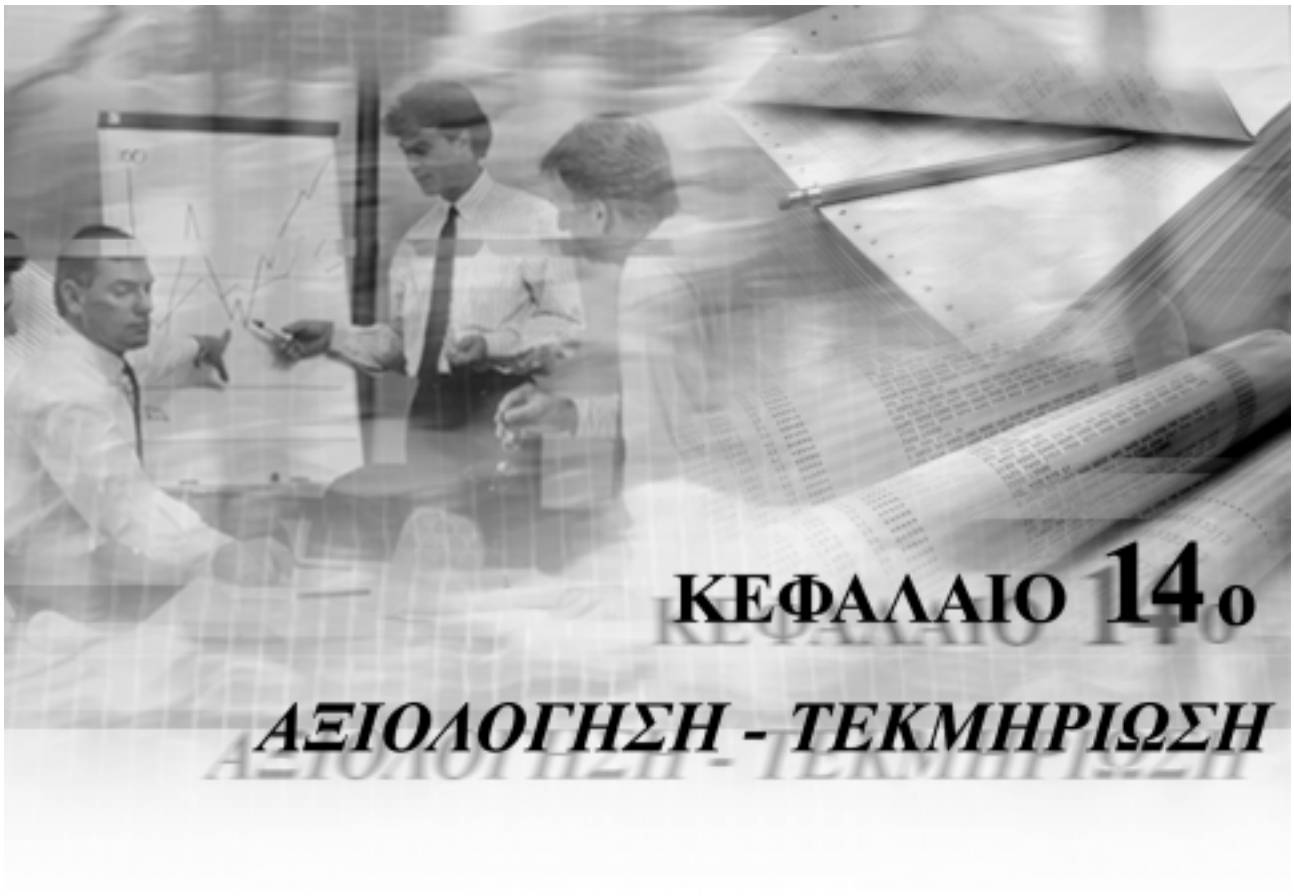

ΚΕΦΑΛΑΙΟ 14ο

ΑΞΙΟΛΟΓΗΣΗ - ΤΕΚΜΗΡΙΩΣΗ

14.1. Προσδοκώμενα αποτελέσματα

Σ' αυτό το τελευταίο κεφάλαιο του βιβλίου θα σου δοθεί η ευκαιρία να γνωρίσεις ποια είναι τα κριτήρια με τα οποία αξιολογούνται τα προγράμματά που κατασκευάζεις. Θα σου δοθεί η δυνατότητα μέσα από τις ασκήσεις που ακολουθούν να διατυπώσεις διαφορετικές λύσεις για το ίδιο πρόβλημα, εφαρμόζοντας τη μικρή πείρα που έχεις ήδη αποκτήσει και τις ιδέες που περιμένουμε από σένα.

Δοκιμάζοντας τις λύσεις που έχουν κατασκευάσει άλλοι, και προσπαθώντας να ανακαλύψεις τα όρια αξιοπιστίας των προγραμμάτων τους, θα συγκεντρώσεις χρήσιμες παρατηρήσεις, που θα μπορείς να εφαρμόσεις στα δικά σου προγράμματα.

14.2. Επιπλέον παραδείγματα

Παράδειγμα 1

Δίνεται ο δισδιάστατος πίνακας A(i,k) όπου στην πρώτη διάσταση περιέχει την κωδικό είδους ενός υλικού και στη δεύτερη την ποσότητα του υλικού. Ο πίνακας Α είναι ταξινομημένος κατά είδος. Ζητείται να εκτυπώνονται σύνολα ποσότητας, σε κάθε αλλαγή είδους και η συχνότητα εμφάνισης κάθε κωδικού. Στο τέλος της επεξεργασίας να τυπώνεται γενικό σύνολο ποσότητας.

Παρακάτω δίνεται ένας πίνακας με εικονικά δεδομένα και τα αποτελέσματα που πρέπει να παράγει το πρόγραμμά μας με αυτά τα δεδομένα.

Δεδομένα		Αποτελέσματα	
Κωδικός Είδους	Ποσότητα	Σύνολο Είδους	Γενικό Σύνολο
101	20		
101	30		
101	15		
101	20		
101	22	107	
105	1		
105	7	8	
200	17		
200	2		
200	3	22	
250	29	29	
280	12		
280	13	25	
310	6	6	
320	7	7	
330	9	9	213

Το πρόβλημα είναι παρόμοιο με το παράδειγμα 1 του βιβλίου. Εδώ τη θέση του αριθμού παίρνει ο κωδικός είδους. Επομένως θα αναζητήσεις την αλλαγή τιμής στο πεδίο του κωδικού είδους. Αν ακολουθήσεις την λογική της πρώτης λύσης του 1ου παραδείγματος του βιβλίου, θα έχεις το παρακάτω πρόγραμμα σαν λύση του προβλήματος.

```
ΠΡΟΓΡΑΜΜΑ Σύνολα_Είδους
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: A, GS, i, S, Προηγ_Α
ΑΡΧΗ
GS <- 0
i <- 1
S <- 0
Προηγ_Α <- Α(1,1)
ΟΣΟ i<1001 ΕΠΑΝΑΛΑΒΕ
    ΑΝ Προηγ_Α<>Α(i,1) ΤΟΤΕ
        ΓΡΑΨΕ Προηγ_Α, S
        GS <- GS+S
        Προηγ_Α <- Α(i,1)
        S <- 0
    ΤΕΛΟΣ_ΑΝ
```

```
    S <- S+A(i,2)
    i <- i+1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Προηγ_Α, S
GS <- GS+S
ΓΡΑΨΕ GS
ΤΕΛΟΣ Σύνολα_Είδους
```

Παράδειγμα 2

'Ενας αριθμός λέγεται πρώτος αν διαιρείται μόνο από το 1 και τον εαυτό του. 'Εστω, λοιπόν, ότι δίνεται ένας θετικός ακέραιος αριθμός n και ζητείται να διαπιστωθεί αν είναι πρώτος ή όχι. Μία πρώτη σκέψη είναι να αρχίσουμε να διαιρούμε τον αριθμό αυτόν διαδοχικά με 2, 3, 4, ..., n-1. 'Ετσι, αν ο αριθμός n δεν διαιρείται ακριβώς με κανένα από τους αριθμούς αυτούς, τότε πράγματι ο αριθμός n είναι πρώτος.

```
Αλγόριθμος ΠρώτοςΑριθμός
Διάβασε n
flag ← Ψευδής
i ← 2
Αρχή_επανάληψης
    Αν n mod i = 0 τότε flag ← Αληθής
    i ← i+1
Μέχρις_ότου flag=Αληθής ή i>n-1
Αποτελέσματα // flag //
Τέλος ΠρώτοςΑριθμός
```

Ο προηγούμενος αλγόριθμος χρησιμοποιεί μία συνάρτηση "mod" που επιστρέφει το υπόλοιπο της ακέραιης διαίρεσης. Αν αυτό είναι μηδέν τότε ο αριθμός δεν είναι πρώτος και η σημαία flag γίνεται Αληθής.

Με λίγη προσοχή και ελάχιστο κόπο, ο προηγούμενος αλγόριθμος μπορεί να γίνει ταχύτερος αν αλλάξει η συνθήκη της εντολής "Αρχή_επανάληψης ... μέχρις ότου" από i>n-1 σε i>Ρίζα(n). Αυτό μπορεί να γίνει με βεβαιότητα, καθώς αποδεικνύεται και μαθηματικά. Για παράδειγμα, αν θέλουμε να διαπιστώσουμε αν ο αριθμός 11 είναι πρώτος, τότε δεν είναι απαραίτητο να διαιρέσουμε το 11 δια 2, 3, κλπ μέχρι το 10, αλλά μέχρι το 4. Ο λόγος είναι ότι αν το 11 διαιρείτο ακριβώς δια του 5, του 6 κλπ, τότε θα διαιρείτο ακριβώς και δια του 11/5, 11/6 κοκ, γεγονός που θα είχε ήδη αποκαλυφθεί.

Παράδειγμα 3. Πίνακας Αποφάσεων

Είναι γνωστό ότι δίσεκτο έτος λέγεται αυτό που ο αριθμός του είναι πολλαπλάσιο του 4, αλλά όχι του 100, εκτός αν είναι πολλαπλάσιο του 400. Για παράδειγμα: το 1984 είναι δίσεκτο (πολλαπλάσιο του 4), το 1900 δεν είναι (πολλαπλάσιο του 4 αλλά και του 100), το 1600 είναι δίσεκτο (πολλαπλάσιο του 4, του 100, αλλά και του 400), τέλος το 1993 δεν είναι. Η διατύπωση της λύσης για ένα τέτοιο έλεγχο είναι γενικά δύσκολη υπόθεση. Ας δούμε ένα τμήμα προγράμματος που δίνει μία λύση.

```
.....
E4 <- E MOD 4
E100 <- E MOD 100
E400 <- E MOD 400
ΑΝ ε4=0 ΤΟΤΕ
    δις <- 1
ΤΕΛΟΣ_ΑΝ
ΑΝ ε100=0 ΤΟΤΕ
    δις <- 0
ΤΕΛΟΣ_ΑΝ
ΑΝ ε400=0 ΤΟΤΕ
    δις <- 1
ΤΕΛΟΣ_ΑΝ
.....
```

‘Οταν το δις είναι 1 το έτος είναι δίσεκτο, όταν είναι 0, δεν είναι.

Η κωδικοποίηση των **Αν** δείχνει ότι είναι ανεξάρτητα, όμως δεν είναι. Πράγματι αν αλλάξεις την σειρά των **Αν** και δοκιμάσεις το πρόγραμμα που θα προκύψει, θα διαπιστώσεις ότι δίνει διαφορετικά αποτελέσματα.

Αυτό συμβαίνει γιατί δόθηκε μία λύση, χωρίς προηγούμενα να έχει κατανοηθεί πλήρως η αλληλεξάρτηση των συνθηκών. Σ’ αυτές τις περιπτώσεις μπορείς να χρησιμοποιήσεις ένα άλλο τρόπο εύρεσης της λύσης, που θα σε βοηθήσει στην βαθύτερη κατανόηση του προβλήματος, άρα και στη σωστή προσέγγιση της λύσης. Μπορείς να χρησιμοποιήσεις ένα **Πίνακα Αποφάσεων**. ‘Ενας πίνακας αποφάσεων έχει την εξής μορφή:

Συνθήκες (που θα ελέγχονται)	Τιμή της Συνθήκης (Αληθής, Ψευδής)
Ενέργειες / Συμπέρασμα	Τιμή ενέργειας (Ναι, ‘Οχι)

Ο πίνακας αποφάσεων ενδείκνυται όταν υπάρχουν πολλές συσχετιζόμενες συνθήκες που προκαλούν διαφορετικές ενέργειες / συμπεράσματα. Η συμπλήρωση των στηλών γίνεται από όλους τους δυνατούς συνδυασμούς Α και Ψ των συνθηκών που πρέπει να ελεγχθούν μέσα στο πρόγραμμα. Αυτοί είναι σε πλήθος 2ν, όπου ν το πλήθος των διαφορετικών συνθηκών. Το τμήμα με τις ενέργειες / συμπεράσματα συγκροτείται από μία ή περισσότερες λογικές προτάσεις που είναι και ενέργειες του προγράμματος σου. Σε κάθε στήλη, αφού ελεγχθεί η λογική του συνδυασμού των Α και Ψ, γράφεται η τιμή της ενέργειας που πρέπει εκτελεστεί ή το συμπέρασμα που προκύπτει.

Για το συγκεκριμένο πρόβλημα θέλεις να γνωρίζεις αν το έτος είναι πολλαπλάσιο, δηλαδή αν διαιρείται ή δεν διαιρείται ακριβώς με το 4, το 100 ή το 400 προκείμενου να συμπεράνεις ότι το έτος είναι δίσεκτο. ‘Αρα στον παραπάνω πίνακα έχεις τρεις συνθήκες, που οι πιθανοί συνδυασμοί των απαντήσεων τους είναι 8. Ενώ το συμπέρασμα είναι ένα, αν δηλαδή το έτος είναι δίσεκτο ή όχι.

Ας σχηματίσεις λοιπόν τον πίνακα αποφάσεων για το δίσεκτο έτος, θα έχεις:

Συνθήκες	Περιπτώσεις							
	1η	2η	3η	4η	5η	6η	7η	8η
E mod 4 =0	Α	Α	Α	Α	Ψ	Ψ	Ψ	Ψ
E mod100=0	Α	Α	Ψ	Ψ	Α	Ψ	Α	Ψ
E mod 400=0	Α	Ψ	Α	Ψ	Α	Α	Ψ	Ψ
Ενέργειες/ Συμπεράσματα								
Δίσεκτο	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ

Από την παρατήρηση του πίνακα προκύπτει ότι μόνο η 1η και η 4η περίπτωση οδηγούν στο συμπέρασμα ότι το έτος είναι δίσεκτο, αυτό το συμπέρασμα στο πρόγραμμα θα το υλοποιήσεις με μία μεταβλητή δύο τιμών (πχ. 1:είναι, 0: δεν είναι). Επομένως τώρα γνωρίζεις πώς να συντάξεις τις σχετικές ερωτήσεις, και την σχέση που υπάρχει μεταξύ τους. Μία λύση δίνεται στο 4ο παράδειγμα του βιβλίου. Πρόσεξε ότι στην 4η περίπτωση του πίνακα, όταν E mod 100 ≠ 0 αναγκαστικά θα είναι και E mod 400 ≠ 0 και άρα δεν χρειάζεται να γίνει επιπλέον έλεγχος.

14.3. Συμβουλές

- Αν στο πρόγραμμα σου υπάρχουν τμήματα για τα οποία μπορείς να δώσεις ένα όνομα όπως "Υπολογισμός Παραγγελίας", "Διάβασμα και έλεγχος", "Αλλαγή κωδικού" και άλλα παρόμοια, τότε αυτά είναι κατάλληλα να γίνουν ενότητες.
- Αντιμετώπιζε παρεμφερή θέματα με τον ίδιο τρόπο. Αυτό θα δώσει "υπογραφή" στα προγράμματα σου. Αν η "υπογραφή" αποδειχτεί "κακή", αυτό θα το καταλάβεις γρήγορα.
- Να κρατάς (μαζεύεις) σημειώσεις όπως ο φιλάργυρος μαζεύει δεκάρες. Αργότερα θα ξεχωρίσεις και θα κρατήσεις αυτές που αξίζουν.
- Να ζητάς από άλλους να δοκιμάσουν το "σωστό" πρόγραμμα σου. Αν δεν είναι φίλοι ακόμα καλύτερα.
- Γράφε σχόλια ανάμεσα στις γραμμές του κώδικα.
- 'Οταν σχεδιάζεις τη λύση του προβλήματος, να σκέπτεσαι, "τι θα μου ζητήσουν μετά;"
- Αν το πρόγραμμα είναι μεγάλο σχεδίασε πρώτα ένα VTOC, είναι καλή ιδέα.
- Αν πρέπει να εξετάσεις πολύπλοκες συνθήκες, δεν κάνεις καλύτερα έναν πίνακα αποφάσεων;
- Να χρησιμοποιείς δοκιμαστικά δεδομένα για να ελέγχεις τη λύση. Αν κάπου βρήκες λάθη, ξανά έλεγξε. Εκεί που βρέθηκε ένα λάθος πιθανόν να υπάρχουν και άλλα.

- Μετά από κάθε αλλαγή στο πρόγραμμά σου, να το ξανά ελέγχεις με όλα τα δοκιμαστικά δεδομένα που είχες χρησιμοποιήσει και να συμπληρώνεις την τεκμηρίωση.
- Μια βιαστική λύση καταλήγει σε πρόγραμμα "μιας χρήσης".
- Μην γράφεις σύνθετες συνθήκες στα **Αν**, προτίμησε να γράφεις φωλιασμένα **Αν**, ένα για κάθε συνθήκη. Εάν το επιτρέπει η λογική του προγράμματος, μετάτρεψέ τα σε απλά **ΑΝ**.
- Μην "κάνεις εξυπνάδες", ακόμη και αν νομίζεις ότι έχεις τον απόλυτο έλεγχο. Εκτός του ότι δεν υπάρχει απόλυτος έλεγχος, σκέψου μήπως δεν υπάρχει και λόγος.
- Μεταξύ των νέων προγραμματιστών κυκλοφορεί η άποψη ότι το πρόγραμμα είναι σαν το χταπόδι. "'Οσο το κτυπάς (με τον compiler), τόσο μαλακώνει, στο τέλος θα τρώγεται".

4. Δραστηριότητες

Στην Τάξη

ΔΤ1. Τι ημέρα ήταν πριν ή μετά από χ μέρες. Διερεύνησε ποια είναι τα απαραίτητα δεδομένα.

Υπόδειξη: Χρησιμοποίησε ένα πίνακα με τις ημέρες της εβδομάδος. Αν γνωρίζεις ότι η σήμερα είναι Τετάρτη, και το χ είναι +10. Τότε επειδή 10mod7=3, σημαίνει ότι μετά από δέκα μέρες θα είναι Τετάρτη και 3 ημέρες, δηλαδή Σάββατο.

ΔΤ2. Ο πίνακας αποφάσεων στο 3ο παράδειγμα περιέχει οκτώ στήλες όσοι και οι δυνατοί συνδυασμοί των τριών προτάσεων. Αφού τον μελετήσεις, να αποκλείσεις τις περιπτώσεις που δεν μπορούν να συμβούν (αλληλοαναιρούνται). Στην συνέχεια με τη βοήθεια του πίνακα αποφάσεων πρότεινε εναλλακτικές λύσεις με τμήμα κώδικα. Δεδομένα ελέγχου: τα 1996, 1600, 2000 είναι δίσεκτα, τα 1500, 1998, 1900, 2006 δεν είναι δίσεκτα.

ΔΤ3. Να γραφεί το τμήμα του προγράμματος που πρέπει να προστεθεί στη λύση του 3ου παραδείγματος του βιβλίου, ώστε να εμφανίζονται έξη αριθμοί α, β, γ, δ, ε, ζ σε αύξουσα σειρά.

ΔΤ4. Δίνεται από το πληκτρολόγιο σειρά λέξεων. Να σχεδιαστεί αλγόριθμος που θα τις κατατάσσει με λεξικογραφική σειρά και θα τις εμφανίζει στην οθόνη σε τρεις στήλες.

Υπόδειξη: 'Ορισε τα όρια του προγράμματος σου και δώσε τη σχετική πληροφορία στο χρήστη.

Στο σπίτι

ΔΣ1. Να κατασκευαστεί πρόγραμμα που θα δέχεται μία ημερομηνία με τη μορφή ηη/μμ/εεεε και θα δίνει την ημέρα της εβδομάδος που αντιστοιχεί. Να γίνει εφαρμογή στην ημέρα γέννησής σου.

Υπόδειξη: Βασίσου στη λύση της ΔΤ1

ΔΣ2. Να μετατραπεί ένας αριθμός του δεκαδικού συστήματος στον ίσο του ελληνικού αριθμητικού συστήματος.

α, β, γ, δ, ε, στίγμα, ζ, η, θ
ι, κ, λ, μ, ν, ξ, ο, π, κόππα
π, ρ, σ, τ, υ, φ, χ, ψ, ω, σαμπί

ϛ = στίγμα
ϟ = κόππα
ϡ = σαμπί

ΔΣ3. Μια αεροπορική εταιρία έχει δύο κατηγορίες θέσεων, Α' θέση και Τουριστική. Να κατασκευαστεί πρόγραμμα που θα εκδίδει εισιτήριο ανάλογα με την επιθυμία του πελάτη. Οι επιθυμίες του πελάτη μετά από έρευνα βρέθηκε ότι είναι:

Ζητά εισιτήριο Α' θέσης αποκλειστικά.
Ζητά εισιτήριο Τουριστικής θέσης αποκλειστικά.
Θέλει να ταξιδέψει οπωσδήποτε.
Να κατασκευαστεί μόνο ο πίνακας αποφάσεων για το συγκεκριμένο πρόγραμμα.
Να συνταχθεί ο φάκελος τεκμηρίωσης του προγράμματος.

Υπόδειξη: Πρόσεξε ότι η έκδοση εισιτηρίου είναι και συνάρτηση της ύπαρξης ή όχι διαθέσιμου εισιτηρίου.

Ο πίνακας αποφάσεων θα έχει πολλές στήλες / περιπτώσεις. Μετά το αρχικό ανάπτυγμα του πίνακα εξέτασε ποιες περιπτώσεις δεν είναι δυνατόν να συμβούν και σύμπτυξε τον πίνακα.

ΔΣ4. Να κατασκευάσετε πρόγραμμα που θα διαβάζει δυάδες αριθμών από το πληκτρολόγιο και θα τους εμφανίζει με αύξουσα σειρά στην οθόνη.

Το ίδιο για τριάδες αριθμών. Τέλος να συνδυάσεις τα παραπάνω σε ένα πρόγραμμα που θα εκτελεί και τις δύο λειτουργίες ανάλογα με την επιθυμία του χρήστη.

Να συνταχθεί ο φάκελος τεκμηρίωσης του προγράμματος.

Υπόδειξη: Συνδύασε το 2ο και 3ο παράδειγμα του βιβλίου.

Στο εργαστήριο

ΔΕ1. Δίνεται ταξινομημένος πίνακας με στοιχεία, 2, 2, 3, 3, 3, 3, 5, 6, 6, 6, 6, 6, 7, 8, 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 9. Ζητείται να κατασκευαστεί πρόγραμμα θα εμφανίζει την συχνότητα του αριθμού και την θέση του. Σε περίπτωση που ο αριθμός βρίσκεται πε-

ρισσότερες από μία φορά, η θέση να είναι η πρώτη που παρουσιάζεται.

Για τα δεδομένα που είδη δώσαμε το πρόγραμμα σου θα πρέπει να δίνει τα παρακάτω αποτελέσματα:

Αριθμός	Συχνότητα	Θέση
2	2	1
3	4	3
5	1	7
6	5	8
7	1	13
8	6	14
9	6	20

Υπόδειξη: ακολούθησε τα βήματα του 1ου παραδείγματος του τετραδίου.

ΔΕ2. Τι αλλαγή θα κάνεις στο πρόγραμμα της ΔΕ1 ώστε να εμφανίζεται η τελευταία θέση που βρέθηκε ο ίσος αριθμός;

Για τα δεδομένα σου τα αποτελέσματα θα είναι:

Αριθμός	Συχνότητα	Θέση
2	2	2
3	4	6
5	1	7
6	5	12
7	1	13
8	6	19
9	6	25

ΔΕ3. Στην αρχαία Ελλάδα η μετάδοση των μηνυμάτων γινόταν με φρυκτωρίες (από φρυκτός = πυρσός και ώρα = φροντίδα). Αναφορές του Όμηρου το επιβεβαιώνουν. Η μέθοδος περιγράφεται από τον ιστορικό Πολύβιο και επινοήθηκε το 350 π.χ. από τους Αλεξανδρινούς τεχνικούς Κλεοξένη και Δημόκλειτο.
Η μέθοδος απαιτούσε δύο ή περισσότερους σταθμούς σε απόσταση που σήμερα υπολογίζεται σε 30 km. Κάθε σταθμός είχε δύο ομάδες των πέντε δαυλών με επάλειψη ρετσίνι ή ακάθαρτο πετρέλαιο από την περιοχή Κερί της Ζακύνθου, από τότε μέχρι σήμερα) το άγγαρο (άσβεστο) πυρ που αναφέρει ο Αισχύλος. Εχοντας χωρίσει την αλφάβητο σε πέντε πεντάδες, η πρώτη ομάδα των πέντε πυρσών έδειχνε τη στήλη ενώ η δεύτερη ομάδα τη γραμμή. Ετσι ο συνδυασμός των αναμμένων πυρσών αντιστοιχούσε σε συγκεκριμένο γράμμα, άρα μπορούσε να μεταδοθεί ένα μήνυμα

γράμμα-γράμμα.
Να σχεδιαστεί αλγόριθμος που να διαβάζει μία λέξη και να εμφανίζει για κάθε γράμμα το αντίστοιχο ζεύγος αριθμών (πυρσών).
Σημείωση: Ο πίνακας των γραμμάτων είναι:

		ΣΤΗΛΗ				
		1	2	3	4	5
ΓΡΑΜΜΗ	1	Α	Β	Γ	Δ	Ε
	2	Ζ	Η	Θ	Ι	Κ
	3	Λ	Μ	Ν	Ξ	Ο
	4	Π	Ρ	Σ	Τ	Υ
	5	Φ	Χ	Ψ	Ω	

ΔΕ4. Στο βιβλίο αναφέρονται διάφοροι τρόποι δημιουργίας ψηφίου ελέγχου. Κατασκεύασε πρόγραμμα που θα παράγει το ψηφίο ελέγχου ενός οκταψήφιου αριθμού. Δοκίμασε διαφορετικές μεθόδους. Δοκίμασε και διαλέξτε αυτήν τη λύση που παράγει αξιόπιστο ψηφίο ελέγχου.

ΔΕ5. Το ρωμαϊκό σύστημα αρίθμησης χρησιμοποιεί όπως είναι γνωστό χαρακτήρες που αντιστοιχούν σε αριθμούς. Οι χαρακτήρες του ρωμαϊκού συστήματος και η αντίστοιχη αξία τους είναι η εξής:

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

Ζητείται να κατασκευαστεί πρόγραμμα το οποίο, θα δέχεται έναν ρωμαϊκό αριθμό, και θα εμφανίζει τον αντίστοιχό του αραβικό στην οθόνη.

Διάβασε τον ρωμαϊκό αριθμό μέσα σε ένα πίνακα, κάθε ψηφίο σε μία θέση πίνακα, από τα αριστερά προς τα δεξιά, αφήνοντας τις τελευταίες θέσεις του πίνακα κενές.

R(1)	R(2)	R(3)	R(4)	R(5)	R(6)	R(7)	R(8)	R(9)	R(10)	R(11)	R(12)	
M	M	D	C	C	C	I	I	I	NULL	NULL	NULL	=2803
C	M	X	X	X	V	NULL	NULL	NULL	NULL	NULL	NULL	=935
M	C	D	L	X	X	X	I	I	NULL	NULL	NULL	=1482
M	C	D	X	L	I	I	NULL	NULL	NULL	NULL	NULL	=1442

Κατασκεύασε ένα VTOC.

Οι κανόνες που υπολογίζουν την αξία του ρωμαϊκού αριθμού είναι:

I. Αν το γράμμα που προηγείται έχει μικρότερη αξία από το επόμενο του τότε η αξία του αφαιρείται. Πχ. CD=500-100=400, IX=10-1=9, XLV=+5+50-10=45

II. Αν το γράμμα που προηγείται έχει μεγαλύτερη αξία από το επόμενο του τότε η αξία του προστίθεται. Πχ. DC=500+100=600, LXV =+5+10+50=65.

Προσοχή: Οι παρακάτω έλεγχοι θα εξασφαλίσουν, την αξιόπιστη λειτουργία του προγράμματος.

I. Δεν επιτρέπονται άλλοι χαρακτήρες εκτός από τους: I, V, X, L, C, D, M.

II. Δεν επιτρέπονται κενά ανάμεσα στους χαρακτήρες.

III. Δεν επιτρέπεται να υπάρχουν 2 συνεχόμενα σύμβολα ίσα, εκτός αν είναι τα I, X, C, M, οπότε δεν μπορεί να είναι παραπάνω από 3. Πχ. όχι LLV=+5+50+50=105 αλλά CV=+5+100=105, όχι XXXX=10+10+10+10=40 αλλά XL=+50-10=40, σωστό XXX=+10+10+10.

IV. Δεν επιτρέπεται να υπάρχουν 2 συνεχόμενα ίσα σύμβολα, αν ακολουθεί μεγαλύτερης αξίας σύμβολο. Πχ. IIX=+10-1-1=8 λάθος, VIII =1+1+1+5=8 σωστό, IM=999 σωστό. IIC=+100-1-1=98 λάθος, XCVIII=+1+1+1+5+100-10=98 σωστό.

V. Δεν επιτρέπεται να υπάρχει δύο φορές το ίδιο σύμβολο και να χωρίζεται από ένα μόνο άλλο σύμβολο.
Πχ. VLVM=1000-5+50-5=1040 λάθος, MXL=+50-10+1000=1040 σωστό, CMXCVIII=+1+1+1+5+100-10+1000-100=998 σωστό.

VI. Δεν επιτρέπεται να υπάρχουν συνεχόμενα, 2 διαφορετικά σύμβολα μικρότερα από το επόμενό τους.
Πχ. LVM=1000-5+50=1045 λάθος, MXLV=5+50-10+1000=1045 σωστό, XLMVC=100-5+1000-50-10=945 λάθος, VLM=1000-50-5=945 λάθος, CMVL=50-5+1000-100=945 σωστό.

Δεδομένα ελέγχου: MMDCCCIII = 2803, CMXXXV = 935, MCDLXXXII = 1482, MCDXLII = 1442, και όλα τα παραπάνω.

14.5. Τεστ Αυτοαξιολόγησης

1. Η μετατροπή μιας δραχμικής αξίας σε αξία euro που γίνεται στις Τράπεζες είναι πρόγραμμα ή εφαρμογή.

2. Η διάρκεια ζωής ενός προγράμματος αυξάνεται όταν, διαθέσεις περισσότερο χρόνο στη φάση:

   Α) της κωδικοποίησης

   Β) των ελέγχων

   Γ) της αρχικής σχεδίασης

3. Με ποιους τρόπους μπορείτε να αυξήσεις την ταχύτητα του προγράμματός σου:

   Α) με έναν ταχύτερο υπολογιστή
   Β) με έρευνα επί των δεδομένων
   Γ) με ένα γρήγορο πρόγραμμα
   Δ) με επινόηση άλλης τεχνικής

4. 'Οταν επιλέγεις ευφυή λύση στο πρόβλημά σου, τι πρέπει να προσέχεις:

   Α) Την πολύ καλή τεκμηρίωση.
   Β) Την πολύ καλή κωδικοποίηση.
   Γ) Τους εξονυχιστικούς ελέγχους.
   Δ) Να μην σου κλέψουν την ιδέα.

5. 'Οταν σχεδιάζεις τη λύση ενός προβλήματος, τι πρέπει να προσέχεις περισσότερο:

   Α) Την ταχύτητα.
   Β) Την αξιοπιστία.
   Γ) Τη φιλικότητα.
   Δ) Την ευελιξία.
   Ε) 'Ολα τα παραπάνω.

6. Δίνεται ο εξαψήφιος κωδικός 123456, ποιο θα είναι το έβδομο ψηφίο ελέγχου που παράγεται με τη μέθοδο των συνεχών προσθέσεων (διέγραψε το σωστό):

   1, 2, 3, 4, 5, 6, 7, 8, 9, 0.

7. Στο 1ο παράδειγμα " Σύνολα_Είδους" ποιες εντολές περιέχει το καθένα από τα δύο τμήματα της αλλαγής αριθμού (τμήμα κύριας επεξεργασίας –Ε-, τμήμα αρχικών τιμών –Α-). Σημείωσε ένα Ε ή ένα Α αριστερά από την εντολή.

__ `ÃÑÁØÅ Ðñïçã_Á, S`
__ `GS <- GS+S`
__ `Ðñïçã_Á <-Á (1,i)`
__ `S <- 0`

8. Ποια χαρακτηριστικά πρέπει να έχει μια ενότητα. (Σωστό, Λάθος)

   Α) Ο βαθμός ανεξαρτησίας καθορίζει το αν και πόσο οι αλλαγές σε μία ενότητα, επιβάλλουν αλλαγές σε άλλες ενότητες.

   Β) Κάθε ενότητα έχει μία είσοδο, μία έξοδο και εκτελεί μία καθορισμένη επεξεργασία (κανόνας 3ε, είσοδος, επεξεργασία, έξοδος).

   Γ) Κάθε ενότητα καλείται από μία άλλη ενότητα και επιστρέφει σ' αυτήν όταν τελειώσει.

   Δ) Κάθε ενότητα αποτελείται από τις βασικές δομές, ακολουθία, επιλογή, επανάληψη.

   Ε) Σε κάθε ενότητα δεν πρέπει να υπάρχει απότομη διακοπή ή άπειρη επανάληψη.

9. Στα προγράμματα ελέγχου ημερομηνιών πρέπει να προσέχεις ώστε:

   Α) Να δίνεται η πληροφορία αν πρόκειται για ελληνική ημερομηνία ή ημερομηνία ευρωπαϊκού κράτους.

   Β) Το έτος να δίνεται πάντα διψήφιο.

Γ) Να ελέγχεται, αν το έτος είναι δίσεκτο.

Δ) Αν δίνονται αρνητικοί αριθμοί.

10. Συντήρηση ενός προγράμματος λέμε:

Α) Την αναζήτηση λαθών.

Β) Την προσθήκη νέων λειτουργιών.

Γ) Τη σύνταξη και ολοκλήρωση της τεκμηρίωσης.

11. Η τεκμηρίωση ενός προγράμματος γίνεται γιατί:

Α) Πρέπει.

Β) Χωρίς αυτήν η συντήρηση του προγράμματος θα είναι μια περιπέτεια με άγνωστο τέλος.

Γ) Χωρίς αυτήν δεν "τρέχει" το πρόγραμμα.

Δ) Χωρίς αυτήν ο εντοπισμός της αιτίας των λαθών είναι δύσκολος.

Ε) Ξεχνάς.