 - Αντικατέστησε το ΠΕΣΑ με ΠΕΥΣΑ
- **Αλλιώς**
 - Αντικατέστησε τον ΥΣΑ με τον αριθμό

Εμφάνισε Συχνότερο Αριθμό και Πλήθος Εμφανίσεων

2^η έκδοσή

Συνεχίστε τον έλεγχο για διάφορα σενάρια ακολουθίας

MostFrequentNumber – S2 (Λεκτική Περιγραφή 3/3)

12 12 91 62 62 62 120 120 57

Πάρε τον πρώτο αριθμό
Θεώρησε τον αριθμό ΥΣΑ με ΠΕ 1
Για κάθε αριθμό της ακολουθίας

- **Αν** αυτός είναι ίσος με τον ΥΣΑ
 - Αύξησε κατά 1 το ΠΕΥΣΑ
 - **Αν** ο ΠΕΥΣΑ είναι μεγαλύτερος από το ΠΕΣΑ
 - Αντικατέστησε τον ΣΑ με τον ΥΣΑ
 - Αντικατέστησε το ΠΕΣΑ με ΠΕΥΣΑ
- **Αλλιώς**
 - Αντικατέστησε τον ΥΣΑ με τον αριθμό

Εμφάνισε ΣΑ και ΠΕΣΑ

Αναγνωσιμότητα (Readability)
Απλότητα (Simplicity)

Συνεχίστε τον έλεγχο για διάφορα σενάρια ακολουθίας.

Divisors	“Αναγνώριση πρώτου αριθμού και παραγοντοποίηση ακεραίου Υπάρχουν ποικίλες μέθοδοι για να προσδιορίσουμε αν ένας αριθμός n είναι πρώτος. Η πιο βασική μέθοδος, η δοκιμαστική διαίρεση, έχει μικρή πρακτική χρησιμότητα επειδή είναι αργή. Ένα τμήμα των σύγχρονων μεθόδων για τον προσδιορισμό αν ένας αριθμός είναι πρώτος είναι εφαρμόσιμο για όλους τους αριθμούς, ενώ οι πιο αποτελεσματικές μέθοδοι είναι διαθέσιμες μόνο για συγκεκριμένες κατηγορίες αριθμών. Οι περισσότερες από αυτές τις μεθόδους λένε μόνο αν ο αριθμός είναι πρώτος ή όχι. Οι μέθοδοι, οι οποίες επιπλέον βρίσκουν και έναν ή περισσότερους παράγοντες του υπό εξέταση αριθμού ονομάζονται αλγόριθμοι παραγοντοποίησης.” wikipedia
Διαιρέτες αριθμού ■ Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα δέχεται ως είσοδο ένα αριθμό και θα ελέγχει αν αυτός είναι πρώτος (prime). Αν ΔΕΝ είναι πρώτος θα εμφανίζει τους διαιρέτες του και το πλήθος τους.	
“Αλγόριθμοι εύρεσης πρώτων Παρατίθενται μερικοί αλγόριθμοι (κατά σειρά ταχύτητας ή και απλότητας) για την εύρεση αν ο $N \geq 2$ είναι πρώτος. Απλός 1 - Εξετάζουμε διαδοχικά όλους τους ακεραίους $M < N$ Απλός 2 - εξετάζοντας όλους τους αριθμούς $M < N/2$ Απλός 3 - εξετάζοντας όλους τους αριθμούς M που είναι μικρότεροι από την τετραγωνική ρίζα του N, Απλός 4 - Εφαρμόζοντας το Θεώρημα του Ουίλσον wikipedia	
Κλεάνθης Θραμπουλίδης	Week Agenda
	45

Divisors - Λεκτική περιγραφή

Διαιρέτες αριθμού ■ Αναπτύξτε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα δέχεται ως είσοδο ένα αριθμό και θα ελέγχει αν αυτός είναι πρώτος (prime). Αν ΔΕΝ είναι πρώτος θα εμφανίζει τους διαιρέτες του και το πλήθος τους.
Λεκτική Περιγραφή – Αφαιρετικότητα στις διεργασίες ■ Πάρε τον αριθμό ■ Αν ο αριθμός είναι Πρώτος ■ Ενημέρωσε τον χρήστη για αυτό ■ Αλλιώς ■ Εμφάνισε διαιρέτες του αριθμού Δώστε στην φάση αυτή μόνο την μετατροπή αυτής της ΛΠ σε πηγαίο κώδικα
Κλεάνθης Θραμπουλίδης
Week Agenda
46

Divisors - Λεκτική περιγραφή -> SC

Λεκτική Περιγραφή – Αφαιρετικότητα στις διεργασίες

Δες ενναλακτική υλοποίηση της getInt στο [I2P_Functions.pdf](#)

- Πάρε τον αριθμό
- Αν ο **αριθμός είναι Πρώτος**
 - Ενημέρωσε τον χρήστη για αυτό
- Αλλιώς
 - **Εμφάνισε διαιρέτες του αριθμού**

```
num=getInt("Dose arithmo:");
if(isPrime(num))
    printf("O arithmos %d einai protos\n",num);
else
    displayDivisors(num);
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdbool.h>
22 bool isPrime(int num){
23     printf("isPrime executed for num =%d\n",num);
24     // return true;
25     return false;
26 }
```

Ορίζει τον τύπο bool

[DivisorsV1.pdf](#)

AddNumbers – Incremental Development

Γράψτε ένα πρόγραμμα σύμφωνα με το οποίο η μηχανή θα υπολογίζει το άθροισμα ακέραιων αριθμών. Προσέξτε τους περιορισμούς!

Περιορισμοί: Υποθέστε πως η μηχανή :

- δεν μπορεί να υπολογίσει άθροισμα αριθμών με περισσότερα από ένα ψηφία. Για τον υπολογισμό του αθροίσματος 2 ψηφίων χρησιμοποιήστε τον τελεστή +.
- μπορεί να κάνει τις πράξεις του πολλαπλασιασμού και διαίρεσης ακέραιων αριθμών με χρήση των τελεστών * και / αντίστοιχα. Επιπλέον μπορεί να βρίσκει το modulo ενός ακέραιου αριθμού με χρήση του τελεστή %.

- 1η έκδοση: Οι αριθμοί είναι 2 με 2 το πολύ ψηφία.
- 2η έκδοση: Οι αριθμοί είναι 2 με 3 το πολύ ψηφία.
- 3η έκδοση: Οι αριθμοί είναι 3 με 3 το πολύ ψηφία.
- 4η έκδοση: Οι αριθμοί είναι k με n το πολύ ψηφία. **Αξιοποιήστε προτάσεις επανάληψης και πίνακες.**

Απαραίτητη η εφαρμογή της Αυξητικής Ανάπτυξης

[I2P_ControlFlowStatements.pdf](#)

[I2P_Arrays.pdf](#)

AddNumbers [S1] Αφαιρετική Αναπαράσταση

AddNumbers

Γράψτε ένα πρόγραμμα σύμφωνα με το οποίο η μηχανή θα υπολογίζει το άθροισμα ακέραιων αριθμών. Προσέξτε τους περιορισμούς!



82
+ 35

AddNumbers [S2] Λεκτική Περιγραφή 1/4

82
+ 35

Προϋποθέτει:

γνώση του αλγορίθμου και δυνατότητα περιγραφής του στην ομιλούμενη γλώσσα

Είναι

η περιγραφή, σε προστακτική μορφή, των ενεργειών που πρέπει να κάνει ο **μαθητής δημοτικού** για να οδηγηθεί στο αποτέλεσμα της **άθροισης δύο διψήφιων αριθμών**.

- Βάλε τον ένα αριθμό κάτω από τον άλλο με στοίχιση στα δεξιά
- Πρόσθεσε το πρώτο δεξιά ψηφίο του 1^{ου} αριθμού με το πρώτο δεξιά του 2^{ου}
- Αν το άθροισμα είναι μικρότερο από 10
 - Γράψε το από κάτω
- Αλλιώς
 - Γράψε από κάτω το δεξιό ψηφίο του αθροίσματος
 - Κράτα ως κρατούμενο το αριστερό ψηφίο του αθροίσματος
- Πρόσθεσε το επόμενο ψηφίο του 1ου αριθμού με το επόμενο του 2ου
- Αν υπάρχει από προηγούμενη πρόσθεση κρατούμενο
 - πρόσθεσε το
- Γράψε το άθροισμα από κάτω

Συνεχίστε τον έλεγχο για διάφορα σενάρια.

AddNumbers [S2] Λεκτική Περιγραφή 2/4

82
+ 35

Πρόσθεση 2 διψήφων αριθμών

- Βάλε τον ένα αριθμό κάτω από τον άλλο με στοίχιση στα δεξιά
- Πρόσθεσε το πρώτο δεξιά ψηφίο του 1^{ου} αριθμού με το πρώτο δεξιά του 2^{ου}
- Αν το άθροισμα είναι μικρότερο από 10
 - Ενημέρωσε το τελικό άθροισμα με το ψηφίο του αθροίσματος
- Αλλιώς
 - Ενημέρωσε το τελικό άθροισμα με το τελευταίο δεξιά ψηφίο του αθροίσματος
 - Κράτα ως κρατούμενο το αριστερό ψηφίο του αθροίσματος
- Πρόσθεσε το επόμενο ψηφίο του 1ου αριθμού με το επόμενο του 2ου
- Αν υπάρχει από προηγούμενη πρόσθεση κρατούμενο
 - πρόσθεσε το
- Ενημέρωσε το τελικό άθροισμα με τα ψηφία του αθροίσματος
- Εμφάνισε το τελικό άθροισμα

1^η εκδοχή

Μετατρέψτε το σε κώδικα και ελέγξτε την ορθότητα του

AddNumbers [S2] Λεκτική Περιγραφή 3/4

382
+ 835

Πρόσθεση 2 τριψήφων αριθμών

- Πρόσθεσε το πρώτο δεξιά ψηφίο του 1^{ου} αριθμού με το πρώτο δεξιά του 2^{ου}
 - Αν το άθροισμα είναι μικρότερο από 9
 - Ενημέρωσε το τελικό άθροισμα με το ψηφίο του αθροίσματος
 - Αλλιώς
 - Ενημέρωσε το τελικό άθροισμα με το τελευταίο δεξιά ψηφίο του αθροίσματος
 - Κράτα ως κρατούμενο το αριστερό ψηφίο του αθροίσματος
- Για κάθε επόμενης τάξης ψηφίο**
- Πρόσθεσε το επόμενο ψηφίο του 1ου αριθμού με το επόμενο του 2ου
 - Αν υπάρχει από προηγούμενη πρόσθεση κρατούμενο
 - πρόσθεσε το
 - Ενημέρωσε το τελικό άθροισμα με τα ψηφία του αθροίσματος
- Εμφάνισε το τελικό άθροισμα**

2^η εκδοχή

Διορθώστε το λάθος. Βελτιώστε τον αλγόριθμο

AddNumbers [S2] Λεκτική Περιγραφή 4/4

Πρόσθεση 3 τριψήφιων αριθμών

382
835
+123

- Θεώρησε τον μεγαλύτερο αριθμό πρώτο.
- για **κάθε ψηφίο του πρώτου αριθμού** ξεκινώντας από το λιγότερο σημαντικό
 - Υπολόγισε το άθροισμα των αντίστοιχης τάξης ψηφίων **όλων των αριθμών**
 - Αν υπάρχει από προηγούμενη πρόσθεση κρατούμενο
 - πρόσθεσε το στο άθροισμα των ψηφίων
 - Αν το άθροισμα των ψηφίων είναι μικρότερο από 9
 - Ενημέρωσε το τελικό άθροισμα με το άθροισμα των ψηφίων
 - Αλλιώς
 - Ενημέρωσε το τελικό άθροισμα με το λιγότερο σημαντικό ψηφίο του αθροίσματος των ψηφίων
 - Ενημέρωσε το κρατούμενο
- Ενημέρωσε το τελικό άθροισμα με βάση το κρατούμενο
- Εμφάνισε το τελικό άθροισμα.

3η έκδοση

Ελέγξτε για διάφορα
σενάρια

WordsHandling – Δείκτες και Δομές (struct)

```
160 void wordStats(void){  
161     int curWord_len,max_len, min_len,total_len=0;  
162     int i;  
163  
164     initializeWordStats(words[0]);  
165     for(i=1;i<freeEntry;i++){  
166         updateWordStats(words[i]);  
167     }  
168     displayWordStats();  
169 }
```

Διαμόρφωσε τις συναρτήσεις που καλεί η wordStats ώστε να λειτουργούν με τις μεταβλητές στατιστικών λέξης ως τοπικές στην wordStats.
Δες τι επιλογές έχεις;

Δοκίμασε με δείκτες ανάλογα με την getIntV2

How I feel when my code works



- Απαραίτητη Προϋπόθεση**
- A) **Κατανόηση του προς ανάθεση έργου** (είσοδοι, έξοδοι, ...)
 - B) **γνώση του αλγορίθμου,**
 - Γ) **δυνατότητα περιγραφής του στην ομιλούμενη γλώσσα**
 - Δ) δυνατότητα μετατροπής της λεκτικής περιγραφής σε source code
 - E) παραγωγή εκτελέσιμου κώδικα

Week 8 - Οργάνωση

■ Μετατροπή Λεκτικής περιγραφής σε Κώδικα

- `getCharV2` [EA-5]
- `AddNumbers` [EA-6]

■ Πέρασμα Ορισμάτων

- `wordsHandling` (`wordStats`) [EA-5]
- `swap()`
- `getExpression()` [EA-7]

■ [EA-7] Πράξεις με κλάσματα

- **Δείκτες και Δομές**

Δράσεις

Η EA-6 αποτελείται από 3 δι

[EA-6] Εργαστήριο 7ης εβδομάδας (Αντικείμενο)

Σε όλες τις δράσεις θα ακολουθησετε τα μέρη της διαδικασίας ανάπτυξης προγράμματος ([i2P_BasicConcepts](#) Ανάπτυξης Προγράμματος (Updated)) αξιοποιώντας ότι υλικό έχετε στη διάθεσή σας και οπωσδήποτε αφαιρέτ abstraction) και αυξητική ανάπτυξη (incremental development).

[Δ1] - `MostFrequentNumber`

η ανάπτυξη ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα διαβάζει μία ακολουθία από η αριθμούς και εμφανίστηκε στην ακολουθία περισσότερες συνεχόμενες φορές και β) το πλήθος των συνεχόμενων εμφανίσεω

[Δ2] - `Divisors`

Αναπτύξετε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα δέχεται ως είσοδο ένα αριθμό και θα ελέγχει ε είναι πρώτος θα εμφανίζει τους διαιρέτες του και το πλήθος τους.

[Δ3] - `AddNumbers`

Δράσεις

Η EA-7 αποτελείται α

Σε όλες τις δράσ

([i2P_BasicConcepts](#) αξιοποιώντας ότι υλ abstraction) και αυξητική ανάπτυξη (incremental development).

[EA-7] Εργαστήριο 8ης εβδομάδας (Αντικείμενο)

[Δ1] - `Ολοκλήρωση των δράσεων της EA-6`

Αν δεν το έχετε κάνει ήδη θα ολοκληρώσετε τις δράσεις της EA-6.

[Δ2] - `FractionsV1` (Πράξεις με κλάσματα αξιοποιώντας δείκτες)

Αναπτύξετε ένα πρόγραμμα για υπολογισμό τιμής εκφράσεων με κλάσματα όπως περ [Άσκηση Fractions](#). Στην δράση αυτή θα αξιοποιήσετε την πρώτη έκδοση της `getExpress getExpressionV1()`.

Η άσκηση σας δίνει τις απαραίτητες οδηγίες για την εκτέλεση της. Πιο αναλυτικές οδηγίες της συνάρτησης θα βρείτε στο [i2PLibrary.pdf](#).

[Δ3] - `FractionsV2` (Πράξεις με κλάσματα αξιοποιώντας δομές)

Παραδοτέο 7ης Εβδομάδας

■ Αναφέρεται στην Δράση [Δ3] (AddNumbers) της [ΕΑ-6]

Θα παραδώσετε σε μορφή pdf τον πηγαίο κώδικα της τελευταίας (από τις 2-5 που δουλέψαμε στην διάλεξη) λειτουργούσας έκδοσης σας του προγράμματος addNumbers.

Στην αρχή του πηγαίου κώδικα θα αναγράφεται η έκδοση που ο κώδικας σας υλοποιεί

Μετατροπή Λεκτικής περιγραφής σε Κώδικα

Παραδοτέο Εργαστηρίου 7ης εβδομάδας (labTimeGroup1)

39

Παραδοτέο Εργαστηρίου 7ης εβδομάδας (labTimeGroup2)

44

Κλεάνθης Θραμπουλίδης

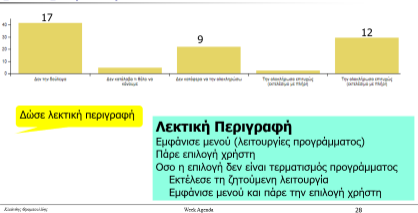
Week Agenda

57

Λεκτική Περιγραφή -> Κώδικας

Μετατρέψτε τη Λεκτική Περιγραφή σε πηγαίο κώδικα

[ΕΑ-5] Δράση Δ3 – Menu



Δώσε λεκτική περιγραφή

Λεκτική Περιγραφή

Εμφάνισε μενού (λειτουργίες προγράμματος)
Πάρε επιλογή χρήστη
Όσο η επιλογή δεν είναι τερματισμός προγράμματος
Εκτέλεσε τη ζητούμενη λειτουργία
Εμφάνισε μενού και πάρε την επιλογή χρήστη

Λεκτική περιγραφή της getStringV2

Κώδικας που είχαμε δώσει πρόχειρα

```
34 // let user's version - to be updated
35 void getString(char message[],char str)
36 {
37     int i;
38     printf(message);
39     char ch;
40     do{
41         ch=getchar();
42         if(ch=='\n')
43             return;
44     }while(ch!='\n');
45     for(i=0;i<maxChars-1;i++){
46         ch=getchar();
47         str[i]=ch;
48     }
49     str[i]='\0';
50     return;
51 }
```

Λεκτική Περιγραφή

Εμφάνισε μήνυμα
Για όσους χαρακτήρες ζητείται
Πάρε χαρακτήρα
Αν ο χαρακτήρας είναι Enter
τερματίσε την επανάληψη
Βάλε τον χαρακτήρα στο αλφαριθμητικό
Βάλε τον χαρακτήρα τερματισμού μετά τον
τελευταίο χαρακτήρα
Καθάρισε βασική είσοδο

Μετατρέψτε την Λεκτική Περιγραφή σε πηγαίο κώδικα

```
22 //V1
23 int main(int argc, char *argv[]) {
24     selectedOp=getUserChoice();
25     while(selectedOp!=TERMINATE){
26         executeOperation(selectedOp);
27         selectedOp=getUserChoice();
28     }
29     return 0;
30 }
31 //V2
32 int main(int argc, char *argv[]) {
33     while((selectedOp=getUserChoice())!=TERMINATE)
34         executeOperation(selectedOp);
35     return 0;
36 }
37
38 void getStringV2(char message[],char str[],int maxChars){
39     char ch;
40     int i;
41     printf("%s", message);
42     for(i=0;i<maxChars-1;i++){
43         ch=getchar();
44         if(ch=='\n')
45             break;
46         str[i]=ch;
47     }
48     str[i]='\0';
49     fflush(stdin);
50 }
```

Κλεάνθης Θραμπουλίδης

Week Agenda

58

AddNumbers: Λεκτική Περιγραφή -> Κώδικας

AddNumbers [S2] Λεκτική Περιγραφή 2/4

82
+ 35

Πρόσθεση 2 διψήφων αριθμών

- Βάλε τον ένα αριθμό κάτω από τον άλλο με στοίχιση στα δεξιά
- Πρόσθεσε το πρώτο δεξιά ψηφίο του 1^{ου} αριθμού με το πρώτο δεξιά του 2^{ου}
- Αν το άθροισμα είναι μικρότερο από 10
 - Ενημέρωσε το τελικό άθροισμα με το ψηφίο του αθροίσματος
- Αλλιώς
 - Ενημέρωσε το τελικό άθροισμα με το τελευταίο δεξιά ψηφίο
 - Κράτα ως κρατούμενο το αριστερό ψηφίο του αθροίσματος
- Πρόσθεσε το επόμενο ψηφίο του 1ου αριθμού με το επόμενο του 2ου
- Αν υπάρχει από προηγούμενη πρόσθεση κρατούμενο
 - πρόσθεσε το
- Ενημέρωσε το τελικό άθροισμα με τα ψηφία του αθροίσματος
- Εμφάνισε το τελικό άθροισμα

```
132 digitSum= digitAt(num1,1) + digitAt(num2,1);
133 if(digitSum<10)
134     sum=digitSum;
135 else{
136     sum=digitAt(digitSum,1);
137     cary=digitAt(digitSum,2);
138 }
139 digitSum= digitAt(num1,2) + digitAt(num2,2);
140 if(cary!=0)
141     digitSum+=cary;
142 sum=sum+digitSum*10;
143 printf("Sum=%d\n",sum);
```

Κλεάνθης Θραμπουζίδης

Week Agenda

59

AddNumbers: Απλοποίηση του κώδικα

```
132 digitSum= digitAt(num1,1) + digitAt(num2,1);
133 if(digitSum<10)
134     sum=digitSum;
135 else{
136     sum=digitAt(digitSum,1);
137     cary=digitAt(digitSum,2);
138 }
139 digitSum= digitAt(num1,2) + digitAt(num2,2);
140 if(cary!=0)
141     digitSum+=cary;
142 sum=sum+digitSum*10;
143 printf("Sum=%d\n",sum);
```

AddNumbers [S2] Λεκτική Περιγραφή 2/4

82
+ 35

Πρόσθεση 2 διψήφων αριθμών

- Βάλε τον ένα αριθμό κάτω από τον άλλο με στοίχιση στα δεξιά
- Πρόσθεσε το πρώτο δεξιά ψηφίο του 1^{ου} αριθμού με το πρώτο δεξιά του 2^{ου}
- Αν το άθροισμα είναι μικρότερο από 10
 - Ενημέρωσε το τελικό άθροισμα με το ψηφίο του αθροίσματος
- Αλλιώς
 - Ενημέρωσε το τελικό άθροισμα με το τελευταίο δεξιά ψηφίο του αθροίσματος
 - Κράτα ως κρατούμενο το αριστερό ψηφίο του αθροίσματος
- Πρόσθεσε το επόμενο ψηφίο του 1ου αριθμού με το επόμενο του 2ου
- Αν υπάρχει από προηγούμενη πρόσθεση κρατούμενο
 - πρόσθεσε το
- Ενημέρωσε το τελικό άθροισμα με τα ψηφία του αθροίσματος
- Εμφάνισε το τελικό άθροισμα

1^ο ερώτη

Μετατρέψτε το σε κώδικα και ελέγξτε την ορθότητα του

```
digitSum= digitAt(num1,1) + digitAt(num2,1)+cary;
sum+=digitAt(digitSum,1);
cary=digitAt(digitSum,2);

digitSum= digitAt(num1,2) + digitAt(num2,2)+cary;
sum+=digitAt(digitSum,1)*pow(10,1);
cary=digitAt(digitSum,2);

sum+=cary*pow(10,2);
printf("Sum=%d\n",sum);
```

Κλεάνθης Θραμπουζίδης

Week Agenda

60

AddNumbers: Για 2 αριθμούς με τρία ψηφία

382
+ 835

```
109 digitSum= digitAt(num1,1) + digitAt(num2,1)+cary;
110 sum+=digitAt(digitSum,1);
111 cary=digitAt(digitSum,2);
112
113 digitSum= digitAt(num1,2) + digitAt(num2,2)+cary;
114 sum+=digitAt(digitSum,1)*pow(10,1);
115 cary=digitAt(digitSum,2);
116
117 sum+=cary*pow(10,2);
118 printf("Sum=%d\n",sum);
```

1^ο ψηφίο

2^ο ψηφίο

Κώδικας για 3^ο ψηφίο

Τροποποιήστε τον κώδικα αξιοποιώντας πρόταση επανάληψης για τον αριθμό των ψηφίων

```
81 digitSum= digitAt(num1,1) + digitAt(num2,1)+cary;
82 sum+=digitAt(digitSum,1);
83 cary=digitAt(digitSum,2);
84
85 digitSum= digitAt(num1,2) + digitAt(num2,2)+cary;
86 sum+=digitAt(digitSum,1)*pow(10,1);
87 cary=digitAt(digitSum,2);
88
89 digitSum= digitAt(num1,3) + digitAt(num2,3)+cary;
90 sum+=digitAt(digitSum,1)*pow(10,2);
91 cary=digitAt(digitSum,2);
92
93 sum+=cary*pow(10,3);
94 printf("Sum=%d\n",sum);
```

Τροποποίηση για 3^ο ψηφίο

Κλεάνθης Θραμπουζίδης

Week Agenda

61

AddNumbers: Για 2 αριθμούς με τρία ψηφία (loop)

```
81 digitSum= digitAt(num1,1) + digitAt(num2,1)+cary;
82 sum+=digitAt(digitSum,1);
83 cary=digitAt(digitSum,2);
84
85 digitSum= digitAt(num1,2) + digitAt(num2,2)+cary;
86 sum+=digitAt(digitSum,1)*pow(10,1);
87 cary=digitAt(digitSum,2);
88
89 digitSum= digitAt(num1,3) + digitAt(num2,3)+cary;
90 sum+=digitAt(digitSum,1)*pow(10,2);
91 cary=digitAt(digitSum,2);
92
93 sum+=cary*pow(10,3);
94 printf("Sum=%d\n",sum);
```

Αναβαθμίστε τον αλγόριθμο για περισσότερους από 2 αριθμούς.

Αξιοποιείστε πρόταση επανάληψης για το πλήθος των αριθμών ή κατάλληλη συνάρτηση.

```
81 for(dOrder=1;dOrder<=NUM_OF_DIGITS;dOrder++){
82     digitSum= digitAt(num1,dOrder) + digitAt(num2,dOrder)+cary;
83     sum+=digitAt(digitSum,1)*pow(10,dOrder-1);
84     cary=digitAt(digitSum,2);
85 }
86 sum+=cary*pow(10,dOrder-1);
87 printf("Sum=%d\n",sum);
```

Κλεάνθης Θραμπουζίδης

Week Agenda

62

Δείκτες (pointers) και Δομές (struct)

Δες για τα Βασικά σε Δείκτες και Δομές : [I2P_Pointers.pdf](#), [I2P_Struct.pdf](#)

```
6  int getInt(char message[]);
7  void getIntV2(char message[],int *numPtr);

28 //void getIntV2(char message[],int *numPtr){
29 void getIntV2(char *message,int *numPtr){
30     // int num;
31     printf(message);
32     // scanf("%d",&num);
33     scanf("%d",numPtr);
34     // return num;
35 }
```

Δες άσκηση (week7)
[DivisorsV1.pdf](#)

Δες (week7)
[wordsHandlingV5.pdf](#)

WordsHandling – Δείκτες και Δομές (struct)

```
160 void wordStats(void){
161     int curWord_len,max_len, min_len,total_len=0;
162     int i;
163
164     initializeWordStats(words[0]);
165     for(i=1;i<freeEntry;i++){
166         updateWordStats(words[i]);
167     }
168     displayWordStats();
169 }
```

Διαμόρφωσε τις συναρτήσεις που καλεί η wordStats ώστε να λειτουργούν με τις μεταβλητές στατιστικών λέξης ως τοπικές στην wordStats.
Δες τι επιλογές έχει;

Δοκίμασε με δείκτες ανάλογα με την getIntV2

WordStats με δείκτες

```
153 void wordStats(int freeEntry){
154     int max_len, min_len,total_len=0;
155     int i;
156     initializeWordStats(words[0],&max_len,&min_len,&total_len);
157     for(i=1;i<freeEntry;i++)
158         updateWordStats(words[i],&max_len,&min_len,&total_len);
159     displayWordStats(max_len,min_len,total_len,freeEntry);
160 }

163 void initializeWordStats(char str[],int *maxLen,int *minLen,int *tLen){
164     int curWord_len=strlen(str);
165     *maxLen = *minLen = curWord_len;
166     *tLen += curWord_len;
167 }

169 void updateWordStats(char str[],int *maxLen,int *minLen,int *tLen){
170     int curWordLen=strlen(str);
171     *tLen += curWordLen;
172     if(*maxLen<curWordLen)
173         *maxLen = curWordLen;
174     if(*minLen>curWordLen)
175         *minLen = curWordLen;
176 }

185 void displayWordStats(int maxLen,int minLen,int tLen,int numOfEntries){
186     printf("max len = %d\n",maxLen);
187     printf("min len = %d\n",minLen);
188     printf("average len = %f\n",freeEntry==0?0:(float)tLen/numOfEntries);
189 }
```

Για reusable wordStats

WordStats με Δομές

```
143 void wordStats(int numOfWeeks){
144     WordStatistics wStats;
145     wStats=initializeWordStats(words[0]);
146     for(int i=1;i<numOfWeeks;i++){
147         wStats=updateWordStats(words[i],wStats);
148     }
149     displayWordStats(wStats,numOfWeeks);
150 }

165 WordStatistics updateWordStats(char str[],WordStatistics ws){
166     int curWord_len=strlen(str);
167     ws.total_len += curWord_len;
168     ws.max_len = ws.max_len<curWord_len? curWord_len:ws.max_len;
169     ws.min_len = ws.min_len>curWord_len? curWord_len:ws.min_len;
170     return ws;
171 }

159 void displayWordStats(WordStatistics ws,int numOfWeeks){
160     printf("max len = %d\n",ws.max_len);
161     printf("min len = %d\n",ws.min_len);
162     printf("average len = %f\n",freeEntry==0?0:(float)ws.total_len/numOfWeeks);
163 }
```

```
134 typedef struct wordStatistics{
135     int max_len, min_len;
136     int total_len;
137 }WordStatistics;

151 WordStatistics initializeWordStats(char str[]){
152     WordStatistics ws;
153     int curWordLen=strlen(str);
154     ws.max_len = ws.min_len = curWordLen;
155     ws.total_len = curWordLen;
156     return ws;
157 }
```

Κλεάνθης Θραμπουλίδης

Week Agenda

Week 10 - Οργάνωση

■ Week9 LabTest

- Υπολογισμός κόστους παραγγελίας
- Παιχνίδια με Αριθμούς

■ [EA-8]

- [Δ2] – Orders
- [Δ3] – FourDigitNumbers

■ [EA-9] OrdersHandling

- Ανάπτυξη εφαρμογής Διαχείρισης Παραγγελιών
- Αρχεία

[Δ1] - Διαχείριση Παραγγελιών (OrdersHandling)

Η Δράση έχει στόχο την ανάπτυξη μιας εφαρμογής που θα επιτρέπει στον χρήστη να διαχειρίζεται παραγγελίες σε ένα εργοστάσιο που παράγει φιάλες εμφιάλωσης.

Αναλυτική περιγραφή και Οδηγίες για την ανάπτυξη του προγράμματος θα βρείτε στο αρχείο *OrdersHandling*.

[EA-9] Εργαστήριο 10ης εβδομάδας (Αντικείμενο)

σήμερα - 12:30 μ.μ. - από τον χρήστη Θραμπουλίδης Κλεάνθης

Δράσεις

Η ΕΑ-9 αποτελείται από την ανάπτυξη μιας εφαρμογής για διαχείριση παραγγελιών.

[Δ1] - Διαχείριση Παραγγελιών (OrdersHandlingApp)

Η Δράση έχει στόχο την ανάπτυξη μιας εφαρμογής που θα επιτρέπει στον χρήστη να διαχειρίζεται παραγγελίες σε ένα εργοστάσιο που παράγει φιάλες εμφιάλωσης.

Αναλυτική περιγραφή και Οδηγίες για την ανάπτυξη του προγράμματος *OrdersHandlingApp* θα βρείτε στο αρχείο [OrdersHandlingApp.pdf](#).

Κλεάνθης Θραμπουλίδης

Week Agenda

66

Υπολογισμός κόστους παραγγελίας

Ένα εργοστάσιο παρασκευάζει μικρά και μεγάλα μπουκάλια εμφιάλωσης. Τα μικρά κοστίζουν 0.08€ και τα μεγάλα 0.2€.

Για παραγγελίες μεγαλύτερες από 300€ ή 1000 μπουκάλια, γίνεται έκπτωση 12%.

Για παραγγελίες μεγαλύτερες από 600€, γίνεται έκπτωση 24%.

Λεκτική Περιγραφή

Πάρε παραγγελία

Υπολόγισε αρχικό κόστος παραγγελίας

Υπολόγισε την έκπτωση

Υπολόγισε τελικό κόστος

Εμφάνισε τελικό κόστος

Structured English

Get Order

Calculate Initial Cost

Calculate Discount

Calculate FinalCost()

Display Cost()

“**Structured English** is the use of the [English language](#) with the [syntax](#) of [structured programming](#) to communicate the design of a computer program to non-technical users by breaking it down into logical steps using straightforward English words. Structured English gives aims **to get the benefits of both the programming logic and natural language**: program logic helps to attain precision, whilst natural language helps with the familiarity of the spoken word.” wikipedia

Κλεάνθης Θραμπουλίδης

Week Agenda

67

Μετατροπή Λεκτικής περιγραφής σε C

■ Μεταβλητές (αναγνωσιμότητα)

Μικρά μπουκάλια	sb	smallB	mikMp	a, k, i, ...
Μεγάλα μπουκάλια	bb	bigB	megMp	b, l, j, ...

■ Συναρτήσεις (αναγνωσιμότητα)

- getOrder()
- calcInitialCost(), **initialCost()**, ...
- calcDiscount(), **discount()**, ...
- calcFinalCost(), **finalCost()**, ...

■ Προτάσεις ελέγχου ροής (σωστή στοίχιση κώδικα)

```
4 typedef struct{
5     int sb,bb;
6     float initialCost;
7     float discount;
8 }Order;
11 Order getOrderV1(Order order);
12 float initialCost(Order order);
13 float discount(Order order);
14 void displayOrder(Order order);
14 int main(int argc, char *argv[]) {
15     Order order;
16     float finalCost;
17     order=getOrder(order);
18     order.initialCost=initialCost(order);
19     finalCost=order.initialCost-discount(order);
20     printf("Initial cost=%f\nfinal cost=%f\n",in
21     return 0;
22 }
```

Κλεάνθης Θραμπουλίδης

Week Agenda

68

68

getOrder() – Εναλλακτικές υλοποιήσεις

1. `Order getOrderV1(Order o);`
2. `Order getOrderV2(void);`
3. `void getOrderV3(Order *orderPtr);`
4. `void getOrderV4(Order o);` ?

```
20 int main(int argc, char *argv[]) {
21     Order order={0,0};
22
23     // order=getOrderV1(order);
24     // order=getOrderV2();
25     getOrderV3(&order);
26     // getOrderV4(order);
27     order.initialCost=initialCost(order);
28     displayOrder(order);
29     return 0;
30 }
```

```
26 Order getOrderV1(Order o){
27     o.sb = getInt("Number of small bottles:");
28     o.bb= getInt("Number of big bottles:");
29     return o;
30 }
```

```
59 void getOrderV3(Order *orderPtr){
60     orderPtr->sb = getInt("Number of small bottles:");
61     orderPtr->bb = getInt("Number of big bottles:");
62 }
```

```
52 Order getOrderV2(void){
53     Order o;
54     o.sb = getInt("Number of small bottles:");
55     o.bb = getInt("Number of big bottles:");
56     return o;
57 }
```

```
64 void getOrderV4(Order o){
65     o.sb = getInt("Number of small bottles:");
66     o.bb= getInt("Number of big bottles:");
67 }
```

Κλεάνθης Θραμπουλίδης

Week Agenda

Armstrong Numbers

Problem Statement

<https://pages.mtu.edu/~shene/COURSES/cs201/NOTES/chap04/arms.html>

An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since $3^3 + 7^3 + 1^3 = 371$.

Write a program to find all Armstrong number in the range of 0 and 999.

Λεκτική περιγραφή

- Για κάθε αριθμό από 1000 μέχρι και 9999 επανέλαβε
 - Αν ο αριθμός είναι ίσος με το άθροισμα των δυνάμεων εις την 4^η των ψηφίων του
 - Εμφάνισε τον αριθμό

```
11 for(num=1000;num<=9999;num++){
12     if(num==sumOfdigitsTo4(num))
13         printf("%d\n",num);
14 }
```

Κλεάνθης Θραμπουλίδης

Week Agenda

[EA-8] Δ3 - FourDigitNumbers

1. Αναθέστε στον υπολογιστή το έργο της εύρεσης των τετραψήφιων αριθμών για τους οποίους το άθροισμα της τέταρτης δύναμης των ψηφίων του είναι ίσο με τον αριθμό. Ένας τέτοιος αριθμός είναι ο 8208 καθώς $8208 = 8^4 + 2^4 + 0^4 + 8^4$.
2. Δώστε τη λεκτική περιγραφή με βάση την οποία δομήσατε την main(). Τη λεκτική περιγραφή θα τοποθετήσετε στην αρχή ακριβώς πάνω από την main().
3. Δείτε την ανάρτηση **FourDigitNumbers και ChatGPT**
4. Τώρα που έχετε την δική σας λειτουργούσα έκδοση, κάντε προσπάθεια να δώσετε μια νέα έκδοση η οποία να βελτιώνει για την μηχανή τον χρόνο εκτέλεσης του έργου.
5. Δώστε εναλλακτικές υλοποιήσεις.

3. FourDigitNumbers και ChatGPT

4. κάντε προσπάθεια να δώσετε μια νέα έκδοση η οποία να **βελτιώνει** για την μηχανή **τον χρόνο εκτέλεσης** του έργου.

5. Δώστε εναλλακτικές υλοποιήσεις.

Κλεάνθης Θραγουλίδης

Week Agenda

Διορθώνοντας το ChatGPT

```
#include <stdio.h>
#include <math.h>

int main() {
    printf("4-digit Armstrong numbers:\n");

    for (int num = 1000; num <= 9999; ++num) {
        int originalNum = num;
        int sum = 0;

        while (num > 0) {
            int digit = num % 10;
            sum += pow(digit, 4);
            num /= 10;
        }

        if (sum == originalNum) {
            printf("%d\n", originalNum);
        }
    }

    return 0;
}
```

```
5 int main() {
6     printf("4-digit Armstrong numbers:\n");
7
8     for (int num = 1000; num <= 9999; ++num) {
9         int originalNum = num;
10        int sum = 0;
11        printf("%dst iteration-Start: num:%d\torNum:%d\tsum:%d\n",
12               num-999, num, originalNum, sum);
13        while (num > 0) {
14            int digit = num % 10;
15            sum += pow(digit, 4);
16            num /= 10;
17        }
18        if (sum == originalNum) {
19            printf("%d\n", originalNum);
20        }
21        printf("%dst iteration-End: num:%d\torNum:%d\tsum:%d\n",
22               num-999, num, originalNum, sum);
23        system("pause");
24    }
25    return 0;
26 }
```

1. Παρακολουθείστε την εξέλιξη των τιμών των μεταβλητών σε κάθε επανάληψη για να εντοπίσετε το λάθος.

2. Το ίδιο μπορεί να γίνει αν χρειάζεται και μέσα στο while loop

Κλεάνθης Θραγουλίδης

Week Agenda

Ενναλακτικές Υλοποιήσεις 1/2

- Ξεκινώντας από τον αριθμό

num -> d4, d3, d2, d1

Εναλλακτικές Υλοποιήσεις ?

```
73 int sumOfDigitsTo4(num){  
74     int sum=0;  
75     while (num > 0) {  
76         int digit = num % 10;  
77         sum += pow(digit, 4);  
78         num /= 10;  
79     }  
80     return sum;  
81 }
```

```
22 int sumOfDigitsTo4(int num){  
23     int sum=0;  
24     char str[12];  
25     int digit;  
26     itoa(num, str, 10);  
27     for(int i=0; i<4; i++){  
28         digit= str[i] - '0';  
29         sum += digit4Pow[digit];  
30     }  
31     return sum;  
32 }
```

Κλεάνθης Θραμπουλίδης

Week Agenda

Ενναλακτικές Υλοποιήσεις 2/2

- Ξεκινώντας από τα ψηφία

d4, d3, d2, d1 -> num

```
: main() {  
    int num=0;  
    printf("4-digit Armstrong numbers:\n");  
    for(int dgt4=1; dgt4<10; dgt4++){  
        for(int dgt3=0; dgt3<10; dgt3++){  
            for(int dgt2=0; dgt2<10; dgt2++){  
                for(int dgt1=0; dgt1<10; dgt1++){  
                    num=dgt4*1000+dgt3*100+dgt2*10+dgt1;  
                    if (num==sumOfDigitsTo4(num))  
                        printf("%d\n", num);  
                }  
            }  
        }  
    }  
    for (int num = 1000; num <= 9999; ++num)  
        if (num==sumOfDigitsTo4(num))  
            printf("%d\n", num);  
    return 0;  
}
```

Κλεάνθης Θραμπουλίδης

Week Agenda

Βελτίωση χρόνου εκτέλεσης

- Ξεκινώντας από τον αριθμό

num -> d4, d3, d2, d1

int pow4[10]; ?

- Ξεκινώντας από τα ψηφία

d4, d3, d2, d1 -> num

int pow4[10];

num = d1*1000+d2*100+d3*10+d4;

int digitContr[10][3]; ?

```
22 int sumOfDigitsTo4(int num){
23     int sum=0;
24     char str[12];
25     int digit;
26     itoa(num, str, 10);
27     for(int i=0; i<4; i++){
28         digit= str[i] - '0';
29         sum += digit4Pow[digit];
30     }
31     return sum;
32 }
```

OrdersHandlingApp

[EA-9] Εργαστήριο 10ης εβδομάδας (Αντικείμενο)

σήμερα - 12:30 μ.μ. - από τον χρήστη Θραμπουλίδης Κλεάνθης

Δράσεις

Η EA-9 αποτελείται από την ανάπτυξη μιας εφαρμογής για διαχείριση παραγγελιών.

[Δ1] - Διαχείριση Παραγγελιών (OrdersHandlingApp)

Η Δράση έχει στόχο την ανάπτυξη μιας εφαρμογής που θα επιτρέπει στον χρήστη να διαχειρίζεται παραγγελίες σε ένα εργοστάσιο που παράγει φιάλες εμφιάλωσης.

Αναλυτική περιγραφή και Οδηγίες για την ανάπτυξη του προγράμματος OrdersHandlingApp θα βρείτε στο αρχείο [OrdersHandlingApp.pdf](#).

[OrdersHandlingApp.pdf](#)

Η Παραγγελία (order)

Θεωρήστε μια παραγγελία να αποτελείται, εκτός από τον αριθμό των μικρών και τον αριθμό των μεγάλων μπουκαλιών, από τα παρακάτω τουλάχιστο στοιχεία:

- 1) Ονοματεπώνυμο πελάτη,
- 2) Ημερομηνία που δόθηκε η παραγγελία,
- 3) Ημερομηνία που πρέπει να εκτελεστεί,
- 4) Αρχικό κόστος παραγγελίας (κόστος χωρίς την έκπτωση),
- 5) Έκπτωση που αντιστοιχεί στην παραγγελία (σύμφωνα με την πολιτική εκπτώσεων),
- 6) Τελικό κόστος παραγγελίας (όπου έχει υπολογιστεί και η έκπτωση), και
- 7) Ημερομηνία που τελικά εκτελέστηκε η παραγγελία.

Η λειτουργικότητα που υλοποιεί η εφαρμογή

Η εφαρμογή θα πρέπει να υποστηρίζει τις παρακάτω λειτουργίες που κρίνονται απαραίτητες για την διαχείριση των παραγγελιών. Τις λειτουργίες αυτές θα μπορεί να επιλέξει ο χρήστης της εφαρμογής μέσα από ένα μενού επιλογών. Οι βασικές λειτουργίες περιλαμβάνουν τις παρακάτω:

0 – Τερματισμός Προγράμματος

1 - Καταχώρηση παραγγελίας

2- Εμφάνιση παραγγελιών πελάτη

A. Πρώτη ομάδα (εκκρεμείς παραγγελίες - pendingOrders):

3 - Εμφάνιση εκκρεμών παραγγελιών

4 - Αποθήκευση εκκρεμών παραγγελιών σε αρχείο

5 - Φόρτωση εκκρεμών παραγγελιών από αρχείο.

6 - Εκτέλεση παραγγελίας (υπολογισμός αρχικού κόστους παραγγελίας)

B. Δεύτερη ομάδα (παραγγελίες προς εξόφληση - readyOrders).

7 - Εμφάνιση των προς εξόφληση παραγγελιών

8 - Εξόφληση παραγγελίας (υπολογισμός τελικού κόστους παραγγελίας)

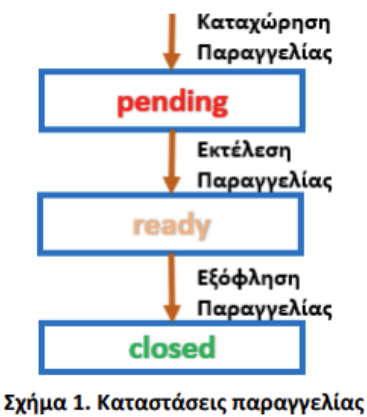
Γ. Τρίτη ομάδα (Εκτελεσμένες παραγγελίες- closedOrders).

9 - Εμφάνιση εκτελεσμένων παραγγελιών

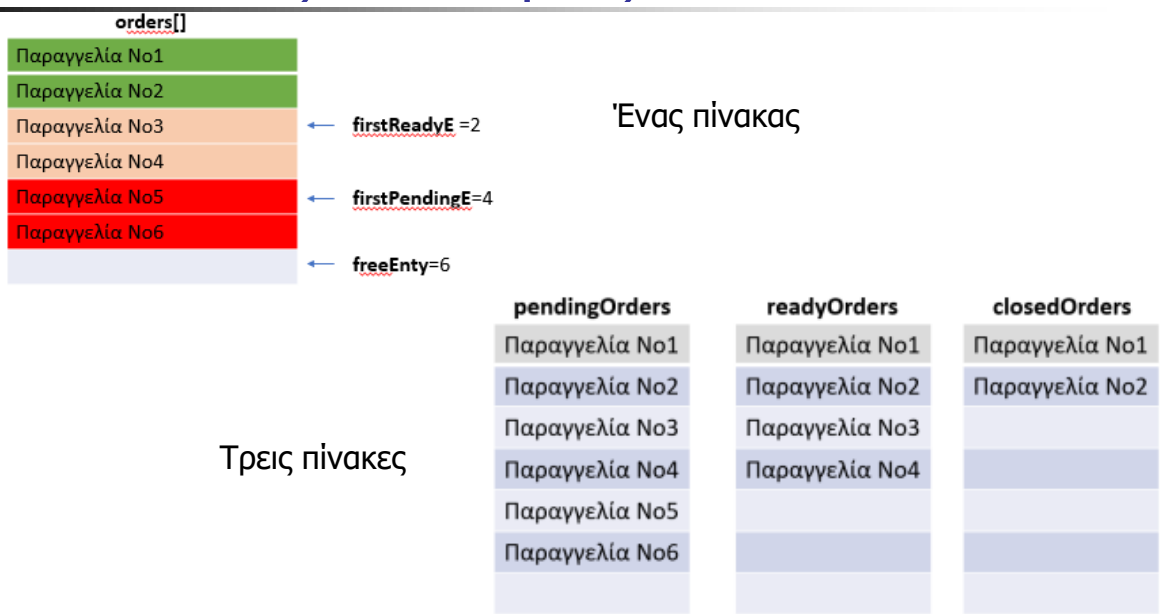
Η εξέλιξη μιας παραγγελίας 1/2

Μια παραγγελία

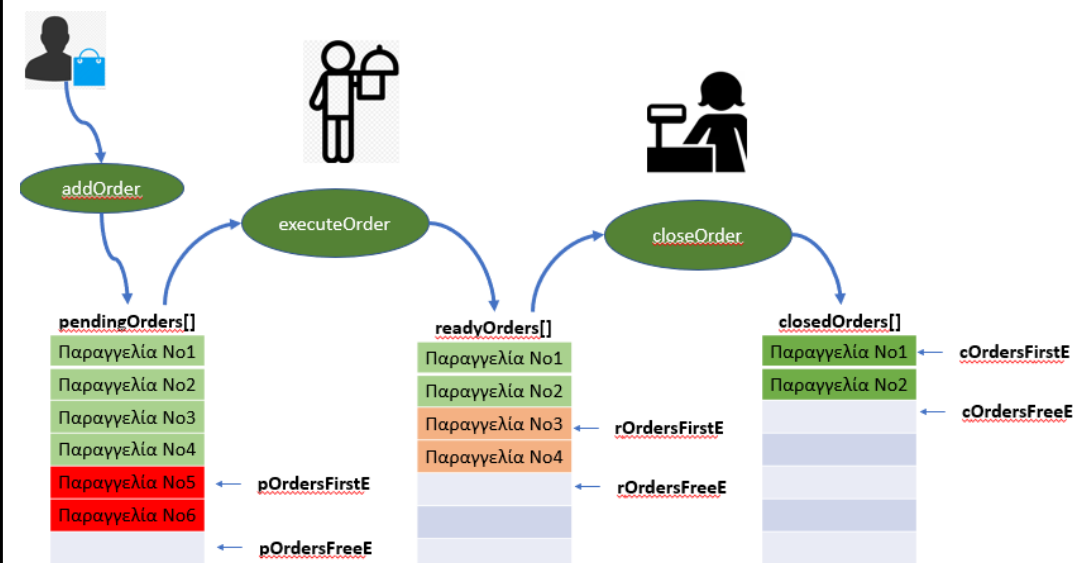
- όταν καταχωρείται είναι «εκκρεμής» (**pendingOrder**).
- όταν εκτελεστεί είναι «προς εξόφληση» (**readyOrder**).
- Μετά την εξόφληση είναι «κλειστή» (**closedOrder**).



Ενναλλακτικές Υλοποιήσεις



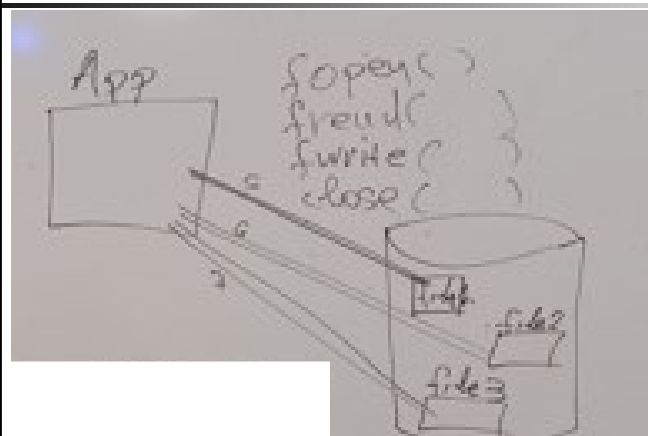
Η εξέλιξη μιας παραγγελίας 2/2



Κλεάνθης Θραμπουλίδης

Week Agenda

Επικοινωνία με αρχείο



Δράση 1

Γράψτε ένα πρόγραμμα που θα αποθηκεύει σε ένα αρχείο ένα αλφαριθμητικό (το ονοματεπώνυμο σας) και μετά θα το διαβάζει.

Δράση 2

Γράψτε ένα πρόγραμμα που θα αποθηκεύει σε ένα αρχείο ένα πίνακα αλφαριθμητικών και μετά θα το διαβάζει.

Χρησιμοποιείτε τις συναρτήσεις `fprintf` και `fscanf` ([OrdersHandlingApp.pdf](#) σελ. 11)

Δράση 3

Ενσωματώστε την λειτουργικότητα της αποθήκευσης και φόρτωσης παραγγελιών στην έκδοση στην οποία έχετε υλοποιήσει την λειτουργικότητα καταχώρησης παραγγελίας και εμφάνισης παραγγελιών.

Κλεάνθης Θραμπουλίδης

Week Agenda

Week 11 - Οργάνωση

- [EA-9] OrdersHandling
 - Σχολιασμός παραδοτέων (Lab time)
- EA-10
 - ClassroomApp
 - Ταξινόμηση (Sorting), Αναζήτηση (Search)
- Απαριθμητικός Τύπος (Enum)
- Αναδρομικότητα (recursion)

[EA-9] Εργαστήριο 10ης εβδομάδας (Αντικείμενο)

σήμερα - 12:30 μ.μ. - από τον χρήστη Θραμπουλίδης Κλεάνθης

Δράσεις

Η EA-9 αποτελείται από την ανάπτυξη μιας εφαρμογής για διαχείριση παραγγελιών.

[Δ1] - Διαχείριση Παραγγελιών (OrdersHandlingApp)

Η Δράση έχει στόχο την ανάπτυξη μιας εφαρμογής που θα επιτρέπει στον χρήστη να διαχειρίζεται παραγγελίες σε ένα εργοστάσιο που παράγει φιάλες εμφιάλωσης.

Αναλυτική περιγραφή και Οδηγίες για την ανάπτυξη του προγράμματος OrdersHandlingApp θα βρείτε στο αρχείο [OrdersHandlingApp.pdf](#).

[EA-10] Εργαστήριο 11ης εβδομάδας (Αντικείμενο)

Παρασκευή, 6 Δεκεμβρίου 2024 - 3:54 μ.μ. - από τον χρήστη Θραμπουλίδης Κλεάνθης

Η EA-10 αποτελείται από δύο δράσεις. Η μία είναι η ClassroomApp και η άλλη έχει σχέση με ταξινόμηση με έμφαση στην bubble sort.

Και τις δύο ασκήσεις θα τις δουλέψουμε στην αίθουσα πριν από το εργαστήριο.

[Δ1] - ClassroomApp

Η Δράση έχει στόχο την ανάπτυξη μιας εφαρμογής που θα επιτρέπει στον χρήστη να διαχειρίζεται τα στοιχεία φοιτητών που παρακολουθούν το εργαστήριο ενός μαθήματος στο Υπολογιστικό κέντρο.

Κλεάνθης Θραμπουλίδης

Week Agenda

83

[EA-9] OrdersHandling Deliverable (Σχολιασμός)

[EA-9] Εργαστήριο 10ης εβδομάδας (Αντικείμενο)

σήμερα - 12:30 μ.μ. - από τον χρήστη Θραμπουλίδης Κλεάνθης

Δράσεις

OrdersHandlingApp.pdf

Η EA-9 αποτελείται από την ανάπτυξη μιας εφαρμογής για διαχείριση παραγγελιών.

[Δ1] - Διαχείριση Παραγγελιών (OrdersHandlingApp)

Η Δράση έχει στόχο την ανάπτυξη μιας εφαρμογής που θα επιτρέπει στον χρήστη να διαχειρίζεται παραγγελίες σε ένα εργοστάσιο που παράγει φιάλες εμφιάλωσης.

Αναλυτική περιγραφή και Οδηγίες για την ανάπτυξη του προγράμματος OrdersHandlingApp θα βρείτε στο αρχείο [OrdersHandlingApp.pdf](#).

```
11: int main(int argc, char** argv) {
12:
13:     int small8; //Μικρα προυκαλια
14:     int big8; //Μεγαλα Μπουκαλια
15:     int tenax; //synolo tenaxiwn
16:
17:     //Plhrofories svetika ne proiouta
18:     printf("Timokatatologos Μπουκαλιw\n");
19:     printf("-----\n");
20:     printf("Μικρα Μπουκαλι: 0.08 euro\n");
21:     printf("Μεγαλο Μπουκαλι: 0.35 euro\n");
22:     printf("-----\n");
23:     printf("Εκπτώσεις:\n");
24:     printf("Paragelies eite anw ton 300 euro eite panw apo 100
ekptwsh 12%%\n");
25:     printf("Paragelies anw ton 600 euro, exoun ekptwsh 24%%\n");
26:     printf("-----\n");
27:     //plhrofories paragelias
28:     printf("Ypologismos Paragelias:\n");
29:     printf("-----\n");
30:     printf("Posa mikra μπουκαλια thes na paragelieis? \n");
31:     scanf("%d", &small8);
32:     return 0;
33: }
```

[EA-9]-LabTimeDeliverable-Group2

44

Ατομική εργασία

Ανάθεση σε: Συγκεκριμένες Ομάδες Χρηστών

[EA-9]-LabTimeDeliverable-Group1

39

Ατομική εργασία

Κλεάνθης Θραμπουλίδης

Week Agenda

84

[EA-9] 1^η έκδοση - Δομή της main (deliverables)

```
18: int main(int argc, char *argv[]) {
19:     struct Order orders[50];
20:     printf("Running Orders Handling\n");
21:     printf("0---Termination_Program\n");
22:     printf("1---Katawrhsh_Paragelias\n");
23:     printf("2---Enfanih_Paragelias_Pelath\n");
24:     choice = getInt("Duse Epilogh:");
25:     while(choice!=0){
26:         choice = getInt("Duse Epilogh:");
27:
28:         switch(choice){
29:             case 0:
30:                 printf("Ginetai termatismos tou programatos...");
31:                 break;
32:             case 1:
33:                 printf("Katawrhsh paragelias.\n");
34:                 struct Order orders[50];
35:                 printf("Duse onoma pelath:");
36:                 scanf("%49s",&orders[totalOrders].Name);
37:                 printf("Duse arithmo mikron poukalion:");
38:                 scanf("%d",&orders[totalOrders].smallBottles);
39:                 printf("Duse arithmo megalon poukalion:");
40:                 scanf("%d",&orders[totalOrders].bigBottles);
41:                 printf("Duse Hmeronhia prothessias(Format :./):");
42:                 scanf("%18s",&orders[totalOrders].date);
43:                 totalOrders++;
44:                 break;
45:             case 2:
46:                 printf("Enfanih zontai oi paragelies.\n");
47:                 int i;
48:                 for(i=0;i<totalOrders;i++){
49:                     printf("Paragelia No%02d:\n Onoma pelath:%s\n",i,orders[i].Name);
50:                 }
51:             }
52:         }
53:         return 0;
54:     }
```

```
36: int main(){
37:     int choice,lastOrder=1;
38:     ORDER orders[1]={"kosmas",50,100,04,12,2024};
39:
40:     printf("Orders Handling V2\n\n");
41:
42:     displayMenu();
43:
44:     choice=getInt("\n\nChoice: ");
45:
46:     printf("\n\n");
47:
48:     while(choice!=EXIT)
49:     {
50:         switch(choice)
51:         {
52:             case GET_ORDER:
53:                 /*
54:                  * if (lastOrder<1){
55:                  *     orders[lastOrder]=getOrder();
56:                  *     lastOrder++;
57:                  * }
58:                  */
59:                 break;
60:             case DISPLAY_CUSTOMERS_ORDER:
61:                 displayOrder(orders,lastOrder);
62:                 break;
63:             case DISPLAY_PENDING_ORDERS:
64:                 displayPendingOrders(orders,lastOrder);
65:                 break;
66:             case SAVE_PENDING_ORDERS:
67:                 // savePendingOrders();
68:                 break;
69:         }
70:         //displayMenu();
71:         choice=getInt("\n\nChoice: ");
72:     }
73: }
```

```
55: int main(int argc, char *argv[]) {
56:     Order order;
57:     printf("OrdersHandlingV2\n");
58:
59:     while (1) {
60:         menu();
61:
62:         switch (choice) {
63:             case 0:
64:                 exit(0);
65:             case 1:
66:                 GetOrder(&order);
67:                 break;
68:             case 2:
69:                 FindOrderMenu();
70:                 break;
71:             case 3:
72:                 MenuPendingOrders();
73:                 break;
74:             case 4:
75:                 MenuReadyOrders();
76:                 break;
77:             case 5:
78:                 MenuClosedOrders();
79:                 break;
80:             default:
81:                 printf("Unavailable option\n");
82:         }
```

Κλεάνθης Θραμπουλίδης

Week Agenda

[EA-9] 1^η έκδοση - Δομή της main (Οδηγίες)

12.5.2 Βήμα 1 – 1^η έκδοση (V1)

Ξεκινήστε ένα νέο project χωρίς να αξιοποιήσετε στην φάση αυτή τον κώδικα του βήματος 0.

Εφαρμόζουμε top-down προσέγγιση σύμφωνα με την οποία **θα γράψουμε πρώτα τη main()**. Θα την γράψουμε και θα ελέγχουμε την λειτουργικότητα της. Αυτό μπορεί να γίνει χωρίς να γράψουμε τα σώματα των συναρτήσεων που αυτή καλεί. Προσέξτε η main() δεν θα κάνει άλλη δουλειά από το να καλεί δικές σας συναρτήσεις. Ορίστε μια συνάρτηση για κάθε μια από τις λειτουργικότητες που το πρόγραμμα σας πρέπει να υλοποιεί. Την συνάρτηση αυτή θα καλεί η main όταν ο χρήστης επιλέξει την αντίστοιχη επιλογή του μενού.

Αν έχετε αποδώσει κατάλληλα ονόματα στις συναρτήσεις σας τότε διαβάζοντας κάποιοι τρίτος τον κώδικα σας θα καταλάβει τι κάνει η εφαρμογή σας σε απόκριση των επιλογών του χρήστη. Το πως η κάθε συνάρτηση θα κάνει την δουλειά της θα το ορίσετε σε επόμενο βήμα όταν γράψετε το σώμα της.

Ορίστε την δομή της main() με την παράλληλη ανάπτυξη του μενού επιλογής λειτουργίας, όπως ακριβώς κάνατε στην [EA-5]. **12.6.1 Βήμα 1 – 1^η έκδοση**

Η διαδικασία ανάπτυξης της κύριας ροής του προγράμματος (που είναι και η κύρια ροή της main) περιγράφεται αναλυτικά στην Άσκηση 4 – Ταξινόμηση Λέξεων (κεφάλαιο 11) και πιο συγκεκριμένα στο Βήμα 1.

Κλεάνθης Θραμπουλίδης

Week Agenda

[EA-9] 2^η έκδοση – ORDER (deliverables)

```
typedef struct {
    char onomateponimo[20];
    int hmeromhniaOrder[20];
    int hmeromhniaEpythimias[20];
    int arxikoKostos[10];
    int ekptwsh[10];
    int tekikoKostos[10];
    char hmeromhniaEkteleshs[10];
}ORDER;

24: struct order{
25:     char onoma;
26:     char eponumo;
27:     Hm hmPar;
28:     Hm hmEkt;
29:     float arkostos;
30:     float mikrhektwsh;
31:     float megalhektwsh;
32:     float telkostos;
33:     Hm hmPragmEkt;
34: };

7: struct bottleOrder{
8:     int amountOfSBottles;
9:     int amountOfBBottles;
10:    char FullName[100];
11:    float discount;
12:    float initCost;
13:    float finalCost;
14:    char dateOrdered[20];
15:    char dateExecuted[20];
16:    char dateFinallyExecuted[20];
17:    int isOrderExecuted;
18:    int isOrderReady;
19:    int isOrderPending;
20: };

13: typedef struct {
14:     char customerName[MAX_NAME];
15:     char orderDate[DATE_LEN];
16:     char executionDate[DATE_LEN];
17:     char executedDate[DATE_LEN];
18:     int smallBottles;
19:     int largeBottles;
20:     double initialCost;
21:     double discount;
22:     double finalCost;
23:     int status;
24: } Order;

30: typedef struct Order{
31:     int sBottles;
32:     int bBottles;
33:     char fullName[40];
34:     double originalCost;
35:     double finalCost;
36: }order;
```

Απαριθμητικός Τύπος (**enum**)

[I2P_Variables&DataTypes.pdf](#) (slide 26)

Παράδειγμα 6-7 (κεφάλαιο 6)

enum choice { ADD=1, SUB, ...

3.6 Απαριθμητικός Τύπος (κεφάλαιο 3)

enum months { JAN=1, ...

Κλεάνθης Θραγουλίδης

Week Agenda

[EA-9] 2^η έκδοση – getOrder() (deliverables)

```
117: void addOrder(Order *orders, int *orderCount) {
118:     if (*orderCount >= MAX_ORDERS) {
119:         printf("Den mporoun na prostethoun alles paraggelies.\n");
120:         return;
121:     }
122:     Order newOrder;
123:     printf("Onoma Pelath: ");
124:     scanf("%s", newOrder.customerName);
125:     printf("Hmeromhnia paraggelias (dd/mm/yyyy): ");
126:     scanf("%s", newOrder.orderDate);
127:     printf("Hmeromhnia ekteleshs (dd/mm/yyyy): ");
128:     scanf("%s", newOrder.executionDate);
129:     printf("Arithmos mikrwv mpoukaliwn: ");
130:     scanf("%d", &newOrder.smallBottles);
131:     printf("Arithmos megalwn mpoukaliwn: ");
132:     scanf("%d", &newOrder.largeBottles);
133:
134:     newOrder.initialCost = calculateInitialCost(newOrder.smallBot
newOrder.largeBottles);
135:     newOrder.discount = 0.0;
136:     newOrder.finalCost = 0.0;
137:     strcpy(newOrder.executedDate, "");
138:     newOrder.status = 0;
139:
140:     orders[*orderCount] = newOrder;
141:     (*orderCount)++;
142:     printf("Kataxwrithe h paraggelia.\n");
143: }

102: void GetOrder(Order *order) {
103:     OrderName(order);
104:     OrderSurname(order);
105:     OrderDate(order);
106:     OrderId(order);
107:     OrderMenu(order);
108:     MovetoPending(order);
109: }
110:
111: void OrderName(Order *order) {
112:     printf("Enter your name (max 20 characters): ");
113:     fgets(order->name, sizeof(order->name), stdin);
114:     size_t len = strlen(order->name);
115:     if (len > 0 && order->name[len - 1] == '\n') {
116:         order->name[len - 1] = '\0';
117:     }
118: }
119:
120: void OrderSurname(Order *order) {
121:     printf("Enter your surname (max 20 characters): ");
122:     fgets(order->surname, sizeof(order->surname), stdin);
123:     size_t len = strlen(order->surname);
124:     if (len > 0 && order->surname[len - 1] == '\n') {
125:         order->surname[len - 1] = '\0';
126:     }
127: }
128:
129: void OrderDate(Order *order) {
130:     int dd, mm, yyyy;
131:     printf("Enter the date (DD-MM-YYYY): ");
132:     if (scanf("%d-%d-%d", &dd, &mm, &yyyy) != 3) {
133:         printf("Invalid date format. Please use DD-MM-YYYY.\n");
134:         exit(1);
135:     }
136:     order->date[0] = dd / 10;
137:     order->date[1] = dd % 10;
138:     order->date[2] = mm / 10;
```

Κλεάνθης Θραγουλίδης

Week Agenda

[EA-9] 2^η έκδοση – getOrder() (Οδηγίες)

12.6.2 Βήμα 2 – 2^η έκδοση

Ορίστε τη δομή (struct) που αναπαριστά την παραγγελία και στην συνέχεια τον πίνακα παραγγελιών.

Καταχώρηση Παραγγελίας

Υλοποιήστε την συνάρτηση που θα διαβάζει από την κύρια είσοδο μία παραγγελία. Πιθανότατα θα την ονομάσετε `getOrder`. Είναι αρμοδιότητα της συνάρτησης που θα καλέσει την `getOrder` να βάλει την παραγγελία στον πίνακα παραγγελιών. Δώστε της κατάλληλο όνομα. Αυτή θα τοποθετεί την παραγγελία στην επόμενη ελεύθερη θέση του πίνακα.

Για να απλοποιήσουμε την υλοποίηση θα θεωρήσουμε πως οι παραγγελίες καταχωρούνται στον πίνακα η μία μετά την άλλη. Μια μεταβλητή με όνομα `freeEntry` μπορεί να περιέχει την τάξη στον πίνακα της πρώτης διαθέσιμης θέσης για καταχώρηση παραγγελίας. Φροντίστε να την κρατάτε ενήμερη κατά τη διάρκεια λειτουργίας του προγράμματός σας.



[EA-9] 2^η έκδοση – displayPendingOrders() (deliverables)

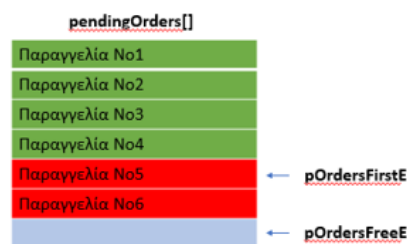
```
212: void displayPendingOrders(Order o[], int currPending, int
    lastPending, int maxlen){
213:     if(lastPending - currPending > 0){
214:         printf("Order No   Buyer's Name      BB    SB    Init
    Cost   Final Cost   Date of Order\n");
215:         printf("-----\n");
216:         for (int i = currPending; i < lastPending; i++){
217:             printf("Order %-3d: %-20s %-6d %-6d %-10.21f %-10.21f
    %d/%d/%d\n", i + 1, o[i].buyerName, o[i].numOfBB, o[i].numOfSB,
    o[i].initCost, o[i].finalCost, o[i].dateOfOrder.day,
    o[i].dateOfOrder.month, o[i].dateOfOrder.year);
219:         } else if(currPending - lastPending > 0) {
220:             for (int i = currPending; i < maxlen - currPending +
    lastPending; i++){
222:                 printf("Order %-3d: %-20s %-6d %-6d %-10.21f %-10.21f
    %d/%d/%d\n", i + 1, o[i % maxlen].buyerName, o[i % maxlen].numOfBB,
    o[i % maxlen].numOfSB, o[i % maxlen].initCost, o[i %
    maxlen].finalCost, o[i % maxlen].dateOfOrder.day, o[i %
    maxlen].dateOfOrder.month, o[i % maxlen].dateOfOrder.year);
223:             }
224:         } else {
225:             printf("There are no pending orders\n");
226:         }
227:     }
```



[EA-9] 2^η έκδοση – displayPendingOrders() (Οδηγίες)

Εκκρεμείς Παραγγελίες

Υλοποιήστε την συνάρτηση που θα εμφανίζει στην κύρια έξοδο όλες τις εκκρεμείς παραγγελίες. Στην φάση αυτή όλες οι παραγγελίες που είναι καταχωρημένες στον πίνακα παραγγελιών είναι εκκρεμείς. Άρα η επιλογή αυτή εμφανίζει στην οθόνη όλες τις παραγγελίες του πίνακα. Το Σχήμα 2 εμφανίζει ένα στιγμιότυπο από μια ενδεικτική υλοποίηση η οποία περιέχει ορισμένα μόνο στοιχεία της παραγγελίας (ονοματεπώνυμο πελάτη, ποσότητα μικρών φιαλών, ποσότητα μεγάλων φιαλών, αρχικό και τελικό κόστος). **Οι παραγγελίες είναι εκκρεμείς οπότε το αρχικό κόστος καθώς και το τελικό δεν έχουν υπολογιστεί ακόμη.**



Κλεάνθης Θραμπουζίδης

Week Agenda

[EA-10] [Δ1] – ClassroomApp

Η Δράση έχει στόχο την ανάπτυξη μιας εφαρμογής που θα επιτρέπει στον χρήστη να διαχειρίζεται τα στοιχεία φοιτητών που παρακολουθούν το εργαστήριο ενός μαθήματος στο Υπολογιστικό κέντρο.

Το υπολογιστικό κέντρο είναι οργανωμένο σε 4 ζώνες (Z1-Z4) και η κάθε ζώνη έχει 18 θέσεις (S1-S18). Ο κάθε φοιτητής έχει συγκεκριμένη θέση σε όλη τη διάρκεια του εξαμήνου.

A) Γράψτε ένα πρόγραμμα σε C σύμφωνα με το οποίο το σύστημα θα εκτελεί τις παρακάτω λειτουργίες:

1. [Καταχώρηση]

Θα δέχεται για κάθε φοιτητή, που κάθεται στην αίθουσα, το ονοματεπώνυμο, τον αριθμό μητρώου και την ημερομηνία εγγραφής του καθώς και την θέση του (ζώνη-θέση) και θα τα αποθηκεύει στη μνήμη.

2. [Εμφάνιση]

2.1 Θα εμφανίζει στην οθόνη όλη την πληροφορία που έχει καταχωρηθεί.

2.2 Θα εμφανίζει στην οθόνη την λίστα των φοιτητών μιας ζώνης με την θέση τους σε αυτή.

3. [Αναζήτηση]

3.1 Θα δέχεται αριθμό μητρώου ή ονοματεπώνυμο φοιτητή και θα ενημερώνει τον χρήστη για την θέση του φοιτητή αν αυτός είναι καταχωρημένος.

3.2 Θα εμφανίζει τα στοιχεία φοιτητή για δεδομένη ζώνη και θέση.

Κλεάνθης Θραμπουζίδης

Week Agenda

[EA-10] [Δ2] – Sorting

13. Άσκηση 13 – Ταξινόμηση και Αναζήτηση (Sorting and Searching)

Η άσκηση έχει στόχο την εισαγωγή βασικών εννοιών ταξινόμησης και αναζήτησης και την εξοικείωση με τις διαδικασίες ταξινόμησης και αναζήτησης στοιχείων σε πίνακα

Η **ενότητα 1** περιγράφει το αντικείμενο το οποίο θα πρέπει να ικανοποιεί η εφαρμογή.

Η **ενότητα 2** αναφέρεται στην αξιοποίηση έτοιμων συναρτήσεων για την ταξινόμηση πίνακα ακεραίων και αλφαριθμητικών. Δίνει έμφαση και στην ανάπτυξη κώδικα που θα έχει δεδομένη έξοδο στην οθόνη.

Η **ενότητα 3** περιγράφει εν συντομία τον αλγόριθμο ταξινόμησης bubble sort

Η **ενότητα 4** ζητά την ανάπτυξη ενός προγράμματος που θα χρησιμοποιεί την συνάρτηση bubbleSort() και θα μας προετοιμάσει για τον ορισμό της.

Εργαστηριακή Άσκηση [EA-10]

Η **ενότητα 5** αναφέρεται στη διαδικασία που θα πρέπει να ακολουθήσετε για την σταδιακή **ανάπτυξη της bubbleSort()**. Περιγράφονται τρεις εκδόσεις οι οποίες υλοποιούν σταδιακά την λειτουργικότητα της ταξινόμησης με στόχο να οδηγηθούμε σε εναλλακτικές υλοποιήσεις. Η διαδικασία δίνει έμφαση και στην ανάπτυξη κώδικα που θα έχει δεδομένη έξοδο στην οθόνη.