|  |
| --- |
| **Ονοματεπώνυμο: …………………………………………………………………..** **Μάθημα: ΑΕΠΠ Υλη: ΕΠΑΝΑΛΗΠΤΙΚΟ** **Επιμέλεια διαγωνίσματος: ΛΑΜΠΡΑΚΗΣ ΜΑΝΩΛΗΣ****Αξιολόγηση : …………………………………………………………………………** |

**ΘΕΜΑ Α**

**Α1.** Να απαντήσετε στις παρακάτω προτάσεις με την λέξη «Σωστό» αν θεωρείτε την πρόταση Σωστή ή «Λάθος» αν την θεωρείτε Λανθασμένη**.**

1. Ο ανθρώπινος εγκέφαλος είναι ένας μηχανισμός επεξεργασίας δεδομένων.
2. Γενικά, μία γλώσσα προγραμματισμού υψηλού επιπέδου, είναι ταχύτερη από μία γλώσσα προγραμματισμού χαμηλού επιπέδου.
3. Κατά τη διαδικασία της εκσφαλμάτωσης, τα λάθη που μας απασχολούν είναι κυρίως τα συντακτικά. .
4. Ο αντικειμενοστραφής προγραμματισμός έχει το πλεονέκτημα πως τα προγράμματα που δημιουργούνται είναι ευέλικτα και επαναχρησιμοποιούμενα**.**
5. Αν κατά την ανάγνωση μίας αριθμητικής μεταβλητής, ο χρήστης εισαγάγει ένα χαρακτήρα, τότε το πρόγραμμα θα τερματιστεί αντικανονικά.

 **(Μονάδες 5)**

**Α2.** Να απαντήσετε στις ακόλουθες ερωτήσεις:

1. Να αναφέρετε τις στοιχειώδεις ενέργειες που μπορεί να εκτελέσει ένας υπολογιστής. **(Μονάδες 4)**
2. Ποια δομή δεδομένων ονομάζεται «Ουρά» και ποιες είναι οι βασικές λειτουργίες της; **(Μονάδες 3)**
3. Να εξηγήσετε, γιατί ο υπολογιστής χρησιμοποιεί τη μέθοδο του «Πολλαπλασιασμού αλά Ρωσικά» για την υλοποίηση του πολλαπλασιασμού.

 **(Μονάδες 3)**

**Α3.** Ένας μαθητής θέλει να οργανώσει το πρόγραμμα του για τις επαναλήψεις που θα κάνει στο μάθημα των μαθηματικών. Σκέφτηκε λοιπόν να χωρίσει την ύλη αρχικά σε 3 ενότητες: Συναρτήσεις, Διαφορικός Λογισμός και Ολοκληρωτικός Λογισμός. Στις Συναρτήσεις, θα πρέπει να μελετήσει Μονοτονία Συνάρτησης και Όρια Συνάρτησης στα οποία θα πρέπει να μελετήσει Ιδιότητες των Ορίων και τα Όρια Συνάρτησης σο Άπειρο. Στο Διαφορικό Λογισμό θα πρέπει να μελετήσει τις Παραγώγους και το Θεώρημα Μέσης Τιμής. Στον Ολοκληρωτικό Λογισμό, θα πρέπει να μελετήσει το Αόριστο Ολοκλήρωμα και το Ορισμένο Ολοκλήρωμα. Να παρουσιάσετε μία διαγραμματική απεικόνιση του προβλήματος της οργάνωσης της επανάληψης του μαθητή με βάση την παραπάνω περιγραφή. **(Μονάδες 6)**

**Α4.** Να συμπληρώσετε τα κενά, ώστε τα δύο τμήματα αλγορίθμου να εμφανίσουν το ίδιο αποτέλεσμα.

|  |  |
| --- | --- |
| **ΤΜΗΜΑ Α****Διάβασε** κ,λ**Αν** κ=5+λ **τότε** **Γράψε** ‘Α’**Αλλιώς\_αν** κ-λ=2 **Γράψε** ‘B’**Αλλιώς\_αν** 10>λ-κ **τότε** **Γράψε** ‘Γ’**Αλλιώς** **Γράψε** ‘Δ’**Τέλος\_αν** | **ΤΜΗΜΑ Β****Διάβασε** κ,λ**Επίλεξε *ΚΕΝΟ1*** **Περίπτωση** 5 **Γράψε** ‘Α’ **Περίπτωση *ΚΕΝΟ2*** **Γράψε** ‘Β’ **Περίπτωση *ΚΕΝΟ3*** **Γράψε** ‘Γ’ ***ΚΕΝΟ4*** **Γράψε** ‘Δ’ ***ΚΕΝΟ5***  |

**Α5.** Το παρακάτω αριθμημένο πρόγραμμα, έχει ως σκοπό να επιλύσει το ακόλουθο πρόγραμμα: «Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που θα διαβάζει πίνακα Α[10] πραγματικών αριθμών, εξασφαλίζοντας πως θα λάβουν θετικές τιμές και θα εμφανίζει τα στοιχεία που είναι μεγαλύτερα από το ακριβώς προηγούμενο στοιχείο τους». Στο πρόγραμμα υπάρχουν λάθη. Να εντοπίσετε σε ποια γραμμή υπάρχει το κάθε λάθος, να το χαρακτηρίσετε ως προς το είδος του και να και να προτείνετε διορθώσεις.

**1.ΠΡΟΓΡΑΜΜΑ** ΑΣΚ\_Α5

**2.ΜΕΤΑΒΛΗΤΕΣ**

**3. ΠΡΑΓΜΑΤΙΚΕΣ:** Ι, Α[10]

**4. ΑΡΧΗ**

**5. ΓΙΑ** Ι **ΑΠΟ** 1  **ΜΕΧΡΙ** 10

**6. ΔΙΑΒΑΣΕ** Α[Ι]

**7.** **ΟΣΟ** Α[Ι]>0 **ΕΠΑΝΑΛΑΒΕ**

**8.**  **ΔΙΑΒΑΣΕ** Α[Ι]

**9.** **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

10. **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**11. ΓΙΑ** Ι **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

**12.** **ΑΝ** Α[Ι]>=Α[Ι-1] **ΤΟΤΕ**

**13. ΓΡΑΨΕ** Α[Ι]

**14.** **ΤΕΛΟΣ\_ΑΝ**

**15. ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**16.ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ (Μονάδες 8)**

**Α6**. Δίνεται ο ακόλουθος κώδικας για την ταξινόμηση ενός πίνακα Α[6] με την μέθοδο επιλογής (Selection Sort).

|  |
| --- |
|  **Για** λ **από** 1 **μέχρι** 5  κ🡨 λ  **Για** μ **από** λ+1 **μέχρι** 6  **Αν** Α[μ] < Α[κ] **τότε**  κ🡨 μ  **Τέλος**\_**αν**  **Τέλος**\_**επανάληψης**  Τ🡨 Α[λ]  Α[λ]🡨 Α[κ]  Α[κ]🡨 Τ **Τέλος**\_**επανάληψης**  |

Να τον εφαρμόσετε στον ακόλουθο πίνακα, παρουσιάζοντας την μορφή του πίνακα μετά από το τέλος κάθε εξωτερικής επανάληψης:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 52 | 12 | 71 | 56 | 5 | 10 |
|  1 | 2 | 3 | 4 | 5 | 6 |

 **(Μονάδες 6)**

**ΘΕΜΑ Β**

**Β1.** Για αύξουσα διάταξη των στοιχείων ενός πίνακα Α[Ν], μπορούμε να χρησιμοποιήσουμε την ακόλουθη τεχνική («ταξινόμηση με εισαγωγή», είναι ιδανική για «μερικώς» ταξινομημένους πίνακες): κάθε στοιχείο του πίνακα (αρχίζοντας από το δεύτερο μέχρι και το τελευταίο), τοποθετείται στη σωστή σειρά σε σχέση με όλα τα προηγούμενα στοιχεία του. Αρχικά το ελέγχουμε με το προηγούμενο από αυτό στοιχείο και αν είναι μικρότερο τα αντιμεταθέτουμε για να έρθει στη σωστή θέση (διαφορετικά σταματάμε), στη συνέχεια κάνουμε την ίδια διαδικασία με το προ-προηγούμενο από αυτό στοιχείο κτλ. Για παράδειγμα στον παρακάτω πίνακα:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | 10 | 20 | 25 | 15 | 30 |
|  1 | 2 | 3 | 4 | 5 | 6 |

* Αρχίζουμε από το Α[2], το συγκρίνουμε με το Α[1], δε χρειάζεται αλλαγή, σταματάμε.
* Συνεχίζουμε με το Α[3], το συγκρίνουμε με το Α[2], δε χρειάζεται αλλαγή, σταματάμε.
* Συνεχίζουμε με το Α[4], το συγκρίνουμε με το Α[3], δε χρειάζεται αλλαγή, σταματάτε.
* Συνεχίζουμε με το Α[5], το συγκρίνουμε με το Α[4], χρειάζεται αλλαγή, άρα τα αντιμεταθέτουμε (το 15 στο Α[4] και το 25 στο Α[5]). Στη συνέχεια συγκρίνουμε το Α[4] με το Α[3], χρειάζεται αλλαγή, άρα τα αντιμεταθέτουμε (το 15 στο Α[3], το 20 στο Α[4]). Στη συνέχεια συγκρίνουμε το Α[3] με το Α[2], δε χρειάζεται αλλαγή άρα σταματάμε (στη μορφή🡪 5 10 15 20 25 30).
* Συνεχίζουμε με το Α[6], το συγκρίνουμε με το Α[5], δε χρειάζεται αλλαγή, σταματάμε.

Για την υλοποίηση του αλγορίθμου, απαιτούνται δύο επαναλήψεις:

* Η εξωτερική επανάληψη ορίζει το στοιχείο το οποίο θα τοποθετηθεί στη σωστή σειρά.
* Η εσωτερική επανάληψη η οποία είναι υπεύθυνη για τον έλεγχο των στοιχείων και εκτελείται όσο το στοιχείο που ορίζει η εξωτερική επανάληψη δεν έχει τοποθετηθεί στη σωστή σειρά και όσο υπάρχουν ακόμη στοιχεία για σύγκριση.

Να συμπληρώσετε τα κενά στο ακόλουθο τμήμα αλγορίθμου, ώστε να υλοποιείται η ταξινόμηση ενός πίνακα Α[Ν] (του οποίου τα στοιχεία θεωρούμε γνωστά) με βάση την παραπάνω περιγραφή.

**Για** i **από** 2 **μέχρι** Ν

 j ← ***ΚΕΝΟ1***

 έλεγχος ← Αληθής

 **Όσο** j > ***ΚΕΝΟ2*** **και** ***ΚΕΝΟ3*** **επανάλαβε**

 **Αν** Α[***ΚΕΝΟ4***] > Α[***ΚΕΝΟ5***] **τότε**

 **Αντιμετάθεσε** ***ΚΕΝΟ6***, ***ΚΕΝΟ7***

 j ← ***ΚΕΝΟ8***

 **Αλλιώς**

 ***ΚΕΝΟ9*** ← ***ΚΕΝΟ10***

 **Τέλος**\_**αν**

 **Τέλος**\_**επανάληψης**

**Τέλος**\_**επανάληψης (Μονάδες 10)**

**Β2.** Να μετατρέψετε τον παρακάτω αλγόριθµο που είναι γραµµένος µε τη µέθοδο της φυσικής γλώσσας κατά βήµατα, σε ισοδύναµο που να είναι συµβατός µε τις αρχές του δοµηµένου προγραµµατισµού.

**Βήµα 1.** Θέσε κ=0

**Βήµα 2.** Θέσε λ=1

**Βήµα 3**. Αν λ<= 10 τότε πήγαινε στο Βήμα 4, αλλιώς πήγαινε στο Βήμα 11

**Βήµα 4.** Διάβασε μ

**Βήµα 5.** Αν μ>0 τότε Πήγαινε στο Βήμα 6 Αλλιώς πήγαινε στο Βήμα 8

**Βήµα 6.** Θέσε κ=κ+1

**Βήμα 7.** Πήγαινε στο Βήμα 9

**Βήµα 8.** Θέσε κ=κ-1

**Βήµα 9.** Θέσε λ=λ+1

**Βήμα10.** Πήγαινε στο Βήμα 3

**Βήμα11.** Εμφάνισε κ **(Μονάδες 10)**

 **ΘΕΜΑ Γ**

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο θα υπολογίζει προσεγγιστικά μία ρίζα της συνάρτησης f(x) = 3x2 + 2x - 3 στο διάστημα [0,1] σύμφωνα με την παρακάτω αναλυτική μέθοδο που στηρίζεται στο θεώρημα Bolzano:

**Γ1.** Θα περιλαμβάνει τμήμα δήλωσης μεταβλητών. **(Μονάδες 1)**

**Γ2.** Αρχικοποιείται το διάστημα [α, β] με τις δοθείσες αρχικές τιμές. **(Μονάδες 2)**

**Γ3.** Το πολύ 100 φορές ή μέχρι να βρεθεί η ρίζα της συνάρτησης, θα εκτελεί τα ακόλουθα: **(Μονάδες 4)**

1. Yπολογίζει το μέσο του διαστήματος **(Μονάδες 2)**
2. Αν η τιμή της συνάρτησης στο μέσο είναι μηδέν τότε βρέθηκε η ρίζα και εμφανίζεται αντίστοιχο μήνυμα. **(Μονάδες 4)**
3. Αν όχι τότε επαναλαμβάνουμε την εργασία για το διάστημα που περικλείεται από το μέσο και το ένα άκρο του διαστήματος, εκείνο το άκρο που η τιμή της συνάρτησης είναι ετερόσημη της τιμής της συνάρτησης στο μέσο.

 **(Μονάδες 5)**

**Γ4.** Στο τέλος θα εμφανίζει πόσες επαναλήψεις χρειάστηκαν για να βρεθεί τελικά η ρίζα της εξίσωσης, ή μήνυμα «Δε βρέθηκε ρίζα» σε αντίθετη περίπτωση.

 **(Μονάδες 2)**

**ΘΕΜΑ Δ**

Για τη μετατροπή ενός αριθμού από το δεκαδικό σύστημα αρίθμησης στο δυαδικό, διαιρούμε συνεχώς τον αριθμό του δεκαδικού συστήματος με το 2 (βάση του δυαδικού συστήματος) και κρατάμε το υπόλοιπο, το οποίο μπορεί να είναι 0 ή 1. Η επαναληπτική διαδικασία τερματίζεται όταν ο αριθμός γίνει ίσος με 0. Ο αντίστοιχος αριθμός στο δυαδικό σύστημα προκύπτει από τα υπόλοιπα των διαιρέσεων, σε αντίθετη σειρά από αυτή που πραγματοποιήθηκαν οι διαιρέσεις. Για παράδειγμα:

20**mod**2=0 (κρατάω το υπόλοιπο), 20**div**2=10 (div 2 για να συνεχίσω)

10**mod**2=0, 10**div**2=5

5**mod**2=1, 5**div**2=2

2**mod**2=0, 2**div**2=1

1**mod**2=1, 1**div**2=0, τέλος επαναληπτικής διαδικασίας

Ό αριθμός 20 στο δυαδικό σύστημα, χρησιμοποιώντας τα υπόλοιπα με την αντίθετη σειρά από αυτή που υπολογίστηκαν, είναι ο: 10100. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

**Δ1.** Θα περιλαμβάνει κατάλληλο τμήμα δήλωσης μεταβλητών. **(Μονάδες 1)**

**Δ2.** Θα εκτελεί την ακόλουθη επαναληπτική διαδικασία:

1. Θα διαβάζει έναν αριθμό (υποθέστε ακέραιος) από το χρήστη, με έλεγχο για θετική τιμή. Αν ο αριθμός δεν είναι θετικός, θα δίνεται ξανά.

 **(Μονάδες 1)**

1. Θα εκτελεί επαναληπτικά διαιρέσεις του αριθμού με το 2 (σύμφωνα με την περιγραφή της εκφώνησης) και θα αποθηκεύει το υπόλοιπο της διαίρεσης σε στοίβα ΑΡΙΘΜΟΣ[10], μόνο αν υπάρχει διαθέσιμος χώρος σε αυτή, διαφορετικά η επανάληψη θα τερματίζεται και θα εμφανίζεται μήνυμα «Ο διαθέσιμος χώρος δεν επαρκεί για τη μετατροπή του αριθμού».

 **(Μονάδες 6)**

1. Στην περίπτωση που τελικά έγινε η μετατροπή, θα καλεί διαδικασία ΕΜΦΑΝΙΣΗ (την οποία και θα κατασκευάσετε), η οποία θα δέχεται κατάλληλες παραμέτρους που θα επιλέξετε και θα αδειάζει τη στοίβα, εμφανίζοντας παράλληλα τα στοιχεία, που είναι ο αντίστοιχος αριθμός στο δυαδικό σύστημα. **(Μονάδες 6)**
2. Η εισαγωγή των αριθμών προς μετατροπή, θα τερματιστεί όταν δοθεί ως απάντηση το «ΟΧΙ» σε σχετική ερώτηση προς το χρήστη η οποία θα γίνεται αφότου έχουν γίνει τουλάχιστον 50 μετατροπές αριθμών.

 **(Μονάδες 3)**

**Δ3.** Μετά το τέλος της εισαγωγής αριθμών, θα εμφανίζει τους 2 μεγαλύτερους αριθμούς οι οποίοι τελικά δεν μετατράπηκαν – να υποθέσετε πως οι αριθμοί που δόθηκαν ήταν διαφορετικοί μεταξύ τους και πως υπήρχαν 2 τουλάχιστον αριθμοί που δεν μετατράπηκαν. **(Μονάδες 3)**

**ΚΑΛΗ ΕΠΙΤΥΧΙΑ !!!!**