

ΘΕΜΑ 1°

1.
 1. Σωστό
 2. Λάθος, μπορούμε να σταματήσουμε μόλις εντοπίσουμε μία τιμή που ξεπερνά την αναζητούμενη
 3. Λάθος, δε μπορεί να εκτελεστεί για το πρώτο κελί
 4. Λάθος, μπορεί να υπολογιστεί καθώς δίνονται οι τιμές
 5. Λάθος, πρέπει να έχει 1000 κελιά, γιατί μπορεί ο A να έχει 1000 τιμές που τελειώνουν σε 0
2.
 1. Ο πηγαίος κώδικας μπορεί να μεταφραστεί σε κώδικα μηχανής είτε από τον **μεταγλωττιστή** που θα παράξει το αντικείμενο πρόγραμμα είτε από τον **διερμηνευτή** που θα τον μεταφράζει και παράλληλα θα τον εκτελεί.
 2. Οι γλώσσες γενικά χωρίζονται στις **φυσικές** που καλύπτουν μια ποικιλία αναγκών και στις **τεχνητές**, των οποίων η χρήση είναι πιο συγκεκριμένη.
 3. Στα πλαίσια της γραμματικής περιλαμβάνονται οι **τυπικοί** κανόνες που προσδιορίζουν τις διάφορες επιτρεπτές μορφές μιας λέξης και οι **συντακτικοί** κανόνες, που καθορίζουν τη σειρά που πρέπει να τοποθετούνται οι λέξεις.
 4. Η ύπαρξη **συντακτικών** σφαλμάτων οδηγεί στην αδυναμία μετάφρασης του κώδικα, ενώ η ύπαρξη **λογικών** σφαλμάτων αφήνει τον κώδικα να μεταφραστεί και να εκτελεστεί, αλλά οδηγεί σε λανθασμένη λειτουργία του προγράμματος.
 5. Η **ιεραρχική** σχεδίαση και ο **τμηματικός** προγραμματισμός στην πράξη υλοποιούνται μέσα από την εφαρμογή του δομημένου προγραμματισμού.
3.
 1. Η υπόθεση «**αλλιώς_αν A[middle] < value τότε**» θα γίνει «**αλλιώς_αν A[middle] > value τότε**»
 2. Η υπόθεση «**αν A[middle] = value τότε**» θα γίνει «**αν A[middle] >= x1 και A[middle] <= x1 τότε**»
 3. Στο τέλος της αναζήτησης θα προστεθούν οι εντολές:
αν θ = 0 τότε
 γράψε 'δεν υπάρχει η αναζητούμενη τιμή'
αλλιώς_αν θ = 1 και A[2] = value τότε
 γράψε 'υπάρχει 2 φορές (στην αρχή του πίνακα)'
αλλιώς_αν θ = N και A[N-1] = value τότε
 γράψε 'υπάρχει 2 φορές (στο τέλος του πίνακα)'
αλλιώς_αν A[θ-1] = value ή A[θ+1] = value τότε
 γράψε 'υπάρχει 2 φορές (κάπου μέσα στον πίνακα)'
αλλιώς
 γράψε 'υπάρχει 1 φορά στο κελί', θ
τέλος_αν
4.
 1. Ο δομημένος προγραμματισμός μας υπαγορεύει να γράφουμε τον κώδικά μας βασιζόμενοι στη δομή ακολουθίας, τη δομή επιλογής, και τη δομή επανάληψης, ενώ μας απαγορεύει να χρησιμοποιούμε εντολές goto.
 2. Πίνακας είναι μία στατική δομή δεδομένων με δεδομένα του ίδιου τύπου, που όλα συμβολίζονται με το ίδιο όνομα, αλλά τα ξεχωρίζουμε μεταξύ τους με τη βοήθεια κάποιου αριθμητικού δείκτη. Η χρήση πίνακα είναι απαραίτητη όταν πρέπει να διατηρηθεί στη μνήμη ένας όγκος δεδομένων ώστε να τα ξαναεπεργαστούμε αργότερα.
 3. Παράλληλοι ονομάζονται δύο πίνακες που έχουν μια διάσταση κοινή, ίδιο αριθμό κελιών, και μεταξύ των κελιών τους παρουσιάζεται μία λογική συσχέτιση. Πχ ένας πίνακας με τα ονόματα των 52 νομών της Ελλάδας κι ένας πίνακας με τα ονόματα των 52 πρωτευουσών είναι παράλληλοι, εφ' όσον για κάθε κελί χ ισχύει ότι η πρωτεύουσα στο κελί χ του ενός πίνακα είναι πρωτεύουσα του νομού που είναι στο κελί χ του άλλου πίνακα.
 4. Σε ένα ταξινομημένο σύνολο στοιχείων εντοπίζουμε το μεσαίο και το συγκρίνουμε με την αναζητούμενη τιμή. Αν είναι ίσα σταματάμε επιτυχώς την αναζήτηση, αλλιώς κρατάμε το υποσύνολο που είναι δεξιά ή αριστερά από το μεσαίο (ανάλογα με το αν τα στοιχεία είναι σε αύξουσα ή φθίνουσα σειρά) και επαναλαμβάνουμε από την αρχή.

5. Ο συνδέτης-φορτωτής μετατρέπει το αντικείμενο πρόγραμμα στο τελικό εκτελέσιμο πρόγραμμα, ενσωματώνοντας κώδικα από τις βιβλιοθήκες.

ΘΕΜΑ 2°

1.	α) Σωστό δ) Λάθος, γιατί βάζει στις ζυγές θέσεις τα στοιχεία του A και στις μονές τα στοιχεία του B για χ από 2 μέχρι 16 με βήμα 2 Γ[χ-1] ← A[χ div 2] Γ[χ] ← B[χ div 2] τέλος_επανάληψης	β) Σωστό ε) Λάθος γιατί κάνει 15 επαναλήψεις, και ξαναγραφει τιμές στα ίδια κελιά του Γ ψ ← 1 για χ από 2 μέχρι 16 με βήμα 2 Γ[χ-1] ← A[ψ] Γ[χ] ← B[ψ] ψ ← ψ + 1 τέλος_επανάληψης	γ) Σωστό στ) Λάθος γιατί ξαναγράφει τιμές στα ίδια κελιά του Γ, μέχρι το Γ[9] μόνο ψ ← 1 για χ από 1 μέχρι 8 Γ[ψ] ← A[χ] Γ[ψ+1] ← B[χ] ψ ← ψ + 2 τέλος_επανάληψης			
2.	α	β	Σ	β>0	Σ>=β	οθόνη
	1					
		16				
				αληθής		
			0			
			6			
					ψευδής	
			12			
					ψευδής	
			18			
					αληθής	
17						
	15					
						17 18
				αληθής		
			0			
			5			
					ψευδής	
			10			
					ψευδής	
			15			
					ψευδής	
			20			
					αληθής	
32						
	-2					
						32 20

ΘΕΜΑ 3°

Η Γιαγιά-Γιούλα παρακολουθεί φανατικά στα social τον Άκη Πετρετζίκη και περνά το χρόνο της δοκιμάζοντας κάθε μέρα και μία από τις συνταγές του. Γράψτε πρόγραμμα το οποίο:

1. Θα περιλαμβάνει τμήμα δήλωσης μεταβλητών. (03/20)
2. Θα καταγράφει σε τρεις πίνακες τρία στοιχεία για κάθε συνταγή που δοκίμασε η Γιαγιά-Γιούλα κατά τη διάρκεια του Δεκέμβρη (δηλαδή για 31 ημέρες): τον τίτλο της, πόσα χρήματα κόστισε και αν πέτυχε ή όχι (θεωρούμε ότι καταχωρούνται οι τιμές «ναι» ή «όχι»). Όποτε δοθεί τίτλος που έχει ξαναδοθεί, να εμφανίζεται το μήνυμα «συγγνώμη παιδάκι μου, είμαι γριά γυναίκα» και να μη γίνεται δεκτός. (08/20)
3. Θα εμφανίζει τους τίτλους είτε των επιτυχημένων συνταγών σε αύξουσα σειρά κόστους, είτε των αποτυχημένων συνταγών σε φθίνουσα σειρά κόστους, ανάλογα με το ποιās κατηγορίας το μέσο χρηματικό κόστος είναι μεγαλύτερο (θεωρούμε ότι υπάρχουν και επιτυχημένες και αποτυχημένες). Δίπλα σε κάθε τίτλο θα πρέπει να εμφανίζεται και ποιά μέρα του Δεκέμβρη εκτελέστηκε η συνταγή. (09/20)

πρόγραμμα θ3
μεταβλητές
ακέραιες: α[31], αα, χ, μ, π1, π2
πραγματικές: κ[31], κκ, σ1, σ2
χαρακτήρες: τ[31], ττ, π[31], ππ
λογικές: ξαναδόθηκε
αρχή

γράψε 'τίτλος; κόστος; πέτυχε;'
διάβασε τ[1], κ[1], π[1]

α[1] ← 1

για χ από 2 μέχρι 31

α[χ] ← χ

γράψε 'τίτλος; κόστος; πέτυχε;'

διάβασε τ[χ], κ[χ], π[χ]

αρχή_επανάληψης

μ ← 0

ξαναδόθηκε ← ψευδής

αρχή_επανάληψης

μ ← μ + 1

ξαναδόθηκε ← τ[χ] = τ[μ]

μέχρις_ότου ξαναδόθηκε = αληθής ή μ = χ-1

αν ξαναδόθηκε = αληθής τότε

γράψε 'συγγνώμη παιδάκι μου, είμαι γριά γυναίκα'

διάβασε τ[χ]

τέλος_αν

μέχρις_ότου ξαναδόθηκε = ψευδής

τέλος_επανάληψης

π1 ← 0

π2 ← 0

σ1 ← 0

σ2 ← 0

για χ από 1 μέχρι 31

αν π[χ] = 'ναι' τότε

π1 ← π1 + 1

σ1 ← σ1 + κ[χ]

αλλιώς

π2 ← π2 + 1

σ2 ← σ2 + κ[χ]

τέλος_αν

τέλος_επανάληψης

για μ από 2 μέχρι 31

για χ από 31 μέχρι μ με βήμα -1

αν κ[χ] < κ[χ-1] τότε

κκ ← κ[χ]

κ[χ] ← κ[χ-1]

κ[χ-1] ← κκ

ππ ← π[χ]

π[χ] ← π[χ-1]

π[χ-1] ← ππ

ττ ← τ[χ]

τ[χ] ← τ[χ-1]

τ[χ-1] ← ττ

αα ← α[χ]

α[χ] ← α[χ-1]

α[χ-1] ← αα

τέλος_αν

τέλος_επανάληψης

τέλος_επανάληψης

! τα στοιχεία της πρώτης συνταγής χωριστά

! για κάθε άλλη συνταγή

! αναζητάω αν ο τίτλος που δόθηκε τώρα

! υπάρχει καταχωρημένος στους προηγούμενους

! τίτλος

! μετρώ πλήθος επιτυχημένων συνταγών και

! αποτυχημένων, και τα αντίστοιχα συνολικά

! τους κόστη

! μετά ταξινομώ τα πάντα σε αύξουσα σειρά

! κόστους

! στο τέλος, ανάλογα με το ποιός μέσος όρος

! είναι μεγαλύτερος, εμφανίζω είτε τις

! επιτυχημένες συνταγές με τη σειρά που είναι

! στον πίνακα (δηλαδή αύξουσα σειρά κόστους),

! είτε τις αποτυχημένες με την ανάποδη σειρά

! που είναι στον πίνακα (δηλαδή φθίνουσα σειρά

! κόστους)

αν σ1/π1 > σ2/π2 τότε

για χ από 1 μέχρι 31

αν π[χ] = 'ναι' τότε

γράψε τ[χ], α[χ]

τέλος_αν

τέλος_επανάληψης

αλλιώς

για χ από 31 μέχρι 1 με βήμα -1

αν π[χ] = 'όχι' τότε

γράψε τ[χ], α[χ]

τέλος_αν

τέλος_επανάληψης

τέλος_αν

τέλος_προγράμματος

ΘΕΜΑ 4°

Η Γιαγιά-Γιούλα περνά και τα Χριστούγεννα εν μέσω lockdown, όμως έχει αποφασίσει ότι θα παίρνει τηλέφωνο και θα επικοινωνεί με φίλους και συγγενείς, για να μη νοιώθει μόνη της. Κάποιους μπορεί να τους πάρει και να τους ξαναπάρει πολλές φορές, ενώ κάποιους καμία. Τη μία φορά που θα πάρει κάποιον, μπορεί να του μιλήσει 10 λεπτά, ενώ την άλλη φορά που θα τον ξαναπάρει, μπορεί να τού μιλήσει 25 λεπτά. Γράψτε πρόγραμμα το οποίο:

1. Θα περιλαμβάνει τμήμα δήλωσης μεταβλητών. (02/20)
2. Θα διαβάζει και θα εισάγει σε πίνακα τα 77 τηλέφωνα που έχει στον κατάλογό της η γιαγιά-Γιούλα, ενώ σε άλλο παράλληλο πίνακα θα σημειώνεται είτε η ένδειξη «Φ» είτε η ένδειξη «Σ» διενεργώντας τον σχετικό έλεγχο εγκυρότητας. (η τιμή «Φ» σημαίνει «φίλος», ενώ η τιμή «Σ» σημαίνει «συγγενής») (03/20)
3. Θα ρωτάει την γιαγιά-Γιούλα ποιό τηλέφωνο θέλει να πάρει, και θα το αναζητά στον πίνακα με τα τηλέφωνα της. Αν βρεθεί, θα ρωτάει ποιά είναι η διάρκεια του τηλεφωνήματος σε λεπτά. Αν δε βρεθεί, θα εμφανίζει το μήνυμα «Λάθος αριθμός τηλεφώνου». Όταν βγαίνει λάθος αριθμός τηλεφώνου για δεύτερη συνεχόμενη φορά, να εμφανίζει το μήνυμα «συγγνώμη παιδάκι μου, είμαι γριά γυναίκα». Αυτά θα πρέπει να συνεχίζονται μέχρι η γιαγιά-Γιούλα να έχει μιλήσει στο τηλέφωνο με καθέναν από τους φίλους της για τουλάχιστον 45 λεπτά. (09/20)

Όταν ολοκληρωθούν τα προηγούμενα, θα εμφανίζει

4. ένα μήνυμα που θα λέει ποιά είναι το ποσοστό των συγγενών με τους οποίους δεν επικοινωνήσε καθόλου. (03/20)
5. για τα 5 τελευταία τηλέφωνα που πήρε η Γιαγιά-Γιούλα, πόσες φορές τηλεφώνησε στο καθένα. (03/20)

πρόγραμμα θ4

μεταβλητές

ακέραιες: πφ, χ, τ[77], λ[77], φ[77], κ, συν_λαθ, οκ, π[5], τηλ, λεπ, οξω

χαρακτήρες: α[77],

αρχή

πφ ← 0 ! πλήθος φίλων

για χ από 1 μέχρι 77

γράψε 'τηλέφωνο;'

διάβασε τ[χ]

αρχή_επανάληψης

γράψε 'φίλος/συγγενής;'

διάβασε α[χ]

μέχρις_ότου α[χ] = 'Φ' ή α[χ] = 'Σ'

αν α[χ] = 'Φ' τότε

πφ ← πφ + 1

τέλος_αν

λ[χ] ← 0

! συνολικά λεπτά ομιλίας με το άτομο χ

φ[χ] ← 0

! φορές επικοινωνίας με το άτομο χ

τέλος_επανάληψης

κ ← 0

! μετρητής επαναλήψεων

συν_λαθ ← 0

! συνεχόμενα λάθη

οκ ← 0

! με πόσους φίλους έχει μιλήσει τουλάχιστον 45 λεπτά

αρχή_επανάληψης

γράψε 'ποιό τηλέφωνο παίρνεις;'

διάβασε τηλ

χ ← 0

αρχή_επανάληψης

χ ← χ + 1

μέχρις_ότου τ[χ] = τηλ ή χ = 77

αν τ[χ] = τηλ τότε

γράψε 'πόσα λεπτά μίλησες;'

διάβασε λεπ

συν_λαθ ← 0

αν λ[χ] < 45 και λ[χ] + λεπ >= 45 και α[χ] = 'Φ' τότε

οκ ← οκ + 1

τέλος_αν

λ[χ] ← λ[χ] + λεπ

κ ← κ + 1

π[(κ-1) mod 5 + 1] ← χ ! πίνακας για τα τελευταία 5 τηλέφωνα

φ[χ] ← φ[χ] + 1

αλλιώς

γράψε 'λάθος αριθμός τηλεφώνου'

συν_λαθ ← συν_λαθ + 1

αν συν_λαθ = 2 τότε

γράψε 'συγγνώμη παιδάκι μου, είμαι γριά γυναίκα'

τέλος_αν

τέλος_αν

μέχρις_ότου οκ = πφ

οξω ← 0

! πλήθος συγγενών που δεν επικοινωνήσε

για χ από 1 μέχρι 77

αν λ[χ] = 0 και α[χ] = 'Σ' τότε

οξω ← οξω + 1

τέλος_αν

τέλος_επανάληψης

γράψε οξω / (77-πφ) * 100, '% των συγγενών της δεν τους τηλεφώνησε καθόλου'

για κ από 1 μέχρι 5

γράψε φ[π[κ]], 'φορές τηλεφώνησε στο', τ[π[κ]]

τέλος_επανάληψης

τέλος_προγράμματος