
ΘΕΜΑ Α

A1. Σωστό-Λάθος**(07/25)**

1. Η αντικειμενοστρεφής σχεδίαση εστιάζει στα δεδομένα, σε αντίθεση με τις παραδοσιακές τεχνικές που εστιάζουν στις ενέργειες.
2. Οι συντακτικοί κανόνες είναι εκείνοι που καθορίζουν τους αποδεκτούς τύπους των λέξεων σε μια γλώσσα.
3. Η εκσφαλμάτωση με τη μέθοδο του Μαύρου Κουτιού προϋποθέτει να έχουμε στα χέρια μας τον κώδικα που προσπαθούμε να διορθώσουμε.
4. Όταν υπάρχει ένας ατέρμων βρόχος, παρουσιάζεται σφάλμα χρόνου εκτέλεσης.
5. Βασιζόμενοι στην αφαιρετικότητα μπορούμε να σχεδιάσουμε υποκλάσεις που να διατηρούν τα στοιχεία των υπερκλάσεων, αλλά και να τα επεκτείνουν ανάλογα με τις ανάγκες μας.
6. Το προγραμματιστικό περιβάλλον προϋποθέτει να γνωρίζουμε μια γλώσσα μηχανής.
7. Μειονέκτημα του διερμηνευτή είναι ότι χρειάζεται να μεταφράζει τον πηγαίο κώδικα κάθε φορά από την αρχή.

A2. Συμπληρώστε τα κενά με τον κατάλληλο όρο**(08/25)**

1. Οι ___ γλώσσες χαρακτηρίζονται από στασιμότητα.
2. Η ιεραρχική σχεδίαση αλλιώς ονομάζεται και σχεδίαση ___.
3. Ο ειδικός επεξεργαστής κειμένου που βοηθάει τον προγραμματιστή στη δημιουργία του πηγαίου κώδικα είναι ο ___.
4. Χαρακτηριστικό του ___ είναι η άμεση μετάφραση και εκτέλεση του πηγαίου κώδικα.
5. Για την εκσφαλμάτωση χρειάζεται οπωσδήποτε να σχεδιάσουμε σενάρια ελέγχου για τις ακραίες τιμές κάθε ___ διαστήματος.
6. Πολυμορφισμός ονομάζεται η δυνατότητα μια κλάση να υλοποιεί μία ___ διαφορετικά από μία άλλη κλάση.
7. Ο συνδέτης ενσωματώνει στον δικό μας κώδικα, κώδικα που βρίσκεται έτοιμος στις ___.
8. Κεφαλή ονομάζεται ο ___ που δηλώνει σε ποιά διεύθυνση μνήμης είναι ο πρώτος κόμβος μιας λίστας.

A3.1. Αναφέρετε τα πλεονεκτήματα του δομημένου προγραμματισμού.**(03/25)****A3.2. Αναφέρετε τα πλεονεκτήματα της αντικειμενοστρεφούς σχεδίασης.****(03/25)****A3.3. Αναφέρετε τα στοιχεία από τα οποία συντίθεται μια γλώσσα.****(04/25)**

ΘΕΜΑ Β

B1.

1. πρόγραμμα B1
2. μεταβλητές
3. πραγματικές: $HL[1833]$
4. ακέραιες: $x, \psi, \Pi[1833], a, \alpha\theta\rho$
5. αρχή
6. για x από 1 μέχρι 1833
7. διάβασε $HL[x] \ \Pi[x]$
8. τέλος_επανάληψης
9. για ψ από 2 μέχρι 101
10. για x από 1833 μέχρι ψ με βήμα -1
11. αν $\Pi[x] < \Pi[x-1]$ τότε
12. $a \leftarrow HL[x]$
13. $HL[x] \leftarrow HL[x-1]$
14. $HL[x-1] \leftarrow a$
15. τέλος_αν
16. τέλος_επανάληψης
17. τέλος_επανάληψης
18. για ψ από 1 μέχρι x
19. $\alpha\theta\rho \leftarrow \alpha\theta\rho + \Pi[\psi]$
20. τέλος_επανάληψης
21. γράψε $\alpha\theta\rho$
22. τέλος_προγράμματος

Για τις 1833 μητέρες που συμμετείχαν σε μία έρευνα, καταγράφεται η ηλικία τους και ο αριθμός των παιδιών τους. Ζητούμενο είναι να εμφανιστεί ο συνολικός αριθμός των παιδιών των 100 μικρότερων σε ηλικία από αυτές. Θεωρούμε ότι κάθε μητέρα έχει διαφορετική ημερομηνία γέννησης. Για την επίλυση αυτού του προβλήματος προτάθηκε ο διπλανός κώδικας, ο οποίος όμως περιέχει 5 σφάλματα. Για κάθε ένα από αυτά τα σφάλματα

- αναφέρετε πού το εντοπίζετε,
- περιγράψτε το με λόγια
- αναφέρετε τι σφάλμα είναι (συντακτικό, λογικό, αντικανονικού τερματισμού)

(15/25)

B2. «Στη ΓΛΩΣΣΑ υπάρχουν κατηγορίες **μεταβλητών** όπως **λογικές, χαρακτήρες και αριθμητικές**. Οι αριθμητικές μεταβλητές επιπλέον χωρίζονται σε **ακέραιες** και **πραγματικές**. Κάθε μεταβλητή έχει ένα **όνομα**, κάποια **τιμή**, και μια **διεύθυνση** στη μνήμη του ΗΥ. Σε όλες τις μεταβλητές μπορεί να γίνει **εκχώρηση** τιμής, αλλά και **εμφάνιση** της τιμής τους. Με όλα τα είδη μεταβλητών μπορεί να γίνει **σύγκριση**, αν και γίνεται με άλλο τρόπο σε κάθε μία από τις τρεις βασικές κατηγορίες μεταβλητών. Στις αριθμητικές μεταβλητές μπορεί επιπλέον να γίνει **πρόσθεση**, ενώ στις λογικές μεταβλητές συναντάμε τις πράξεις της **άρνησης**, της **σύζευξης** και της **διάζευξης**».

Με βάση την παραπάνω περιγραφή, φτιάξτε ένα διάγραμμα κλάσεων στο οποίο θα φαίνονται κλάσεις, ιδιότητες, μέθοδοι και σχέσεις ιεραρχίας μεταξύ των κλάσεων. Στο διάγραμμά σας θα συμπεριλάβετε -στις κατάλληλες θέσεις- μόνο τις ακόλουθες λέξεις:

(10/25)
ακέραια, αριθμητική, άρνηση, διάζευξη, διεύθυνση, εκχώρηση, εμφάνιση, λογική, μεταβλητή, όνομα, πραγματική, πρόσθεση, σύγκριση, σύζευξη, τιμή, χαρακτήρας

ΘΕΜΑ Γ

Γ1. Γράψτε **συνάρτηση πλήθος(A, j, x)**: **ακέραια** όπου A πίνακας πραγματικών αριθμών με διαστάσεις 18x44, j ο αριθμός μίας στήλης του πίνακα A και x ένας πραγματικός αριθμός. Η συνάρτηση θα υπολογίζει και θα επιστρέφει πόσα στοιχεία της στήλης j του πίνακα A έχουν τιμή μικρότερη-ίση του x.

Σε ένα αγώνα Formula1 που περιλαμβάνει 44 γύρους, συμμετείχαν 18 οδηγοί και οι χρόνοι τους καταγράφονται σε δευτερόλεπτα, με ακρίβεια χιλιοστού. Να γράψετε πρόγραμμα όπου:

Γ2. θα υπάρχει συμπληρωμένο τμήμα δήλωσης μεταβλητών (03/25)

Γ3α. θα διαβάζεται πίνακας N[18] με τα ονόματα των οδηγών (02/25)

Γ3β. θα διαβάζεται για κάθε οδηγό και για κάθε γύρο πόσα δευτερόλεπτα χρειάστηκε για την ολοκλήρωσή του. Σε πίνακα XP[18,44] και σε κάθε κελί XP[i,j] εσείς πρέπει να αποθηκεύσετε τον **συνολικό** χρόνος του οδηγού i για την ολοκλήρωση των γύρων από 1 έως j.

Πχ αν υποθέσουμε ότι για τον 12ο οδηγό οι πρώτες τρεις τιμές που διαβάζονται είναι 67.5", 62.9", 66.4", αυτό θα σημαίνει ότι σε 67,5" ολοκλήρωσε τον 1ο γύρο, σε 62.9" ολοκλήρωσε τον 2ο γύρο και σε 66.4" ολοκλήρωσε τον 3ο γύρο... εσείς θα πρέπει να καταχωρείτε στο XP[12,1] η τιμή 67.5, στο XP[12,2] η τιμή 130.4, και στο XP[12,3] η τιμή 196.8"

(05/20)
Γ4. θα διαβάζεται ένα όνομα, το οποίο θα αναζητείται στον πίνακα N. Εφ' όσον βρεθεί, με τη βοήθεια της συνάρτησης που φτιάξατε νωρίτερα θα εμφανίζεται η θέση του οδηγού στην κατάταξη, σε καθένα από τους 44 γύρους (θα λέει πχ ότι ήταν 5ος στον 1ο γύρο, 8ος στο 2ο γύρο, 17ος στον 3ο γύρο...). Η κατάταξη των οδηγών στο γύρο j γίνεται με βάση το ποιος χρειάστηκε το μικρότερο συνολικό χρόνο για να ολοκληρώσει τους γύρους από 1 έως j. Εφ' όσον το όνομα δε βρεθεί, θα εμφανίζεται ανάλογο μήνυμα.

(07/25)
Γ5. η λειτουργία του ερωτήματος Γ4 θα επαναλαμβάνεται, ώσπου να διαβαστεί το όνομα του οδηγού που βγήκε 18ος στην κατάταξη του 44ου γύρου (03/25)

ΘΕΜΑ Δ

- Θεωρήστε ότι υπάρχει η **διαδικασία αρχικοποίηση(A, top)**, όπου A πίνακας ακεραίων 128 θέσεων που χρησιμεύει για την υλοποίηση μίας στοίβας, και top ο αντίστοιχος δείκτης διαχείρισης της στοίβας. Η διαδικασία επιστρέφει μία άδεια στοίβα.

- Θεωρήστε ότι υπάρχει η **διαδικασία ώθηση(A,top,x)** όπου A πίνακας ακεραίων 128 θέσεων που χρησιμεύει για την υλοποίηση μίας στοίβας, top ο αντίστοιχος δείκτης διαχείρισης της στοίβας και x η ακέραια τιμή που επιθυμούμε να ωθήσουμε στη στοίβα. Εάν υπάρχει χώρος, η τιμή x ωθείται στη στοίβα, ενώ αν δεν υπάρχει χώρος εμφανίζει μήνυμα υπερχειλίσης και η μεταβλητή x επιστρέφεται με την τιμή -1.

Δ1. γράψτε **διαδικασία απώθηση(A,top,x)** όπου A πίνακας ακεραίων 128 θέσεων που χρησιμεύει για την υλοποίηση μίας στοίβας και top ο αντίστοιχος δείκτης διαχείρισης της στοίβας. Εάν η στοίβα δεν είναι άδεια, θα απωθείται μια τιμή από τη στοίβα και θα επιστρέφεται μέσω της μεταβλητής x, ενώ αν είναι άδεια εμφανίζει μήνυμα υποχειλίσης και η μεταβλητή x θα επιστρέφεται με την τιμή -1.

(06/25)

Χωρίς να κάνετε καμία απ' ευθείας εκχώρηση σε πίνακα ή σε κάποιο δείκτη top, αλλά μόνο καλώντας τα προαναφερθέντα υποπρογράμματα, γράψτε πρόγραμμα το οποίο :

Δ2. θα περιέχει συμπληρωμένο τμήμα δηλώσεων **(02/25)**

Δ3. θα κάνει τις απαραίτητες αρχικοποιήσεις **(02/25)**

Δ4.α. θα διαβάζει διαδοχικά ακέραιες τιμές, και θα τερματίζει όταν διαβαστεί η τιμή 0 **(03/25)**

Δ4.β. όποτε διαβαστεί άρτια τιμή, θα επιχειρείται ώθησή της στη στοίβα A1 **(02/25)**

Δ4.γ. όποτε διαβαστεί περιττή τιμή, θα επιχειρείται απώθηση -και εμφάνιση- από τη στοίβα A1 τόσων στοιχείων, όσα και η τιμή που διαβάστηκε. Αν η στοίβα A1 δεν έχει αρκετά στοιχεία, θα απωθούνται μόνο όσα έχει. **(05/25)**

Δ5. Μόλις τερματιστεί η παραπάνω επανάληψη, χωρίς να γίνει καμία επεξεργασία μετά τον τερματισμό, να εμφανίζεται η μεγαλύτερη τιμή που έχει απομείνει στη στοίβα A1. Αυτό θα μπορούσε να το καταφέρετε χρησιμοποιώντας κατάλληλα μία δεύτερη στοίβα A2, στην κορυφή της οποίας θα αποθηκεύετε σε ποιά θέση της στοίβας A1 υπάρχει η τρέχουσα μέγιστη τιμή της A1. Θεωρήστε δεδομένο ότι η A1 δε θα είναι άδεια. **(05/25)**