

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΕΡΕΥΝΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ

**ΕΟΠΠΕΠ**

ΕΘΝΙΚΟΣ ΟΡΓΑΝΙΣΜΟΣ ΠΙΣΤΟΠΟΙΗΣΗΣ ΠΡΟΣΩΝΤΩΝ  
& ΕΠΑΓΓΕΛΜΑΤΙΚΟΥ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ

**ΘΕΜΑΤΑ ΕΞΕΤΑΣΕΩΝ**

**ΠΙΣΤΟΠΟΙΗΣΗΣ ΑΡΧΙΚΗΣ ΕΠΑΓΓΕΛΜΑΤΙΚΗΣ**

**ΚΑΤΑΡΤΙΣΗΣ ΕΙΔΙΚΟΤΗΤΑΣ Ι.Ε.Κ.**

**“ΤΕΧΝΙΚΟΣ ΕΦΑΡΜΟΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**(ΠΟΛΥΜΕΣΑ /**

**WEBDESIGNER - DEVELOPER /**

**VIDEOGAMES)”**

# ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ ..... 2

1. Εξετάσεις Πιστοποίησης Αρχικής Επαγγελματικής Κατάρτισης Αποφοίτων Ι.Ε.Κ. .... 3

2. Διάρκεια του Πρακτικού Μέρους των εξετάσεων ..... 3

3. Θεωρητικό Μέρος: Θέματα Εξετάσεων Πιστοποίησης Αρχικής Επαγγελματικής Κατάρτισης  
Ειδικότητας Ι.Ε.Κ. .... 4

    ΟΜΑΔΑ Α. ΓΕΝΙΚΕΣ ΕΡΩΤΗΣΕΙΣ ..... 4

    ΟΜΑΔΑ Β. ΕΙΔΙΚΕΣ ΕΡΩΤΗΣΕΙΣ .....103

4. Πρακτικό Μέρος: Κατάλογος Στοχοθεσίας Πρακτικών Ικανοτήτων και Δεξιοτήτων (Στοχοθεσία  
Εξεταστέας Ύλης Πρακτικού Μέρους) .....182

## 1. Εξετάσεις Πιστοποίησης Αρχικής Επαγγελματικής Κατάρτισης Αποφοίτων Ι.Ε.Κ.

Οι εξετάσεις Πιστοποίησης Αρχικής Επαγγελματικής Κατάρτισης αποφοίτων Ινστιτούτων Επαγγελματικής Κατάρτισης (Ι.Ε.Κ.) της ειδικότητας «**ΤΕΧΝΙΚΟΣ ΕΦΑΡΜΟΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ(ΠΟΛΥΜΕΣΑ/WEBDESIGNER-DEVELOPER/VIDEOGAMES)**» διεξάγονται σύμφωνα με τα οριζόμενα στις διατάξεις της αριθμ. **2944/2014 Κοινής Υπουργικής Απόφασης Οικονομικών και Παιδείας και Θρησκευμάτων (Φ.Ε.Κ. Β' 1098/2014)**, όπως τροποποιήθηκε και ισχύει, η οποία εκδόθηκε βάσει της διάταξης της παρ. 5, του άρθρου 25, του **Ν. 4186/2013 (Φ.Ε.Κ. Α' 193/2013)**, όπως τροποποιήθηκε με τη διάταξη της παρ. 1, του άρθρου 11, του **Ν. 4229/ 2014 (Φ.Ε.Κ. Α' 8/2014)** και ισχύει.

## 2. Διάρκεια του Πρακτικού Μέρους των εξετάσεων

Η διάρκεια εξέτασης του Πρακτικού Μέρους των εξετάσεων Πιστοποίησης Αρχικής Επαγγελματικής Κατάρτισης αποφοίτων Ινστιτούτων Επαγγελματικής Κατάρτισης (Ι.Ε.Κ.) της ειδικότητας **ΤΕΧΝΙΚΟΣ ΕΦΑΡΜΟΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΟΛΥΜΕΣΑ / WEBDESIGNER - DEVELOPER / VIDEOGAMES)**» καθορίζεται **σε τρεις (3) ώρες**.

### 3. Θεωρητικό Μέρος: Θέματα Εξετάσεων Πιστοποίησης Αρχικής Επαγγελματικής Κατάρτισης Ειδικότητας Ι.Ε.Κ.

#### ΟΜΑΔΑ Α. ΓΕΝΙΚΕΣ ΕΡΩΤΗΣΕΙΣ

1. Από ποια βασικά τμήματα αποτελείται ένας ηλεκτρονικός υπολογιστής; Να περιγράψετε συνοπτικά τις κύριες λειτουργίες του κάθε τμήματος.

Τα βασικά τμήματα από τα οποία αποτελείται ένας ηλεκτρονικός υπολογιστής είναι τα εξής:

Α) **Κεντρική μονάδα**: η οποία εξωτερικά είναι ένα μεταλλικό κουτί, περιέχει όλα τα βασικά εξαρτήματα με τα οποία δουλεύει ο υπολογιστής. Όλη η ουσιαστική εργασία του Η/Υ γίνεται μέσα σε αυτή τη μονάδα. Στην πρόσοψη της κεντρικής μονάδας το λαμπάκι λειτουργίας του σκληρού δίσκου και το βασικό διακόπτη μέσω του οποίου θέτουμε σε λειτουργία τον υπολογιστή. Στο εσωτερικό της κεντρικής μονάδας βρίσκονται: Οι συσκευές αποθήκευσης (με πιο συνηθισμένη το σκληρό δίσκο), η μητρική πλακέτα (motherboard), Το τροφοδοτικό, το οποίο διαχειρίζεται το ηλεκτρικό ρεύμα της πρίζας και εφοδιάζει με ρεύμα τον Η/Υ και οι κάρτες επέκτασης, οι οποίες προσθέτουν κάποιες νέες δυνατότητες στον υπολογιστή σας.

Β) **Οθόνη**: Η οθόνη του Η/Υ εξωτερικά παρουσιάζει αρκετά κοινά χαρακτηριστικά με μια τηλεόραση. Πράγματι, βρίσκουμε ένα διακόπτη ON-OFF, με τη βοήθεια του οποίου ανοίγουμε ή κλείνουμε την οθόνη, καθώς και διακόπτες για τη ρύθμιση της φωτεινότητας (Brightness), της αντίθεσης (Contrast) και διόρθωσης παραμορφώσεων. Οι οθόνες κατασκευάζονται σε διάφορα μεγέθη. Η μονάδα μέτρησης των διαστάσεων της οθόνης είναι η ίντσα ("). Τυπικές αναλύσεις που χρησιμοποιούνται στις οθόνες των προσωπικών υπολογιστών είναι 640X480, 800X600, 1024X768, 1280X1024 και 1600X1200.

Γ) **Περιφεριακά**: Οποιαδήποτε άλλη συσκευή συνδέεται στην Κεντρική μονάδα του υπολογιστή ονομάζεται περιφεριακή μονάδα. Οι βασικές από αυτές είναι το πληκτρολόγιο, το ποντίκι, ο εκτυπωτής κτλ.

2. Ποιες είναι οι διαθέσιμες τεχνολογίες εκτυπωτών και σε ποια χρήση απευθύνεται η κάθε μία; Να αναφέρετε τα κριτήρια αξιολόγησης ενός εκτυπωτή.

Υπάρχουν αρκετά είδη εκτυπωτών, με πιο διαδεδομένους τους ακόλουθους: α) Dot Matrix, β) Inkjet, που λειτουργούν με σύστημα ψεκασμού στο χαρτί, γ) Laser και δ) Θερμικοί εκτυπωτές.

α) Οι **Dot Matrix** παρέχουν χαμηλή ποιότητα εκτύπωσης και χρησιμοποιούνται κυρίως για εκτυπώσεις αποδείξεων, τιμολογίων κ.λπ., εκεί δηλαδή όπου η ποιότητα της εκτύπωσης δεν απασχολεί ιδιαίτερα. Η λειτουργία των εκτυπωτών ακίδων βασίζεται στην χρήση μιας μήτρας που έχει μία ή δύο στήλες με ακίδες.

β) Οι **Inkjet** παρέχουν εκτυπώσεις ικανοποιητικής ποιότητας και μπορούν να εξυπηρετήσουν τις ανάγκες ενός μικρού γραφείου και έχουν μικρό κόστος, ακόμα και οι έγχρωμοι, και είναι αρκετά αθόρυβοι. Η εκτύπωση των εκτυπωτών ψεκασμού (Inkjet) πραγματοποιείται με ψεκασμό μελάνης πάνω στο χαρτί. Η κεφαλή εκτύπωσης περιέχει το μελάνι το οποίο ψεκάζει πάνω στο χαρτί έτσι, ώστε να ζωγραφίζει την εικόνα που στέλνει ο υπολογιστής.

γ) Οι **εκτυπωτές Laser** υπερτερούν, καθώς επιτυγχάνουν τις ποιοτικότερες εκτυπώσεις αλλά το κόστος είναι μεγαλύτερο σε σχέση με τους Inkjet εκτυπωτές και ιδιαίτερα οι έγχρωμοι είναι πολύ ακριβοί. Οι εκτυπωτές Laser ονομάζονται επίσης, εκτυπωτές σελίδας, γιατί μια ακτίνα Laser ζωγραφίζει την εικόνα της σελίδας επάνω σ' ένα ηλεκτρικά φορτισμένο κύλινδρο, το τύμπανο. Το μελάνι, με μορφή σκόνης, κολλά στα φορτισμένα μέρη του τύμπανου και στη συνέχεια μεταφέρεται στο χαρτί, όπου και στερεοποιείται με τη βοήθεια θερμότητας.

δ) Οι **Θερμικοί εκτυπωτές** βρίσκουν εφαρμογή στις βιομηχανικές και παραγωγικές μονάδες, εξυπηρετώντας εκτυπωτικές ανάγκες ετικετών, σημάτων, επιγραφών κ.ά. και το κόστος είναι απαγορευτικό για αγορά για προσωπική χρήση.

### 3. Ποιες κατηγορίες αποθηκευτικών μέσων έχουμε; Σε ποια από αυτές ανήκουν οι σκληροί δίσκοι και σε ποια οι οπτικοί δίσκοι;

Οι κατηγορίες αποθηκευτικών στην επιστήμη των Η/Υ είναι οι εξής:

A. Τα μαγνητικά μέσα αποθήκευσης είναι οι **ταινίες** (magnetic tapes) και οι **δίσκοι** (disks). Η εγγραφή των ψηφιακών δεδομένων σε αυτά βασίζεται στη μόνιμη μαγνήτιση μιας μικρής περιοχής του μαγνητικού μέσου με την βοήθεια ενός ηλεκτρομαγνήτη. Η ανάγνωση γίνεται με το πέρασμα του ηλεκτρομαγνήτη πάνω από τη μαγνητισμένη περιοχή και τη δημιουργία επαγωγικού ρεύματος. Η φορά του επαγωγικού ρεύματος εξαρτάται από την πολικότητα της μαγνήτισης και επομένως καθορίζει τη τιμή της ψηφιακής πληροφορίας: 0 ή 1. Στην περίπτωση της εγγραφής έχουμε το αντίστροφο φαινόμενο. Ανάλογα με την τιμή 0 ή 1 που θέλουμε να γραφεί στο δίσκο, διοχετεύεται ρεύμα αντίστοιχης φοράς, δημιουργείται λοιπόν μαγνητικό πεδίο που μαγνητίζει αντίστοιχα την περιοχή του δίσκου.

B. Οι **οπτικοί δίσκοι** (Compact Disk) είναι τα τελευταία χρόνια το πιο διαδεδομένο μέσο αποθήκευσης μεγάλου όγκου ψηφιακών δεδομένων, ειδικά για εφαρμογές πολυμέσων. Οι συσκευές ανάγνωσης οπτικών δίσκων βασίζονται στην ανάκλαση μιας φωτεινής δέσμης laser πάνω στην επιφάνεια του δίσκου. Το υλικό κατασκευής τους ποικίλλει ανάλογα με την τεχνολογία ανάγνωσης και εγγραφής. Το κοινό χαρακτηριστικό τους είναι η μεγάλη διάρκεια ζωής τους λόγω του προστατευτικού στρώματος πλαστικού που τους περιβάλλει. Οι πιο διαδεδομένες κατηγορίες οπτικών δίσκων είναι οι εξής:

- CD-ROM: Μουσικά CD (CD-Audio) και CD δεδομένων (CD-Data)
- Εγγράψιμο CD (CD-Recordable ή WORM-Write Once Read Many)
- Επανεγγράψιμο CD (CD-RW Rewritable ή CD-Erasable)
- DVD (Digital Versatile Disks)

Γ. **USB Flash Memory**. Συνδέεται με τον υπολογιστή μέσω θύρας USB και αποθηκεύει δεδομένα σε ολοκληρωμένα κυκλώματα (τσιπ) μνήμης που βρίσκονται στο εσωτερικό του. Το είδος αυτών των τσιπ ονομάζεται "μνήμη flash" (φλας - "flash memory") και γι αυτόν το λόγο

η συσκευή ονομάζεται flash drive. Όση ώρα είναι συνδεδεμένο στον υπολογιστή μπορούμε σε αυτό να γράψουμε ή να διαβάσουμε δεδομένα με μεγάλη ταχύτητα ενώ όταν το αποσυνδέσουμε τα δεδομένα μας παραμένουν στα τσιπ μνήμης για απεριόριστο χρόνο.

Δ. Οι κάρτες μνήμης είναι πολύ μικρά αποθηκευτικά μέσα που περιέχουν ολοκληρωμένα κυκλώματα μνήμης flash (φλας). Η διαφορά τους από τα USB flash drive είναι ότι δε μπορούν να συνδεθούν απευθείας στον υπολογιστή για να διαβάσουμε και να γράψουμε δεδομένα σε αυτές, αλλά ότι χρειάζονται ειδική συσκευή ανάγνωσης καρτών μνήμης ("card reader"). Στους επιτραπέζιους υπολογιστές συνήθως απαιτείται να προμηθευτούμε μία εξωτερική συσκευή ανάγνωσης καρτών. Η τιμή μιας τέτοιας συσκευής είναι αρκετά χαμηλή.

#### **4. Τι είναι οι σαρωτές (scanners) και τι τα προγράμματα OCR;**

Ο σαρωτής (scanner) είναι μια συσκευή που μεταφράζει μια τυπωμένη εικόνα σε ψηφιακά δεδομένα, δηλαδή ψηφιοποιεί τα εικόνα και τη μεταφέρει στον υπολογιστή, είτε αυτή είναι ασπρόμαυρη είτε έγχρωμη. Τα βασικά χαρακτηριστικά που καθορίζουν την ποιότητα και την τιμή ενός σαρωτή είναι η ταχύτητα σάρωσης, η ανάλυση και τα υποστηριζόμενα μεγέθη (A3, A4, κ.λπ.). Το κόστος ενός σαρωτή είναι αρκετά προσιτό.

Η οπτική αναγνώριση χαρακτήρων, (Optical Character Recognition), ή αλλιώς OCR, είναι μια διαδικασία κατά την οποία ένα ειδικό πρόγραμμα διαβάζει μια εικόνα και αν η εικόνα έχει γράμματα εν σειρά, υπό μορφή κειμένου, μπορεί να μετατρέψει την εικόνα σε κείμενο αναγνωρίζοντας τους χαρακτήρες οπότε το αποτέλεσμα να μπορεί να είναι επεξεργάσιμο σε κάποιον επεξεργαστή κειμένου, όπως πχ το word του Office.

#### **5. Τι είναι μια Άδεια Χρήσης λογισμικού και ποιο δικαίωμα παραχωρεί;**

Άδεια λογισμικού είναι ο όρος ο οποίος περιγράφει τα νομικά δικαιώματα τα οποία έχουν οι χρήστες ως προς την χρήση, επεξεργασία και διανομή του λογισμικού, και τα οποία καθορίζονται από τον αρχικό δημιουργό του λογισμικού. Μια τυπική άδεια λογισμικού παρέχει σε έναν τελικό χρήστη άδεια να χρησιμοποιήσει ένα ή περισσότερα αντίγραφα του λογισμικού, με τρόπους που αν δεν υπήρχε η είχε δοθεί η άδεια θα αποτελούσαν καταπάτηση των δικαιωμάτων του εκδότη ο οποίος είναι και ως ο ιδιοκτήτης των δικαιωμάτων χρήσης/copyright είναι αυτός που τα καθορίζει. Πρακτικά, η άδεια λογισμικού καθορίζει τις ελευθερίες του χρήστη ως προς το τι μπορεί να κάνει νόμιμα, με την αρχική δημιουργία κάποιου άλλου.

#### **6. Τί είναι η άρνηση υπηρεσίας DOS (Denial of Service) σαν μορφή επίθεσης σε ένα δίκτυο και με ποιους μηχανισμούς μπορεί ένα δίκτυο να προστατευτεί από τέτοιου είδους επιθέσεις;**

Επιθέσεις άρνησης εξυπηρέτησης (DoS attack) ονομάζονται γενικά οι επιθέσεις εναντίον ενός υπολογιστή, ή μιας υπηρεσίας που παρέχεται, οι οποίες έχουν ως σκοπό να καταστήσουν τον υπολογιστή ή την υπηρεσία ανίκανη να δεχτεί άλλες συνδέσεις και έτσι να μην μπορεί να

εξυπηρετήσει άλλους πιθανούς πελάτες. Οι αμυντικές απαντήσεις σε επιθέσεις άρνησης εξυπηρέτησης περιλαμβάνουν συνήθως τη χρήση συνδυασμού εντοπισμού επίθεσης, ταξινόμησης επισκεψιμότητας και εργαλείων απόκρισης, με στόχο την παρεμπόδιση της επισκεψιμότητας που αναγνωρίζουν ως παράνομη και επιτρέπουν την αναγνώριση της κυκλοφορίας ως νόμιμης. Παρατίθεται ένας κατάλογος εργαλείων πρόληψης και αντίδρασης: Εφαρμογή εξοπλισμού front end, πρόληψη βάσει IPS, άμυνα βασισμένη στο DDS, παραμετροποίηση δρομολογητών και switches, Upstream φιλτράρισμα.

**7. Τι είναι οι Ιοί των υπολογιστών (virus) και σε ποιες κυρίες κατηγορίες μπορούν χωριστούν;**

Ένας ιός υπολογιστών είναι ένα κακόβουλο πρόγραμμα υπολογιστή, το οποίο μπορεί να αντιγραφεί χωρίς παρέμβαση του χρήστη και να "μολύνει" τον υπολογιστή χωρίς τη γνώση ή την άδεια του χρήστη του. Ο αρχικός ιός μπορεί να τροποποιήσει τα αντίγρατά του ή τα ίδια τα αντίγραφα μπορούν να υποστούν από μόνα τους τροποποίηση, όπως συμβαίνει σε έναν μεταμορφικό ιό. Ένας ιός μπορεί να διαδοθεί από έναν υπολογιστή σε άλλους, παραδείγματος χάριν από ένα χρήστη που στέλνει τον ιό μέσω δικτύου ή του Διαδικτύου, ή με τη μεταφορά του σε ένα φορητό μέσο αποθήκευσης, όπως δισκέτα, οπτικό δίσκο ή μνήμη flash USB. Μερικοί ιοί δημιουργούνται για να προξενήσουν ζημιά στον υπολογιστή, στον οποίο εγκαθίστανται, είτε με την καταστροφή των προγραμμάτων του είτε με τη διαγραφή αρχείων ή με τη μορφοποίηση (format) του σκληρού δίσκου. Άλλοι δεν έχουν ως σκοπό να προκαλέσουν οποιαδήποτε ζημιά, αλλά απλά γνωστοποιούν την παρουσία τους με την εμφάνιση στην οθόνη κειμένου, βίντεο, ή ηχητικών μηνυμάτων, μερικές φορές αρκετά χιουμοριστικών.

Οι ιοί μπορούν να ταξινομηθούν σε δύο μεγάλες κατηγορίες:

- Ανάλογα με το σημείο του υλικού ή του λογισμικού που μολύνουν:
  - Τομείς σκληρού δίσκου συστήματος (system sectors)
  - Αρχεία
  - Ιοί μακροεντολών (Macros)
  - Ιοί πηγαίου κώδικα (Source Code Viruses)
  - Ιοί συμπλεγμάτων (σκληρού) δίσκου ((Hard) Disk Clusters)
- Ανάλογα με τον τρόπο με τον οποίο πραγματοποιούν τη μόλυνση:
  - Πολυμορφικοί ιοί
  - Αόρατοι ιοί (Stealth Viruses)
  - Θωρακισμένοι ιοί (Armored Viruses)
  - Πολυτμηματικοί ιοί (Multipartite Viruses)
  - Ιοί πλήρωσης κενών (Spacefiller Viruses)
  - Ιοί παραλλαγής (Camouflage Viruses)

**8. Ποιός ο ρόλος των αρχών πιστοποίησης ψηφιακών πιστοποιητικών και ποιά η διαδικασία έκδοσης ενός πιστοποιητικού για ένα ψηφιακό κατάστημα;**

Με την υπηρεσία Ψηφιακών Πιστοποιητικών οι χρήστες μπορούν να εκδώσουν πιστοποιητικά τόσο για προσωπική χρήση όσο και για διαδικτυακές υπηρεσίες.

Τα ψηφιακά πιστοποιητικά βελτιώνουν το επίπεδο ασφάλειας στις διαδικτυακές επικοινωνίες, εξασφαλίζοντας:

- την ασφαλή πρόσβαση σε ιστοσελίδες δικτυακών τόπων που πιστοποιούνται για την ακρίβεια των στοιχείων τους και συνεπώς για την αληθινή ταυτότητα του υπευθύνου φορέα.
- τη δυνατότητα εξακρίβωσης των στοιχείων του αποστολέα ηλεκτρονικών μηνυμάτων ή άλλων εγγράφων, και τον έλεγχο της ορθότητας του περιεχομένου τους, εφόσον ο αποστολέας επιλέγει να τα υπογράψει με προσωπικό ψηφιακό πιστοποιητικό που έχει εκδοθεί για λογαριασμό του.

Η έκδοση των πιστοποιητικών βασίζεται στις αρχές της επιστήμης της Κρυπτογραφίας. Η Υποδομή Δημόσιου Κλειδιού (Public Key Infrastructure, PKI) αποτελεί ένα συνδυασμό λογισμικού, τεχνολογιών κρυπτογραφίας και διαδικασιών, ο οποίος κατά βάση πιστοποιεί την εγκυρότητα κάθε εμπλεκόμενου σε μια ψηφιακή συναλλαγή.

Η υπηρεσία συμβάλλει στη μείωση της γραφειοκρατίας αλλά και στην προώθηση των ασφαλών διαδικτυακών συναλλαγών και της ηλεκτρονικής επικοινωνίας.

**9. Ποια είδη διαμόρφωσης ενός σκληρού δίσκου γνωρίζετε, ποιο δίνει μεγαλύτερη ασφάλεια και γιατί χρειάζεται;**

Δύο είδη διαμόρφωσης σκληρού δίσκου υπάρχουν: η γρήγορη και η κανονική.

Η διαφορά της γρήγορης με την κανονική διαμόρφωση δίσκου είναι πως κατά την κανονική, μαζί με την διαγραφή των αρχείων, γίνεται ταυτόχρονα και έλεγχος για κατεστραμμένους τομείς στο δίσκο. Στη γρήγορη διαμόρφωση, αφαιρούνται μόνο τα αρχεία, χωρίς να γίνεται έλεγχος. Αυτό έχει σαν αποτέλεσμα η κανονική διαμόρφωση δίσκου σε ένα δίσκο μεγάλης χωρητικότητας να παίρνει ώρες. Αν ο δίσκος είναι σχετικά καινούριος και γενικά δεν έχετε προβλήματα μαζί του, η γρήγορη διαμόρφωση δίσκου είναι η καλύτερη επιλογή.

**10. Τι χρειάζεται προκειμένου να κατασκευαστεί ένα στοιχειώδες σύστημα Η/Υ και με ποιο τρόπο επιτυγχάνεται η διασύνδεση μεταξύ των μονάδων του;**

Για κατασκευαστεί ένας στοιχειώδης ηλεκτρονικός υπολογιστής είναι απαραίτητη αρχικά η ύπαρξη μιας μητρικής πλακέτας και της Κεντρικής Μονάδας Επεξεργασίας που βρίσκεται σε αυτήν. Εξίσου απαραίτητη είναι και η ύπαρξη μνήμης RAM και ROM, οι οποίες επίσης βρίσκονται πάνω στη μητρική πλακέτα. Η διασύνδεση αυτών των στοιχείων γίνεται με ειδικούς διαύλους ηλεκτρικής τάσης.

Επίσης απαιτούνται ορισμένες περιφερειακές μονάδες όπως το τροφοδοτικό, ο σκληρός δίσκος, το πληκτρολόγιο, το ποντίκι και η οθόνη. Η διασύνδεση με αυτές τις μονάδες γίνεται με συγκεκριμένους τύπους καλωδίων.



**11. Ποια είναι τα αρχεία που λαμβάνουν μέρος στη διαδικασία συγχώνευσης αλληλογραφίας, σε έναν επεξεργαστή κειμένου τύπου Word και ποιος ο ρόλος τους;**

Τρία είναι τα βασικά έγγραφα που χρησιμοποιούνται στη διαδικασία συγχώνευσης αλληλογραφίας.

- Το Κύριο Έγγραφο ή Έγγραφο εκκίνησης, το οποίο χρησιμοποιείτε ως το βασικό έγγραφο για το είδος αλληλογραφίας που έχουμε επιλέξει (επιστολές, φάκελοι κτλ).
- Το αρχείο προέλευσης δεδομένων, το οποίο περιέχει όλα τα απαραίτητα στοιχεία για τη δημιουργία των επιστολών ή των φακέλων κτλ. Συνήθως χρησιμοποιείται το πρόγραμμα Microsoft Excel του Office ή αναπαράγεται νέο αρχείο (λίστα) μέσα από το πρόγραμμα Microsoft Word.
- Το συγχωνευμένο έγγραφό, το οποίο αποτελεί το αποτέλεσμα της όλης διαδικασίας και διατίθεται για επεξεργασία, εκτύπωση αλλά και αποθήκευση.

**12. Αναφέρετε τις κατηγορίες των συναρτήσεων σε ένα λογιστικό φύλλο τύπου Excel και περιγράψτε τη χρήση της συνάρτησης If.**

Οι συναρτήσεις φύλλου εργασίας του Excel είναι οργανωμένες στις εξής κατηγορίες

- Financial (Οικονομικές)
- Logical (Λογικές)
- Text (Κειμένου)
- Date & Time (Ημερομηνίας και ώρας)
- Lookup & Reference (Αναζήτησης και αναφοράς)
- Math & Trig (Μαθηματικών και τριγωνομετρίας)
- Statistical (Στατιστικές)
- Engineering (Μηχανολογικές)
- Database (Βάση δεδομένων)
- Cube (Κύβου)
- Information (Πληροφοριών)
- User defined (Συναρτήσεις χρήστη)

Η συνάρτηση IF είναι μία από τις πιο δημοφιλείς συναρτήσεις στο Excel και σας επιτρέπει να κάνετε λογικές συγκρίσεις μεταξύ μιας τιμής και της τιμής που αναμένετε. Στην απλούστερη μορφή της, η συνάρτηση IF αναφέρει ότι:

IF(εάν κάτι είναι αληθές, τότε κάνε κάτι, διαφορετικά κάνε κάτι άλλο)

Επομένως, μια πρόταση IF μπορεί να έχει δύο αποτελέσματα. Το πρώτο αποτέλεσμα προκύπτει εάν η σύγκριση είναι αληθής, το δεύτερο εάν η σύγκριση είναι ψευδής.

**13. Ποιες επιλογές προβολής εγγράφων περιλαμβάνει ένας επεξεργαστής κειμένου, τύπου Word; Περιγράψτε τις.**

Οι δυνατότητες προβολής εγγράφου στο Microsoft Word του Office είναι οι εξής:

- Η Διάταξη Εκτύπωσης στην οποία το έγγραφο εμφανίζεται ως σελίδα όπως περίπου θα τυπωθεί
- Η προβολή Ανάγνωσης πλήρους οθόνης η οποία διευκολύνει την ανάγνωση ενός εγγράφου και χρησιμοποιείται κυρίως όταν ο χρήστης δεν επιθυμεί την οποιαδήποτε παραμετροποίηση του.
- Η Διάταξη web εμφανίζει το έγγραφο ως μέρος μιας ιστοσελίδας και κατά συνέπεια καταργεί ορισμένους περιορισμούς στην εμφάνιση του εγγράφου όπως για παράδειγμα τα περιθώρια εκτύπωσης.
- Η προβολή Διάρθρωσης χρησιμοποιείτε για τη δημιουργία ή επεξεργασία επικεφαλίδων και οργάνωση ενός πολύπλοκου εγγράφου. Παρέχει τη δυνατότητα στο χρήστη να μπορεί εύκολα να προσαρμόσει κατάλληλα τα επίπεδα επικεφαλίδας, και να αναδιατάξει το περιεχόμενο μέχρι όλα τα στοιχεία να βρίσκονται εκεί που επιθυμεί.
- Η προβολή Πρόχειρη μοιάζει με πάρα πολύ με την προβολή εκτύπωσης αλλά διαθέτει λιγότερα στοιχεία γραφικού περιβάλλοντος κατά την εμφάνισή της και χρησιμοποιείται για συγγραφή πρόχειρου κειμένου δίχως ιδιαίτερη επεξεργασία.

**14. Ποιος ο σκοπός χρησιμοποίησης αναφορών στους τύπους ενός λογιστικού φύλλου, τύπου Excel; Πόσα και ποια είδη αναφορών τύπων υπάρχουν;**

Οι τύποι μπορεί μερικές φορές να έχουν ως αποτέλεσμα τιμές σφάλματος, καθώς και να επιστρέφουν μη αναμενόμενα αποτελέσματα. Είναι απαραίτητο να χρησιμοποιούμε αυτές τις αναφορές για να βρίσκουμε και να διερευνούμε τις αιτίες για αυτά τα σφάλματα και να είμαστε σε θέση να προσδιορίσουμε λύσεις.

#####	Η στήλη δεν έχει αρκετό πλάτος για να εμφανίσει την τιμή.
#Τιμή! (#Value!)	Χρησιμοποιείται λάθος τύπος ορίσματος ή τελεστή.
#ΔΙΑΙΡ/0! (#DIV/0!)	Αριθμός διαιρείται με το μηδέν.
#ΟΝΟΜΑ? (#NAME?)	Η εφαρμογή δεν αναγνωρίζει κείμενο σε έναν τύπο.
#Δ/Y (#N/A)	Η τιμή δεν είναι διαθέσιμη σε μια συνάρτηση ή έναν τύπο.
#ΑΝΑΦ! (#REF!)	Η αναφορά κελιού δεν είναι έγκυρη.
#ΑΡΙΘ! (#NUM!)	Άκυρες αριθμητικές τιμές σε έναν τύπο ή μια συνάρτηση.
#ΚΕΝΟ! (#NULL!)	Έχει καθοριστεί μια τομή δύο περιοχών που δεν τέμνονται.

**15. Ποιες είναι οι κατηγορίες και τα βασικά χαρακτηριστικά των εκτυπωτών;**

Οι εκτυπωτές διαχωρίζονται σε τρεις κατηγορίες ανάλογα με την τεχνολογία εκτύπωσης που χρησιμοποιούν:

- Κρουστικοί εκτυπωτές (Impact printers)
- Εκτυπωτές ψεκασμού (Inkjet printers)
- Εκτυπωτές laser (Laser printers)

Ένας άλλος διαχωρισμός των εκτυπωτών γίνεται ως προς τον τρόπο εκτύπωσης, σε εκτυπωτές γραμμής και εκτυπωτές σελίδας.

- Οι εκτυπωτές σελίδας (page printers) ολοκληρώνουν το σχηματισμό της σελίδας στην μνήμη τους πριν προχωρήσουν στην εκτύπωση και κατόπιν την εκτυπώνουν.
- Οι εκτυπωτές γραμμής (line printers) εκτυπώνουν τη σελίδα γραμμή-γραμμή.

Τα βασικότερα χαρακτηριστικά των εκτυπωτών είναι τα εξής:

- Ταχύτητα εκτύπωσης: Ανάλογα με την τεχνολογία εκτύπωσης έχουμε και διαφορετικές μονάδες μέτρησης. Στους εκτυπωτές γραμμής, η ταχύτητα εκφράζεται σε χαρακτήρες ανά δευτερόλεπτο (characters per second - cps) ενώ στους εκτυπωτές σελίδας, σε σελίδες ανά λεπτό (pages per minute - ppm).
- Ανάλυση εκτύπωσης: Η ελάχιστη μονάδα εκτύπωσης (με άλλα λόγια, το πιο μικρό ίχνος) που μπορεί να τυπωθεί στο χαρτί και ονομάζεται κουκίδα (dot). Η ανάλυση εκτύπωσης ορίζεται ως το πλήθος των διαφορετικών κουκίδων που μπορούν να εκτυπωθούν ανά ίντσα1 (dots per inch - dpi).

Η ποιότητα της τελικής εκτύπωσης ενός εκτυπωτή εξαρτάται σε μεγάλο βαθμό από την ανάλυση εκτύπωσης που αναφέρει ο κατασκευαστής του, όχι όμως αποκλειστικά από αυτήν. Στην πράξη η τεχνολογία που χρησιμοποιείται για την εκτύπωση καθορίζει και την ποιότητα, όπως θα δούμε στη συνέχεια. Το πλήθος των κουκίδων ανά ίντσα είναι σημαντικό, όμως παίζουν ρόλο και άλλα χαρακτηριστικά, όπως το μέγεθος και το σχήμα της κουκίδας, η ακρίβεια στοίχισης των κουκίδων που ανήκουν στην ίδια ευθεία, το χαρτί εκτύπωσης και άλλα. Έτσι η επιλογή ενός ποιοτικού εκτυπωτή, εξαρτάται από πολλούς άλλους παράγοντες εκτός της τιμής της ανάλυσης εκτύπωσης.

**16. Τι ονομάζεται κώδικας ή σύνολο χαρακτήρων; Τι γνωρίζετε για τους κώδικες χαρακτήρων ASCII και Unicode; Από πόσα bit αποτελείται ο καθένας;**

Μια κωδικοποίηση χαρακτήρων αποτελείται από έναν κώδικα που συσχετίζει ένα σύνολο χαρακτήρων όπως πχ οι χαρακτήρες που χρησιμοποιούμε σε ένα αλφάβητο με ένα διαφορετικό σύνολο πχ αριθμών , ή ηλεκτρικών σημάτων, προκειμένου να διευκολυνθεί η αποθήκευση ,διαχείριση κειμένου σε υπολογιστικά συστήματα καθώς και η μεταφορά κειμένου μέσω τηλεπικοινωνιακών δικτύων.

Ο κώδικας ASCII (American Standard Code for Information Interchange, Αμερικανικός Πρότυπος Κώδικας για Ανταλλαγή Πληροφοριών) είναι ένα κωδικοποιημένο σύνολο χαρακτήρων του λατινικού αλφάβητου όπως αυτό χρησιμοποιείται σήμερα στην Αγγλική γλώσσα και σε άλλες δυτικοευρωπαϊκές γλώσσες. Χρησιμοποιείται για αναπαράσταση κειμένου στους υπολογιστές, σε συσκευές τηλεπικοινωνίας, καθώς και σε άλλες συσκευές που δουλεύουν με κείμενο. Οι περισσότερες σύγχρονες κωδικοποιήσεις χαρακτήρων βασίζονται στον ASCII, αν και υποστηρίζουν πολύ περισσότερους χαρακτήρες.

Το Unicode έχει πολύ ευρύτερο μητρώο χαρακτήρων, και οι διάφορες μορφές κωδικοποίησής του άρχισαν να εκτοπίζουν το ASCII ταχύτατα σε πολλά περιβάλλοντα. Ενώ ο ASCII περιορίζεται σε 128 χαρακτήρες, το Unicode υποστηρίζει πολύ περισσότερους διαχωρίζοντας τις έννοιες της μοναδικής ταυτοποίησης (χρησιμοποιώντας φυσικούς αριθμούς που αποκαλούνται κωδικά σημεία, code points) και της κωδικοποίησης (σε δυαδικές μορφές 8, 16 ή 32 bit, που ονομάζονται UTF-8, UTF-16 και UTF-32).

Για να είναι δυνατή η προς τα πίσω συμβατότητα, οι 128 χαρακτήρες του ASCII έχουν τοποθετηθεί στα ίδια κωδικά σημεία της κωδικοποίησης Unicode. Συνεπώς ο ASCII μπορεί να θεωρηθεί διάταξη 7-bit για ένα πολύ μικρό υποσύνολο της Unicode και αντίστοιχα οι μορφές κωδικοποίησης UTF- είναι δυαδικά συμβατές με τον ASCII για κωδικά σημεία κάτω από το 128, που σημαίνει ότι όλος ο ASCII είναι έγκυρος εντός του UTF-8. Οι άλλες μορφές κωδικοποίησης αναπαριστούν τον ASCII με τον τρόπο που αναπαριστούν τους πρώτους 128 χαρακτήρες του Unicode, αλλά χρησιμοποιούν 16 ή 32 bit ανά χαρακτήρα και έτσι απαιτούν μετατροπή για να είναι συμβατές.

**17. Ποιες βασικές λειτουργίες μπορούμε να εκτελέσουμε σ' ένα αρχείο;**

Άνοιγμα, (Εκτύπωση), Αποκοπή, Αντιγραφή, Επικόλληση, Διαγραφή, Μετονομασία, Δημιουργία Συντόμευσης, Συμπίεση.

**18. Τι είναι τα δεδομένα και ποια η έννοια της πληροφορίας; Ποια είναι η διαφορά μεταξύ πληροφορίας και δεδομένων;**

Ειδικότερα στην πληροφορική συναντούμε τα δεδομένα στον πληθυντικό αριθμό, σπανιότερα στον ενικό (δεδομένο). Γενικότερα, δεδομένο ονομάζεται ένα γνωστό ή αποδεκτό στοιχείο το οποίο χρησιμοποιείται ως βάση ή προϋπόθεση στην επίλυση προβλημάτων. Τα δεδομένα μπορεί να είναι σημεία πληροφοριών επί επιστημονικών παρατηρήσεων ή συμπεριφοράς και να περιλαμβάνουν λέξεις - έννοιες, αριθμούς, σύμβολα, διαγράμματα, σχέδια, φωτογραφίες, μαγνητοταινίες κλπ που περιγράφουν ή αντιπροσωπεύουν ποσότητες, έννοιες, ιδέες, αντικείμενα, γεγονότα, καταστάσεις και λειτουργίες. Ενδεχομένως κάποιοι από τους τύπους δεδομένων που παρατίθενται εμπεριέχουν ήδη εμφανείς πληροφορίες, όχι όμως την πληροφορία στο επίπεδο που συνθέτει μια αξιολόγησή τους.

Γενικά, μπορούμε να ονομάσουμε Δεδομένα (Data) τα στοιχεία που χρησιμοποιούμε για επεξεργασία. Τα αποτελέσματα που παίρνουμε από την επεξεργασία των δεδομένων και μας μεταδίδουν κάποια επιπρόσθετη γνώση, τα χαρακτηρίζουμε ως Πληροφορία (Information). Η

επεξεργασία των δεδομένων έχει ποικίλες μορφές, χωρίς να είναι πάντοτε απαραίτητη η εφαρμογή αριθμητικών πράξεων.

Αν και αρχικά τα δεδομένα δε φαίνεται να έχουν τόση σημασία για μας, αποτελούν πολύτιμα στοιχεία, για να πάρουμε χρήσιμες πληροφορίες. Η πληροφορία, σε αντίθεση με τα δεδομένα, είναι κάτι που έχει σημασία για μας, κάτι που θέλουμε να μάθουμε. Οι πληροφορίες προκύπτουν από τα επιλεγμένα δεδομένα μετά από κατάλληλη επεξεργασία. Με άλλα λόγια, η πληροφορία ή οι πληροφορίες που παίρνουμε εξαρτώνται και από το είδος της επεξεργασίας που εφαρμόζουμε στα δεδομένα τα οποία διαθέτουμε. Αν, για παράδειγμα, τα δεδομένα είναι μουσικές νότες, η διαφορετική παράθεσή τους, δηλαδή η διαφορετική επεξεργασία τους, μας δίνει ως αποτέλεσμα μια διαφορετική σύνθεση, δηλαδή ένα διαφορετικό αποτέλεσμα.

**19. Τι είναι η μνήμη RAM και ROM; Να αναφέρετε τις διαφορές τους.**

Η μνήμη ROM είναι μόνιμη μνήμη, δηλαδή κρατάει το περιεχόμενό της ακόμα και όταν ο υπολογιστής είναι κλειστός. Στη ROM είναι αποθηκευμένο το πρόγραμμα εκκίνησης του υπολογιστή που φορτώνει τα Windows. Από τη μνήμη ROM, όμως, μπορούμε μόνο να διαβάζουμε δεδομένα και δεν μπορούμε να αποθηκεύσουμε καμία πληροφορία σε αυτήν.

Η μνήμη RAM, γνωστή ως Μνήμη Τυχαίας Προσπέλασης (Random Access Memory) αποτελεί την κύρια μνήμη του υπολογιστή. Χρησιμοποιείται για την αποθήκευση δεδομένων κατά την εκτέλεση ενός προγράμματος στον υπολογιστή. Το μεγαλύτερο μειονέκτημα της μνήμης RAM είναι ότι χάνει τα δεδομένα της μόλις κλείσετε τον υπολογιστή σας και για αυτό δεν μπορεί να χρησιμεύσει για μακροχρόνια αποθήκευση.

**20. Τι είναι η λανθάνουσα - κρυφή μνήμη cache; Πως λειτουργεί;**

Η λανθάνουσα ή κρυφή μνήμη (cache memory) L1, L2 και L3, είναι μνήμη που βρίσκεται εσωτερικά στο chip του επεξεργαστή. Πρόκειται για ταχύτερη μνήμη προσωρινής αποθήκευσης δεδομένων, στην οποία αποθηκεύονται πρόσφατα χρησιμοποιημένα δεδομένα ή δεδομένα που χρησιμοποιούνται συχνότερα από τον επεξεργαστή. Ο επεξεργαστής όταν χρειάζεται κάποιο δεδομένο, ελέγχει πρώτα τη μνήμη Cache και στην περίπτωση που δε το εντοπίσει εκεί το αναζητά στην κύρια μνήμη (RAM). Η αύξηση της μνήμης Cache ενός επεξεργαστή αυξάνει και την συνολική απόδοσή του.

**21. Να μετατρέψετε τον δεκαδικό αριθμό 120 σε δυαδικό και οκταδικό αριθμό.**

Για να μετατρέψουμε έναν δεκαδικό αριθμό σε δυαδικό κάνουμε διαδοχικές διαιρέσεις στο πηλίκο του δεκαδικού αριθμού με διαιρέτη το αριθμό 2. Τα υπόλοιπα από την εκάστοτε διαίρεση σε αντίστροφη σειρά αποτελούν το ζητούμενο δυαδικό αριθμό. Συνεπώς ο δεκαδικός αριθμός 120 θα γίνει (αριστερά εμφανίζεται το πηλίκο των διαιρέσεων ενώ δεξιά το υπόλοιπο):

120 |  
60 | 0  
30 | 0  
15 | 0

7 | 1

3 | 1

1 | 1

0 | 1

Άρα  $120_{<10>} = 1111000_{<2>}$ 

**22. Στον υπολογιστή τι εργασία εκτελεί ο επεξεργαστής; Αναφέρετε μερικά είδη επεξεργασιών.**

Ο Επεξεργαστής (processor) ή αλλιώς Κεντρική Μονάδα Επεξεργασία – ΚΜΕ (Central Processing Unit – CPU) είναι το βασικότερο τμήμα ενός υπολογιστή και θεωρείται ο εγκέφαλος ενός υπολογιστή. Ο επεξεργαστής βρίσκεται τοποθετημένος πάνω στη μητρική πλακέτα (motherboard). Εκτελεί ένα μεγάλο μέρος από τη δουλειά του υπολογιστή και δεδομένα φτάνουν και φεύγουν από τον επεξεργαστή συνεχώς. Δέχεται εντολές επεξεργασίας δεδομένων, που πρέπει να εκτελέσει. Συνεπώς, μπορούμε να πούμε ότι η δουλειά του είναι να κάνει πράξεις και να μεταφέρει δεδομένα. Τα δεδομένα που δέχεται μπορούν να χωριστούν σε δύο κατηγορίες:

- Εντολές (instructions) για το πως θα διαχειριστεί τα υπόλοιπα δεδομένα.
- Δεδομένα, τα οποία πρέπει να διαχειριστεί σύμφωνα με τις εντολές.

Αυτό που αποκαλούμε εντολές, είναι κώδικας προγράμματος. Περιλαμβάνει τα μηνύματα (εντολές) που συνεχώς στέλνουμε στον υπολογιστή μέσω του ποντικιού ή του πληκτρολογίου ή κάποιων άλλων μονάδων εισόδου. Μηνύματα για να εκτυπώσει, να αποθηκεύσει, να ανοίξει ένα αρχείο, κλπ.

**23. Ποιες οι διαφορές μεταξύ ενός interpreter και ενός compiler;**

Ειδικά προγράμματα που μεταφράζουν ένα πρόγραμμα από γλώσσα προγραμματισμού σε γλώσσα μηχανής. Διακρίνονται σε δύο κατηγορίες

- a. compiler – μεταγλωττιστής γλώσσας υψηλού επιπέδου
- b. interpreter – διερμηνέας δηλ. μεταγλωττιστής με ταυτόχρονη εκτέλεση του προγράμματος.

**A) Compiler :**

- Διαβάζει το πηγαίο πρόγραμμα και κάνει γραμματικό και συντακτικό έλεγχο εμφανίζοντας τα λάθη ( diagnostics)
- Η διαδικασία επαναλαμβάνεται μέχρι να μηδενιστούν τα λάθη
- Παράγει τον αντικείμενο κώδικα
- Κάνει κατάληψη του απαραίτητου χώρου στην κύρια μνήμη για να μπορεί να εκτελεστεί το πρόγραμμα
- Φυλάσσει το αντικείμενο πρόγραμμα στο δίσκο



## B) Interpreter :

- Ελέγχει συντακτικά και γραμματικά το πηγαίο πρόγραμμα εξετάζοντας εντολή προς εντολή το πρόγραμμα. Αν δεν υπάρχουν λάθη στην εντολή , παράγει την αντίστοιχη εντολή σε γλώσσα μηχανής, την εκτελεί και ελέγχει-μεταφράζει την επόμενη εντολή.
- Η χρήση interpreter είναι γνωστή από τη γλώσσα Basic, η οποία οφείλει τη δημοτικότητα της στο συγκεκριμένο μεταγλωττιστικό πρόγραμμα.

## Διαφορές interpreter σε σχέση με τον compiler

- Δεν ελέγχει τη σύνταξη ολόκληρου του πηγαίου προγράμματος πριν από την εκτέλεση
- Επαναλαμβάνει την μετάφραση-έλεγχο κάθε φορά που εκτελείται το πρόγραμμα
- Ο έλεγχος και η διόρθωση λαθών γίνονται ευκολότερα γιατί πληροφορούμαστε αμέσως κάθε λάθος, το διορθώνουμε και συνεχίζεται η εκτέλεση
- Η εκτέλεση ενός προγράμματος με Interpreter είναι πολύ πιο αργή σε σύγκριση με τον Compiler.

**24. Ποιες είναι οι διαφορές ανάμεσα στην επαναληπτική εντολή "όσο...επανάλαβε" (while ... do) και την επαναληπτική εντολή "αρχή\_επανάληψης...μέχρις\_ότου" (repeat.until);**

Η διαφορά, έγκειται στο ότι η επαναληπτική εντολή while ... do (ελέγχει πρώτα την συνθήκη και έπειτα εκτελεί τις εντολές εντός του βρόχου), ενώ η επαναληπτική εντολή repeat...until ελέγχει την ορθότητα της συνθήκης μετά το πέρας μιας τουλάχιστον επανάληψης. Ακόμη μια διαφορά είναι το γεγονός ότι, για την while ... do επαναληπτική εντολή απαιτείται να ισχύει η συνθήκη προκειμένου να εκτελεστούν οι εντολές του βρόχου ενώ για τη επαναληπτική εντολή repeat...until ισχύει ότι επαναλαμβάνεται μέχρι να γίνει ψευδής η συνθήκη.

**25. Ποιους τρόπους γνωρίζετε για την αναπαράσταση ενός αλγορίθμου; Δώστε μία σύντομη περιγραφή για τον καθένα.**

- 1) *Ελεύθερο κείμενο*: Αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρουσίασης αλγορίθμου. Μπορεί εύκολα να οδηγήσει σε μη εκτελέσιμη παρουσίαση παραβιάζοντας την "αποτελεσματικότητα".
- 2) *Φυσική γλώσσα κατά βήματα*: Στην περίπτωση αυτή χρειάζεται προσοχή, γιατί μπορεί να παραβιασθεί ο "καθορισμός".
- 3) *Διάγραμμα ροής*: Αποτελείται από ένα σύνολο γεωμετρικών σχημάτων, όπου το καθένα δηλώνει μία συγκεκριμένη ενέργεια ή λειτουργία. Τα γεωμετρικά σχήματα ενώνονται μεταξύ τους με βέλη, που δηλώνουν τη σειρά εκτέλεσης των ενεργειών αυτών.

**26. Ποιες είναι οι διαφορές ανάμεσα στα συντακτικά και τα λογικά λάθη; Δώστε από ένα παράδειγμα στη γλώσσα προγραμματισμού C.**

Τα λογικά λάθη που είναι τα πλέον σοβαρά και δύσκολα στη διόρθωση τους, οφείλονται σε σφάλματα κατά την υλοποίηση του αλγορίθμου, ενώ τα συντακτικά οφείλονται σε αναγραμματισμούς ονομάτων εντολών, παράληψη δήλωσης δεδομένων και πρέπει πάντα να

διορθωθούν, ώστε να παραχθεί το τελικό εκτελέσιμο πρόγραμμα. Τα λογικά λάθη εμφανίζονται μόνο στην εκτέλεση, ενώ τα συντακτικά λάθη στο στάδιο της μεταγλώττισης

#### Παράδειγμα Λογικού Λάθους

```
int main() {  
    int matrix [10];  
    for(int i=0;i<15;i++){  
        matrix[i]=i;  
    }  
}
```

Στη περίπτωση αυτή, λόγω του ότι πρόκειται να προσπελαστούν θέσεις του πίνακα οι οποίες υπερβαίνουν το μέγεθος του, κατά την εκτέλεση θα προκύψει λογικό σφάλμα.

#### Παράδειγμα Συντακτικού Λάθους

```
int main() {  
    int x=20;  
    int y=5;  
  
    while (x<y) {  
        printf ("%d", z);  
        x=x-1;  
    }  
}
```

Στη περίπτωση αυτή, λόγω του ότι μέσα στην printf() εμφανίζεται μια μεταβλητή η οποία δεν έχει αρχικοποιηθεί ο μεταγλωττιστής θα το αντιληφθεί και θα ορίσει ένα σφάλμα κατά τη μεταγλώττιση του προγράμματος.

#### 27. Τι εννοούμε με τους όρους LIFO (Last In First Out) και FIFO (First In First Out) και σε ποιες δομές δεδομένων βρίσκουν εφαρμογή;

Με τους όρους **LIFO (Last In First Out)** και **FIFO (First In First Out)** αναφερόμαστε στη διαχείριση των δεδομένων αναφορικά με τον τρόπο με τον οποίο αυτά αποθηκεύονται (εισάγονται) μέσα σε μια δομή δεδομένων και εν συνεχεία διαχειρίζονται (εξάγονται) από αυτήν. Πιο συγκεκριμένα, για τις δομές δεδομένων Στοίβα και Ούρα, όπου και οι ανωτέρω τεχνικές εισαγωγής και επεξεργασίας δεδομένων βρίσκουν εφαρμογή, ισχύουν τα παρακάτω.

Στη στοίβα τα δεδομένα που βρίσκονται στην κορυφή τα επεξεργαζόμαστε πρώτα και τα δεδομένα που βρίσκονται στο τέλος τα επεξεργαζόμαστε τελευταία. Η μέθοδος αυτή ονομάζεται LIFO (Last-In-First-Out).

Σε μια ουρά εξυπηρετούνται εκείνα τα δεδομένα που τοποθετήθηκαν στην ουρά πρώτα από όλα τα υπόλοιπα. Η μέθοδος αυτή ονομάζεται FIFO (First-In-First-Out). Σε αυτή τη δομή δεδομένων επιτρέπεται όλες οι εισαγωγές στοιχείων να γίνονται από το ένα άκρο, και όλες οι διαγραφές,



από το άλλο. Κατά συνέπεια χρειαζόμαστε δύο δείκτες τον μπρος (head) και τον πίσω(tail).

**28. Τι είναι οι στατικές και τι οι δυναμικές δομές δεδομένων; Ποιες οι διαφορές τους;**

Οι δομές δεδομένων διακρίνονται σε δύο μεγάλες κατηγορίες: τις *στατικές* (static) και τις *δυναμικές*(dynamic).

Οι δυναμικές δομές δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης αλλά στηρίζονται στην τεχνική της λεγόμενης *δυναμικής παραχώρησης μνήμης* (dynamic memory allocation). Οι δυναμικές δομές δεν έχουν σταθερό μέγεθος, αλλά ο αριθμός των κόμβων τους μεγαλώνει και μικραίνει καθώς στη δομή εισάγονται νέα δεδομένα ή διαγράφονται κάποια δεδομένα αντίστοιχα.

Από την άλλη πλευρά στις στατικές δομές δεδομένων ορίζεται “εκ προοιμίου” το ακριβές μέγεθος της απαιτούμενης κύριας μνήμης (καθορίζεται κατά τη στιγμή του προγραμματισμού τους) κατά τη στιγμή της μετάφρασής τους και όχι κατά τη στιγμή της εκτέλεσής τους προγράμματος.

Επιπρόσθετα, εν αντιθέσει με τις δυναμικές δομές τα στοιχεία τα οποία τοποθετούνται σε στατικές δομές αποθηκεύονται σε συνεχόμενες θέσεις μνήμης.

**29. Να γραφεί πρόγραμμα σε C το οποίο θα δέχεται από το πληκτρολόγιο έναν ακέραιο. Εν συνεχεία θα υπολογίζει και θα εκτυπώνει το παραγοντικό αυτού του αριθμού. Για παράδειγμα με είσοδο 5 το πρόγραμμα θα υπολογίζει το  $5! = 1 * 2 * 3 * 4 * 5 = 120$  και θα εκτυπώνει στην οθόνη το 120.**

```
#include <stdio.h>
int main() {
    int n, i;
    unsigned long long factorial = 1;
    printf("Enter an integer: ");
    scanf("%d", &n);
    if (n < 0)
        printf("ERROR - Negative Number");
    else{
        for(i=1; i<=n; ++i){
            factorial = factorial*i;
        }
        printf("Factorial of %d = %llu", n, factorial);
    }
    return 0;
}
```

**30. Αναφέρατε πέντε, από τους βασικούς τύπους δεδομένων (Data Types) της Pascal ή C. Να αναφέρετε τα όριά τους (κατά προσέγγιση και όπου αυτά υπάρχουν) καθώς και ένα παράδειγμα από τον κάθε τύπο.**

Ένας τύπος δεδομένων είναι ένα σύνολο τιμών και ένα σύνολο λειτουργιών (πράξεων) που μπορούν να εφαρμοστούν σε αυτές τις τιμές. Ορίζουμε λοιπόν δύο βασικές κατηγορίες τύπων

δεδομένων, αναφορικά, τους βασικούς τύπους δεδομένων (char, int, float, double), και τους σύνθετους τύπους δεδομένων “structs” (τύποι δεδομένων δηλαδή, που συντίθενται από συνδυασμό δύο ή περισσότερων βασικών τύπων δεδομένων π.χ. `struct Time { int hours; int minutes; int seconds;};`). Στη συνέχεια παρατίθεται μια σύντομη αναφορά στους βασικούς τύπους δεδομένων της C.

### Βασικοί τύποι δεδομένων στη C

- *int* (ακέραιος – integer)

Αναπαριστά ακέραιους αριθμούς, έχει μέγεθος 4 bytes και πεδίο τιμών -231.. +(231-1) δηλαδή περίπου : από [-2G..+2G] [-(δυο δισεκατομμύρια .. +δυο δισεκατομμύρια]

Οι επιτρεπόμενες πράξεις είναι οι εξής: πρόσθεση (+), αφαίρεση (-), πολλαπλασιασμός (\*), διαίρεση (/) κλπ. Παράδειγμα `int x=4;`

- *float* (κινητής υποδιαστολής απλής ακρίβειας – floating point)

Αναπαριστά τους πραγματικούς αριθμούς, και έχει μέγεθος 4 bytes.

Οι επιτρεπόμενες πράξεις είναι οι εξής:: πρόσθεση (+), αφαίρεση (-), πολλαπλασιασμός (\*), διαίρεση (/). Παράδειγμα `float z=2.7;`

- *double* (κινητής υποδιαστολής διπλής ακρίβειας – double precision)

Τύπος δεδομένων παρόμοιος με τον τύπο δεδομένων με float αλλά με μεγαλύτερη ακρίβεια (περιέχει δηλ. διπλό αριθμό δεκαδικών ψηφίων απ’ ότι η μεταβλητή float). Λόγω της μεγαλύτερης ακρίβειας έχει μέγεθος 8 bytes. Παράδειγμα `double pi=3,14159265;`

- *char* (χαρακτήρας – character)

Αναπαριστά ατομικούς χαρακτήρες (A-Z, a-z, 0-9, !@\$\_%&#, ειδικά σύμβολα \n, κλπ.) και έχει μέγεθος 1 byte. Κυριολεκτικές τιμές εσωκλείονται σε αποστρόφους, π.χ. 'A', 'a', '9', "'", '\'', '\\*', '\n', '\", κτλ, και κάθε χαρακτήρας αντιστοιχεί σε ένα μοναδικό κωδικό όπου για τη C χρησιμοποιείται ο ASCII code. Παράδειγμα `char letter='A';`

31. Τι ονομάζουμε **recursion** (αναδρομή); α) Να γραφεί αναδρομική συνάρτηση (σε C) με όνομα **power** που να δέχεται σαν παραμέτρους δυο ακέραιους αριθμούς (a και b) και να επιστρέφει το αποτέλεσμα **ab**. Ισχύει:  $ab = a * a * a * \dots * a$  (b φορές) π.χ.  $3^5 = 3 * 3 * 3 * 3 * 3$  (5 φορές) β) Περιγράψτε σχηματικά πως δουλεύει για  $a=3$  και  $b=5$

Η αναδρομή είναι στην ουσία μια στρατηγική επίλυσης προβλημάτων και πιο συγκεκριμένα μια τεχνική προγραμματισμού σύμφωνα με την οποία ένα σύνθετο πρόβλημα ανάγεται σε ένα απλούστερο της ίδιας μορφής. Σε κάθε περίπτωση όπου χρησιμοποιείται αναδρομή, είναι δυνατόν, εναλλακτικά, να χρησιμοποιηθεί επανάληψη. Η χρήση της αναδρομής προσφέρει, σε κάποιες περιπτώσεις, απλούστερες λύσεις. Η αναδρομή στη C πραγματοποιείται ορίζοντας συναρτήσεις που καλούν τον εαυτό τους (άμεση αναδρομή).

```

#include <stdio.h>
int power(int n1, int n2);
int main() {
    int a, b, result;
    printf("Enter base number: ");
    scanf("%d",&a);
    printf("Enter power number(positive integer): "); scanf("%d",&b);
    result = power(a, b);
    printf("%d^%d = %d", a, b, result);
    return 0;
}

int power(int a, int b) {
    if (b != 0){
        return (a*power(a, b-1));
    }
    else {
        return 1;
    }
}

```

Στην συνέχεια παρατίθεται μια σχηματική επεξήγηση της λειτουργίας του κώδικα.

```

#include <stdio.h>
int power(int n1, int n2);    Pass Values to the Variables a,b
int main() {
    int a, b, result;
    printf("Enter base number: ");
    scanf("%d",&a);
    printf("Enter power number(positive integer): ");
    scanf("%d",&b);
    result = power(a, b);
    printf("%d^%d = %d", a, b, result);
    return 0;
}

int power(int a, int b) {
    if (b != 0){
        return (a*power(a, b-1));
    }
    else {
        return 1;
    }
}

```

*Call function "power()" set as input the Variables a = 3 and b = 5 as to compute  $a^b$ :  $3^5$*

*"Recursively" call the function "power()" with argument a = 3 and b = b - 1 each time called it computes:*

- 1)  $a = 3, b = 4 \rightarrow 3 \times 3^4$
- 2)  $a = 3, b = 3 \rightarrow 3 \times 3^3$
- 3)  $a = 3, b = 2 \rightarrow 3 \times 3^2$
- 4)  $a = 3, b = 1 \rightarrow 3 \times 3^1$

32. Να γραφεί αλγόριθμος ο οποίος να διαβάζει τα ονόματα 10 πωλητών μιας εταιρείας και να τα αποθηκεύει σε ένα πίνακα A, και τις αντίστοιχες πωλήσεις αυτών των πωλητών το 2005 και να τις αποθηκεύει σε ένα πίνακα B. Ο αλγόριθμος πρέπει να εκτυπώνει: α) Το μέσο όρο των πωλήσεων όλων των πωλητών, β) Το όνομα του πωλητή με τις μέγιστες πωλήσεις, γ) Το όνομα του πωλητή με τις μικρότερες πωλήσεις

Algorithm sellers

INPUT: String Array sellers, Integer Array sales

OUTPUT: mean\_sales, seller\_with\_max\_sales,seller\_with\_min\_sales

BEGIN

int A[] //seller names

int B[] //seller sales

Integers counter,max,min,position;

Float mean;

for i=1:1:10

print("enter seller credentials:")

A[i]=read(name);

print("enter seller sale:")

B[i]=read(sales);

end

//----- FIND MEAN SALES -----

for i=1:1:10

counter=counter+B[i];

end

mean=counter/10;

print("The mean sales are %d", mean);

//----- FIND SELLER WITH MAX SALES -----

max=B[0]

for i=1:1:10

if B[i]> max

max=B[i];

position=i;

endif

end

print("The seller with the max sales is %s", A[position]);

//----- FIND SELLER WITH MAX SALES -----

min=B[0]

for i=1:1:10

if B[i]< min

min=B[i];

position=i;

endif

end

print("The seller with the min sales is %s", A[position]);

END

**33. Α. Εξηγήστε σύντομα τι κάνει το ακόλουθο πρόγραμμα:**

```
#include <stdio.h>
#include <math.h>
int main() {
int n; for (n=50;n!=60;n++) printf("%d \t %f \n",n,log(n));
return 0;
}
```

**B. Αν στην for αντικαταστήσουμε τον αριθμό 60 με τον αριθμό 40 τι θα συμβεί;**

Εκτυπώνεται ο λογάριθμος των τιμών από 50 έως 59;

Προφανώς αν αντικατασταθεί το άνω φράγμα του βρόχου με μια τιμή η οποία δεν πρόκειται να τη φτάσουμε ποτε τότε δημιουργείται ατέρμων βρόχος. Οι υπολογισμοί που θα γίνουν είναι ο υπολογισμός των λογαρίθμων των τιμών στο εύρος [50, +∞).

**34. Στον ακόλουθο κώδικα τι θα εκτυπώσει η printf αν δώσουμε από το πληκτρολόγιο τις τιμές: -3 και 7 όταν μας ζητηθεί να εισάγουμε τιμές;**

```
int n=-3,i,k;
float x=3.,y;
printf("Dwse times: ");
scanf("%d%f",&i,&y);
k=3/5;
x=++n;
x*=i;
n=-k;
y=n%i;
printf("%d\t%d\t%d\t%f\t%f\n",n,i,k,x,y);
```

n=-1

i=-3

k=-1

x=6.000000

y=-1.000000

**35. Να γραφεί πρόγραμμα που να δέχεται ένα διψήφιο ακέραιο αριθμό, και να επιστρέφει τον αριθμό αναστραμμένο καθώς και το άθροισμα των ψηφίων του αριθμού (αν π.χ. ο χρήστης δώσει τον αριθμό 53, το πρόγραμμα πρέπει να επιστρέφει 35 και 8). Σημείωση: το πρόγραμμα πρέπει να ελέγχει αν ο αριθμός είναι διψήφιος και για την περίπτωση που δεν είναι να βγάζει μήνυμα λάθους.**

```
#include <stdio.h>

int main(){
    int num;
    int number;
    int count = 0;

    while(count!=2){
```

```

        printf("Enter an integer: ");
        scanf("%d", &num);
        number=num;
        count=0;
        while(number != 0){
            number /= 10;
            count++;
        }
        printf("Number of digits: %d\n\n\n", count);
        if(count!=2){
            printf("ERROR NOT A 2-Digit Number\n");
            count=0;
        }
    }
    reverseDigits(num);
    getchar();
    return 0;
}

```

```

int reverseDigits(int num){
    int number=num;
    printf("The number is: %d \n",num);
    int rev_num = 0;
    while(num > 0){
        rev_num = rev_num*10 + num%10;
        num = num/10;
    }
    printf("The revised is: %d \n",rev_num);
    int a=number/10;
    int b=number%10;
    int c=a+b;
    printf("the sum is: %d",c);
    return rev_num;
}

```

### 36. Να γραφεί σε C η διαδικασία της γρήγορης ταξινόμησης (quick sort).

```

#include<stdio.h>

int main(){
    int arr[] = {10, 7, 1, 5, 8, 9};
    int n = sizeof(arr)/sizeof(arr[0]);
    printf("\n Initial Array: ");
    printArray(arr, n);
    quickSort(arr, 0, n-1);
    printf("\n \n Sorted Array: ");
    printArray(arr, n);
}

```

```
        return 0;
    }

    void swap(int* a, int* b){
        int t = *a;
        *a = *b;
        *b = t;
    }

    int partition (int arr[], int low, int high){
        int pivot = arr[high];
        int i = (low - 1);
        int j;
        for (j = low; j <= high- 1; j++) {
            if (arr[j] <= pivot){
                i++;
                swap(&arr[i], &arr[j]);
            }
        }
        swap(&arr[i + 1], &arr[high]);
        return (i + 1);
    }

    void quickSort(int arr[], int low, int high){
        if (low < high){
            int pi = partition(arr, low, high);
            quickSort(arr, low, pi - 1);
            quickSort(arr, pi + 1, high);
        }
    }

    void printArray(int arr[], int size){
        int i;
        for (i=0; i < size; i++){
            printf("%d ", arr[i]);
        }
    }
```

- 37. Να ταξινομήσετε τον παρακάτω πίνακα με τις τεχνικές selection sort και bubble sort, περιγράφοντας αναλυτικά το κάθε βήμα της ταξινόμησης. 6 7 9 3 5 8**

Sorting with Selection-Sort

initial: 6, 7, 9, 3, 5, 8,

Step 1): 3, 7, 9, 6, 5, 8,

Step 2): 3, 5, 9, 6, 7, 8,

Step 3): 3, 5, 6, 9, 7, 8,

Step 4): 3, 5, 6, 7, 9, 8,

Step 5): 3, 5, 6, 7, 8, 9,

final: 3, 5, 6, 7, 8, 9,

#### Sorting with Bubble-Sort

initial: 6, 7, 9, 3, 5, 8,

Step 1): 6, 7, 3, 5, 8, 9,

Step 2): 6, 3, 5, 7, 8, 9,

Step 3): 3, 5, 6, 7, 8, 9,

Step 4): 3, 5, 6, 7, 8, 9,

Step 5): 3, 5, 6, 7, 8, 9,

Step 6): 3, 5, 6, 7, 8, 9,

Step 7): 3, 5, 6, 7, 8, 9,

final: 3, 5, 6, 7, 8, 9,

38. Να γραφεί πρόγραμμα σε C που να αποθηκεύει ακέραιους αριθμούς από το πληκτρολόγιο σε έναν πίνακα διαστάσεων 5x5 και κατόπιν να εκτυπώνει: α. όλα τα στοιχεία του πίνακα β. όλα τα στοιχεία της κύριας διαγωνίου του πίνακα γ. όλα τα στοιχεία της αναστροφής διαγωνίου του πίνακα

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

```
int main() {
    int i,j, element;
    int n=5;
    int A[n][n];
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            printf("Give the element A[%d][%d]: ",i,j);
            scanf("%d",&element);
            A[i][j]=element;
            printf("\n");
        }
    }
    printf("\nThe Matrix A is: \n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            printf("%d,",A[i][j]);
        }
    }
}
```



```

        }
        printf("\n");
    }
    printf("\nThe diagonal elements of Matrix A are:");
    for(i=0;i<n;i++){
        printf("%d, ",A[i][i]);
    }
    printf("\nThe reverse-diagonal elements of Matrix A are: ");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            if(i+j==n-1){
                printf("%d, ",A[i][j]);
            }
        }
    }
    return 0;
}

```

### 39. Τι εννοούμε με τους όρους "Αλγόριθμος" και "Πρόγραμμα";

Ως αλγόριθμος ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος. Πιο απλά (αλγόριθμο) ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και έχουν ως σκοπό την επίλυση κάποιου προβλήματος.

Ένα πρόγραμμα υπολογιστών είναι ουσιαστικά ένας αλγόριθμος που λέει στον υπολογιστή ποια συγκεκριμένα βήματα να εκτελέσει (σε ποια συγκεκριμένη σειρά) προκειμένου να επιτευχθεί ένας συγκεκριμένος στόχος, όπου οι εντολές αυτές έχουν μια αυστηρά συγκεκριμένη δομή ανάλογα με τη γλώσσα προγραμματισμού στην οποία υλοποιείται το πρόγραμμα.

### 40. Τι ονομάζουμε δομή δεδομένων;

Ένα σύνολο από στοιχεία δεδομένων αποτελεί δομή όταν υπάρχουν καθορισμένες σχέσεις μεταξύ των στοιχείων. Μια δομή δεδομένων ορίζεται ως η διαδικασία εισαγωγής και απομάκρυνσης στοιχείων με τρόπο ώστε όλη η δομή να μην αλλοιώνεται. Κάθε δομή δεδομένων έχει ως αφηρημένη έννοια συγκεκριμένο ορισμό, δηλαδή διαδικασία εισαγωγής/απομάκρυνσης στοιχείων, αλλά μπορεί να υλοποιείται σε έναν Η/Υ με διαφορετικούς τρόπους.

Η έννοια της δομής δεδομένων αναφέρεται στους διαφορετικούς δυνατούς τρόπους οργάνωσης και αποθήκευσης δεδομένων μέσα σε έναν υπολογιστή, ώστε τα δεδομένα αυτά να μπορούν να χρησιμοποιηθούν αποδοτικά. Για παράδειγμα ένα σύνολο από δεδομένα μπορεί να αποθηκευτεί σε δομή πίνακα, στοίβας, συνδεδεμένης λίστας, σωρού, ουράς και ούτω καθ' εξής.

**41. Σε ποιες κατηγορίες διακρίνονται τα προβλήματα με κριτήριο το είδος επίλυσης τους;**

- a) Προβλήματα Απόφασης: Η λύση σε αυτά τα προβλήματα είναι του τύπου «Ναι» και «Όχι».
- b) Υπολογιστικά Προβλήματα: Για την επίλυση απαιτείται η διενέργεια υπολογισμών.
- c) Προβλήματα Βελτιστοποίησης: Η λύση που ζητάμε είναι το βέλτιστο αποτέλεσμα για τα συγκεκριμένα δεδομένα.

**42. Ποιες είναι οι βασικές λειτουργίες (ή πράξεις) επί των δομών δεδομένων;**

- a) Προσπέλαση: πρόσβαση σε ένα κόμβο με σκοπό να εξετασθεί ή να τροποποιηθεί το περιεχόμενό του.
- b) Εισαγωγή: προσθήκη νέων κόμβων σε μία υπάρχουσα δομή.
- c) Διαγραφή: αποτελεί το αντίστροφο της εισαγωγής, δηλαδή ένας κόμβος αφαιρείται από μία δομή.
- d) Αναζήτηση: προσπελούνται οι κόμβοι μιας δομής, προκειμένου να εντοπιστούν ένας ή περισσότεροι που έχουν μια δεδομένη ιδιότητα.
- e) Ταξινόμηση: οι κόμβοι μιας δομής διατάσσονται κατά αύξουσα ή φθίνουσα σειρά.
- f) Αντιγραφή: όλοι οι κόμβοι ή μερικοί από τους κόμβους μίας δομής αντιγράφονται σε μία άλλη δομή.
- g) Συγχώνευση: δύο ή περισσότερες δομές συνενώνονται σε μία ενιαία δομή.
- h) Διαχωρισμός: αποτελεί την αντίστροφη πράξη της συγχώνευσης.

**43. Τι εννοούμε με τον όρο "πρόβλημα" και σε τι αναφερόμαστε με τον όρο "Ανάλυση Προβλήματος";**

Με τον όρο Πρόβλημα εννοούμε μία κατάσταση που χρήζει αντιμετώπισης (επίλυσης), η δε λύση της δεν είναι γνωστή και ούτε προφανής.

Με τον όρο Ανάλυση Προβλήματος εννοούμε ότι ξεκινάμε να αποκαλύπτουμε τη δομή του προβλήματος, δηλαδή να χωρίσουμε το πρόβλημα σε μικρότερα και απλούστερα υπο-προβλήματα (υποδιαρέσεις του ιδίου προβλήματος), καθένα από τα οποία λύνεται ευκολότερα.

Στόχος μας λοιπόν είναι να αναλύσουμε τα συστατικά μέρη από τα οποία συντίθεται το πρόβλημα, δηλαδή τα επιμέρους τμήματα που το αποτελούν καθώς και τον τρόπο με τον οποίο αυτά συνδέονται μεταξύ τους αποτυπώνοντας εν τέλει τη *δομή του προβλήματος*.

Έτσι, η ανάλυση μπορεί να γίνει με δομημένο λεκτικό τρόπο ή καλύτερα χρησιμοποιώντας ένα ιεραρχικό διάγραμμα.

**44. Να αναφέρετε τις πιο συνηθισμένες τεχνικές σχεδίασης αλγορίθμων.**

Οι πιο συνηθισμένες τεχνικές είναι η διαίρει και βασίλευε, ο δυναμικός προγραμματισμός και η άπληστη μέθοδος.

Στη *διαίρει και βασίλευε* το αρχικό σύνθετο πρόβλημα διασπάται σε μικρότερα επιμέρους προβλήματα της ίδιας φύσης με το αρχικό αλλά μικρότερα σε μέγεθος. Ένα παράδειγμα τέτοιας μεθόδου είναι η δυαδική αναζήτηση στην οποία η αναζήτηση ενός στοιχείου σε έναν πίνακα ανάγεται στην αναζήτηση του στοιχείου στο πάνω ή κάτω μισό του πίνακα. Αυτή η προσέγγιση εντάσσεται στη top-down μεθοδολογία, από πάνω προς τα κάτω δηλαδή από το σύνθετο στο απλό.

Στο *δυναμικό προγραμματισμό* αρχικά επιλύονται μικρότερα σε μέγεθος άρα απλούστερα στιγμιότυπα του αρχικού προβλήματος και μέσω της σύνθεσης αυτών επιχειρείται η επίλυση του αρχικού συνθετότερου προβλήματος.

Στην *άπληστη μέθοδο* επιχειρείται σε κάθε βήμα επίλυσης του προβλήματος η επιλογή που φαίνεται καλύτερη εκείνη τη στιγμή χωρίς να γίνεται προσπάθεια συνολικής θεώρησης του προβλήματος. Σε πολλές περιπτώσεις αυτή η προσέγγιση οδηγεί στην καλύτερη δυνατή λύση.

**45. Ποια είναι τα χαρακτηριστικά ενός αλγορίθμου;**

- a) Είσοδος. Καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο. Η περίπτωση που δεν δίνονται τιμές δεδομένων εμφανίζεται, όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών
- b) Έξοδος. Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή δεδομένων ως αποτέλεσμα προς το χρήστη ή προς έναν άλλο αλγόριθμο.
- c) Καθοριστικότητα. Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της. Λόγου χάριν, μία εντολή διαίρεσης πρέπει να θεωρεί και την περίπτωση, όπου ο διαιρέτης λαμβάνει μηδενική τιμή.
- d) Περαιτότητα. Ο αλγόριθμος να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μία διαδικασία που δεν τελειώνει μετά από ένα συγκεκριμένο αριθμό βημάτων δεν αποτελεί αλγόριθμο, αλλά λέγεται απλά υπολογιστική διαδικασία (computational procedure).
- e) Αποτελεσματικότητα. Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μία εντολή δεν αρκεί να έχει οριστεί, αλλά πρέπει να είναι και εκτελέσιμη.

**46. Με ποιους τρόπους πραγματοποιείται η περιγραφή ενός αλγορίθμου;**

- a) Ελεύθερο Κείμενο. Αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρουσίασης αλγορίθμου. Έτσι εγκυμονεί τον κίνδυνο ότι μπορεί εύκολα να οδηγήσει σε μη εκτελέσιμη παρουσίαση παραβιάζοντας το τελευταίο χαρακτηριστικό των αλγορίθμων, δηλαδή την αποτελεσματικότητα.
- b) Διαγραμματικές Τεχνικές. Συνιστούν ένα γραφικό τρόπο παρουσίασης του αλγορίθμου. Από τις διάφορες διαγραμματικές τεχνικές που έχουν επινοηθεί, η πιο παλιά και η πιο γνωστή ίσως, είναι το διάγραμμα ροής (flow chart). Ωστόσο η χρήση διαγραμμάτων ροής για την παρουσίαση αλγορίθμων δεν αποτελεί την καλύτερη λύση, γι'αυτό και εμφανίζονται όλο και σπανιότερα στη βιβλιογραφία και στην πράξη.
- c) Φυσική Γλώσσα (κατά βήματα). Στην περίπτωση αυτή χρειάζεται προσοχή, γιατί μπορεί να παραβιασθεί το τρίτο βασικό χαρακτηριστικό ενός αλγορίθμου, όπως προσδιορίσθηκε προηγουμένως, δηλαδή το κριτήριο του καθορισμού.
- d) Κωδικοποίηση . Με ένα πρόγραμμα γραμμένο είτε σε μία ψευδογλώσσα είτε σε κάποια γλώσσα προγραμματισμού που όταν εκτελεσθεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο.

**47. Ποια είναι τα χαρακτηριστικά που είναι απαραίτητα προκειμένου να θεωρήσουμε έναν αλγόριθμο πλήρη;**

- a) Είσοδος. Καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο. Η περίπτωση που δεν δίνονται τιμές δεδομένων εμφανίζεται, όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών
- b) Έξοδος. Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή δεδομένων ως αποτέλεσμα προς το χρήστη ή προς έναν άλλο αλγόριθμο.
- c) Καθοριστικότητα. Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της. Λόγου χάριν, μία εντολή διαίρεσης πρέπει να θεωρεί και την περίπτωση, όπου ο διαιρέτης λαμβάνει μηδενική τιμή.
- d) Περαιτότητα. Ο αλγόριθμος να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μία διαδικασία που δεν τελειώνει μετά από ένα συγκεκριμένο αριθμό βημάτων δεν αποτελεί αλγόριθμο, αλλά λέγεται απλά υπολογιστική διαδικασία (computational procedure).
- e) Αποτελεσματικότητα. Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μία εντολή δεν αρκεί να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.

**48. Να αναφέρετε πόσα είναι τα είδη της δομής επιλογής.**

Η διαδικασία της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (Αληθής ή Ψευδής) και ακολουθεί η απόφαση εκτέλεσης κάποιας ενέργειας με βάση την τιμή της λογικής αυτής συνθήκης. Στη συνέχεια δίνονται δύο παραδείγματα ενεργειών με βάση κάποια συνθήκης επιλογής. Το πρώτο παράδειγμα αφορά στην εκτέλεση κάποιας ενέργειας όταν η συνθήκη είναι Αληθής, ενώ το δεύτερο παράδειγμα αφορά στην εκτέλεση μίας ενέργειας όταν η συνθήκη είναι Αληθής και κάποιας άλλης ενέργειας όταν η συνθήκη είναι Ψευδή.

**49. Να αναφέρετε πόσα είναι τα είδη της δομής επανάληψης.**

- a) Επαναληπτικό σχήμα με έλεγχο επανάληψης στην αρχή
- b) Επαναληπτικό σχήμα με έλεγχο επανάληψης στο τέλος
- c) Επαναληπτικό σχήμα ορισμένων φορών επανάληψης

**50. Στη δομή δεδομένων τι είναι η στοίβα; Δώστε ένα παράδειγμα.**

Η στοίβα είναι μία δομή δεδομένων στην οποία τα δεδομένα εισάγονται το ένα μετά το άλλο ενώ ο χρήστης έχει τη δυνατότητα να προσπελάσει μόνο το τελευταίο που έχει εισαχθεί ενώ το πρώτο στοιχείο που εισήχθη θα προσπελαστεί τελευταίο. Είναι μία δομή LIFO (Last In First Out). Η υλοποίηση μίας στοίβας μπορεί να γίνει απλούστερα χρησιμοποιώντας έναν μονοδιάστατο πίνακα χρησιμοποιώντας πάντοτε έναν δείκτη που να δείχνει το πρώτο στοιχείο της στοίβας που είναι και το τελευταίο που εισήχθη σ' αυτήν.

Οι βασικές πράξεις σε μια στοίβα είναι:

- a) Empty (), επιστρέφει 1 αν η στοίβα είναι άδεια ή 0 στην αντίθετη περίπτωση.
- b) Push (a), τοποθετεί το στοιχείο a στην κορυφή της στοίβας
- c) Top (), προσπελαύνει το στοιχείο που βρίσκεται στην κορυφή της στοίβας.
- d) Pop (), προσπελαύνει και διαγράφει το στοιχείο που βρίσκεται στην κορυφή της στοίβας.

**51. Να αναφέρετε τα είδη της αναζήτησης και της ταξινόμησης των στοιχείων ενός πίνακα.**

Αναζήτηση σε πίνακες

Σειριακή Αναζήτηση ή Γραμμική Αναζήτηση

Διαδική Αναζήτηση

Αναζήτηση με Παρεμβολή

Ταξινόμηση σε Πίνακες

Ταξινόμηση με Εισαγωγή

Ταξινόμηση Φυσαλίδας

Ταξινόμηση Με Επιλογή

Ταξινόμηση Σωρού

Ταξινόμηση Με Συγχώνευση

Ταχεία Ταξινόμηση

Ταξινόμηση με διαμερισμό και αντιμετάθεση

## 52. Τι είναι RAM και πώς είναι δομημένη;

Γενικότερα με τον όρο **RAM** αναφερόμαστε στην κύρια ή κεντρική μνήμη ενός υπολογιστικού συστήματος αρχιτεκτονικής φον Νόιμαν, δηλαδή τη μνήμη στην οποία αποθηκεύονται προγράμματα και δεδομένα, προκειμένου είτε να εκτελεστούν είτε να υποστούν επεξεργασία αντίστοιχα.

Η Μνήμη τυχαίας προσπέλασης (RAM, Random access memory) είναι όρος που χρησιμοποιούμε για να αναφερθούμε σε ηλεκτρονικές διατάξεις προσωρινής αποθήκευσης ψηφιακών δεδομένων (μνήμης υπολογιστή), οι οποίες επιτρέπουν πρόσβαση στα αποθηκευμένα δεδομένα στον ίδιο χρόνο οπουδήποτε και αν βρίσκονται αυτά, δηλαδή με «τυχαία πρόσβαση». Σε αντιδιαστολή βρίσκονται συσκευές αποθήκευσης δεδομένων, όπως οι μαγνητικές ταινίες, οι μαγνητικοί δίσκοι («σκληροί» ή «εύκαμπτοι»), στα οποία η πρόσβαση στα δεδομένα μπορεί να γίνει μόνο με κάποιον προκαθορισμένο τρόπο, συνήθως σειριακά, λόγω του τρόπου κατασκευής τους. Υπάρχουν δύο βασικοί τύποι RAM : η δυναμική RAM (DRAM) και η στατική RAM (SRAM). Η DRAM είναι η πιο κοινή μορφή αλλά πρέπει να «ανανεώνεται» (refresh) χιλιάδες φορές ανά δευτερόλεπτο, ενώ η SRAM δεν χρειάζεται κάτι τέτοιο. Η SRAM, ως διάταξη, είναι πιο δαπανηρή στην κατασκευή της - και επομένως στην αγορά της - σε σχέση με την DRAM.

## 53. Πόσα είδη καταχωρητών υπάρχουν; Να αναφερθούν τουλάχιστον τρεις χαρακτηριστικοί καταχωρητές γενικής χρήσης.

Η ΚΜΕ περιέχει πολλά είδη καταχωρητών, τα οποία ταξινομούνται ανάλογα με το περιεχόμενό τους ή ανάλογα με τις διαδικασίες που επιτελούν.

- Καταχωρητές Προσπελάσιμοι από τον Χρήστη (User Accessible Registers). Χωρίζονται σε Καταχωρητές Δεδομένων και Καταχωρητές Διευθύνσεων. Στους Καταχωρητές Δεδομένων (Data Registers) αποθηκεύονται αριθμητικές τιμές. Στους Καταχωρητές Διευθύνσεων (Address Registers) αποθηκεύονται οι διευθύνσεις των αποθηκευμένων δεδομένων.
- Καταχωρητές Συνθήκης (Conditional Registers). Στους Καταχωρητές Συνθήκης αποθηκεύονται λογικές τιμές.
- Καταχωρητές Γενικής Χρήσης (General Purpose Registers). Στους Καταχωρητές Γενικής Χρήσης μπορούν να αποθηκευθούν συνδυασμοί δεδομένων και διευθύνσεων δεδομένων.
- Καταχωρητές Κινητής Υποδιαστολής (Floating point registers). Εδώ αποθηκεύονται αριθμοί κινητής υποδιαστολής.
- Σταθεροί Καταχωρητές (Constant Registers). Στους Σταθερούς Καταχωρητές αποθηκεύονται τιμές που παραμένουν σταθερές, όπως το 0, το 1 και το  $\pi$  (3,14..).

Τρεις χαρακτηριστικοί καταχωρητές είναι οι εξής:

- Ο \$2 ή \$v0 είναι ο καταχωρητής που επιστρέφει τις τιμές από τις συναρτήσεις που εκτελούνται στο πρόγραμμα.
- Ο \$4 ή \$a0 είναι ο καταχωρητής όπου αποθηκεύεται προσωρινά η μεταβλητή μιας συνάρτησης – ρουτίνας που πρόκειται να εκτελεστεί.
- Ο \$6 ή \$t0 είναι ο καταχωρητής που χρησιμοποιείται για προσωρινή αποθήκευση δεδομένων του βασικού προγράμματος ή ενός υποπρογράμματος.



**54. Τι είναι το bit, το byte και το word; Τι αναπαριστά το καθένα; Πως συνδέονται μεταξύ τους;**

Το δυαδικό ψηφίο (bit) είναι η στοιχειώδης μονάδα πληροφορίας στην Επιστήμη Υπολογιστών. Ένα bit είναι η μικρότερη δυνατή ποσότητα πληροφορίας που μπορεί να αποθηκευτεί από μία δυαδική συσκευή, ή από άλλο φυσικό σύστημα το οποίο μπορεί να υπάρχει σε μία από δύο διακριτές καταστάσεις (π.χ. 0 και 1).

Το byte (μπάιτ) είναι μονάδα μέτρησης ποσότητας πληροφορίας στα υπολογιστικά συστήματα, εμφανιζόμενη συνήθως στα διάφορα επίπεδα της ιεραρχίας μνήμης τους. Ένα byte ισοδυναμεί με 8 bit. Το byte μπορεί να αντιπροσωπεύει τιμές από 0 έως και 255 στο δεκαδικό σύστημα.

Το word στις γλώσσες προγραμματισμού της επιστήμης της πληροφορικής αναφέρεται σε μια αριθμοσειρά, η οποία χρησιμοποιείται ως μεταβλητή δυναμικότητας 32 bit, δηλαδή 4 Byte.

**55. Να αναφέρετε τρία διαφορετικά είδη διευθυνσιοδότησης και δώστε σύντομα παραδείγματα.**

**Άμεση Διευθυνσιοδότηση**

Η απλούστερη μορφή διευθυνσιοδότησης είναι η άμεση διευθυνσιοδότηση, όπου ο τελεστέος στην πραγματικότητα βρίσκεται στην εντολή. Το πλεονέκτημα της άμεσης διευθυνσιοδότησης είναι ότι, για να πάρουμε τον τελεστέο, δεν χρειάζεται άλλη αναφορά μνήμης εκτός από την προσαγωγή της εντολής, πράγμα που εξοικονομεί μια περίοδο μνήμης ή μνήμης cache στον κύκλο εντολής. Το μειονέκτημα είναι ότι το μέγεθος του αριθμού περιορίζεται στο μέγεθος του πεδίου διεύθυνσης το οποίο, στις περισσότερες ομάδες εντολών, είναι μικρό σε σύγκριση με το μήκος λέξης.

**Απευθείας Διευθυνσιοδότηση**

Μια πολύ απλή μορφή διευθυνσιοδότησης είναι η απευθείας διευθυνσιοδότηση, όπου το πεδίο διεύθυνσης περιέχει την ενεργό διεύθυνση του τελεστέου. Ο προφανής περιορισμός είναι ότι προσφέρει μόνο περιορισμένο χώρο διεύθυνσης.

**Έμμεση Διευθυνσιοδότηση**

Στην απευθείας διευθυνσιοδότηση, το μήκος του πεδίου διεύθυνσης είναι συνήθως μικρότερο από το μήκος λέξης, πράγμα που περιορίζει το πεδίο τιμών της διεύθυνσης. Μια λύση είναι να κάνουμε το πεδίο διεύθυνσης να αναφέρεται στην διεύθυνση μιας λέξης στην μνήμη, η οποία με την σειρά της θα περιέχει μια διεύθυνση πλήρους μήκους του τελεστέου. Αυτό είναι γνωστό ως έμμεση διευθυνσιοδότηση (indirect addressing).

**56. Από τι εξαρτάται η τεχνική διαχείρισης της μνήμης ενός υπολογιστή;**

Η τεχνική διαχείρισης μνήμης που χρησιμοποιούν τα λειτουργικά συστήματα εξαρτάται κυρίως από την Εικονική Μνήμη. Η τεχνική της εικονικής μνήμης είναι υπεύθυνη για την διατήρηση της σταθερότητας των λειτουργικών συστημάτων, αφήνοντας πάντα χώρο στην φυσική μνήμη (RAM), έτσι ώστε να εκτελούνται διεργασίες οι οποίες απαιτούν (στο σύνολό τους ή και η καθεμία) περισσότερη μνήμη RAM από αυτή που διαθέτει το υπολογιστικό σύστημα. Για τη σωστή διαχείριση της εικονικής μνήμης χρησιμοποιούνται διάφορες μέθοδοι. Η σελιδοποίηση είναι η πιο κοινά χρησιμοποιούμενη μέθοδος, γιατί συνήθως υποστηρίζεται από το υλικό των υπολογιστών. Δύο άλλες μέθοδοι που χρησιμοποιούνται, αλλά όχι τόσο

ευρέως, είναι η κατάτμηση και η κατατμημένη σελιδοποίηση· η δεύτερη είναι συνδυασμός κατάτμησης και σελιδοποίησης.

**57. Από τι εξαρτάται το μέγεθος της μνήμης που μπορεί να χρησιμοποιήσει ένας υπολογιστής;**

Το κύριο ερώτημα, που σχετίζεται με το μέγεθος της μνήμης ενός υπολογιστή, είναι το πόσο είναι αρκετό. Η απάντηση είναι σχετικά απλή. Αν ένα υπολογιστικό σύστημα χρησιμοποιεί συνεχώς το μεγαλύτερο ποσοστό της μνήμης του, αυτό σημαίνει πως χρειάζεται αναβάθμιση. Άρα η βασική προϋπόθεση για να αποφασίσει κάποιος το μέγεθος της μνήμης του υπολογιστή του είναι κυρίως ο λόγος για τον οποίο θα χρησιμοποιηθεί ο υπολογιστής (internet surfing, gaming, online streaming κτλ), αλλά και τα ιδιαίτερα χαρακτηριστικά του (ταχύτητα επεξεργαστή, τεχνολογία και μέγεθος σκληρού δίσκου κ.α.).

**58. Εξηγήστε γιατί αυξάνοντας την κεντρική μνήμη, κατά κανόνα, επιταχύνουμε τη λειτουργία του υπολογιστικού συστήματος.**

Διότι δίνουμε τη δυνατότητα στον υπολογιστή να φορτώσει περισσότερα ή μεγαλύτερα προγράμματα, ολόκληρα ή το μεγαλύτερο μέρος αυτών στη κεντρική του μνήμη, με την οποία ο επεξεργαστής έχει άμεση σύνδεση και ταχύτατη ανταπόκριση. Στην αντίθετη περίπτωση ο μεγαλύτερος όγκος δεδομένων θα παραμείνει στη δευτερεύουσα ή στην εικονική μνήμη από τις οποίες ο επεξεργαστής αργεί περισσότερο να λάβει δεδομένα. Συνεπώς αυξάνοντας τη κεντρική μνήμη αυτόματα αυξάνουμε και την ταχύτητα του υπολογιστή μας.

**59. Να αναφερθούν περιληπτικά, ποια είναι τα κύρια μέρη από τα οποία αποτελείται ένας μικροεπεξεργαστής.**

Ένας σύγχρονος μικροεπεξεργαστής αποτελείται από τις ακόλουθες μονάδες.

- Μονάδα αποκωδικοποίησης (Decoding Unit)
- Αριθμητική και Λογική Μονάδα (Arithmetic and Logical Unit, ALU): Η μονάδα στην οποία εκτελούνται μία προς μία οι αριθμητικές ή λογικές πράξεις, όπως υπαγορεύονται από τις εντολές που έχουν δοθεί στον υπολογιστή.
- Καταχωρητές (Registers): Μικρά κελιά μνήμης στο εσωτερικό του επεξεργαστή, που χρησιμοποιούνται για την προσωρινή αποθήκευση των δεδομένων, καθώς αυτά υφίστανται επεξεργασία. Οι καταχωρητές διαφέρουν ανάλογα με τον τύπο του επεξεργαστή και τον κατασκευαστή, τόσο ως προς την οργάνωση όσο και ως προς τη χωρητικότητά τους.
- Μονάδα ελέγχου (Control Unit): Ελέγχει τη ροή δεδομένων από και προς την ALU, τους καταχωρητές, τη μνήμη και τις περιφερειακές μονάδες εισόδου/εξόδου.
- Μονάδα προσκόμισης (Fetch Unit): Μεταφέρει τις εντολές από τη μνήμη στον επεξεργαστή.
- Μονάδα προστασίας (Protection Unit): Εξασφαλίζει το αποδεκτό της κάθε διεργασίας που εκτελεί ο επεξεργαστής, ώστε να μη τροποποιούνται δεδομένα που δεν πρέπει ή να μην εκτελούνται μη αποδεκτές εντολές, όπως π.χ. διαίρεση αριθμού με το μηδέν.



**60. Το ψηφίο ισοτιμίας τι είδους πληροφορία δίνει και τι είδη ισοτιμίας έχουμε;**

Το ψηφίο ισοτιμίας (parity bit) είναι ένα ψηφίο της μορφής 0 ή 1 και προστίθεται πάντα στο τέλος μιας αριθμοσειράς για να δηλώσει αν το σύνολο των ψηφίων με τον αριθμό 1 στην αριθμοσειρά έχουν μονό ή ζυγό αριθμό. Ουσιαστικά αποτελεί την απλούστερη μορφή ελέγχου εγκυρότητας των δεδομένων. Υπάρχουν δύο είδη ισοτιμίας: α) η άρτια ισοτιμία, όταν το πλήθος των ψηφίων 1 σε μια αριθμοσειρά έχει ζυγό αριθμό και β) η περιττή ισοτιμία, όταν το πλήθος των ψηφίων 1 σε μια αριθμοσειρά έχει μονό αριθμό.

**61. Ποια είδη διαμόρφωσης ενός σκληρού δίσκου γνωρίζετε και ποιο δίνει μεγαλύτερη ασφάλεια;**

Δύο είδη διαμόρφωσης σκληρού δίσκου υπάρχουν: η γρήγορη και η κανονική. Η διαφορά της γρήγορης με την κανονική διαμόρφωση δίσκου είναι πως κατά την κανονική, μαζί με την διαγραφή των αρχείων, γίνεται ταυτόχρονα και έλεγχος για κατεστραμμένους τομείς στο δίσκο. Στη γρήγορη διαμόρφωση, αφαιρούνται μόνο τα αρχεία, χωρίς να γίνεται έλεγχος. Αυτό έχει σαν αποτέλεσμα η κανονική διαμόρφωση δίσκου σε ένα δίσκο μεγάλης χωρητικότητας να παίρνει ώρες. Αν ο δίσκος είναι σχετικά καινούριος και γενικά δεν έχετε προβλήματα μαζί του, η γρήγορη διαμόρφωση δίσκου είναι η καλύτερη επιλογή.

**62. Τι είναι τα I/O bus, το interrupt; και τί ο Controller;**

Το I/O Bus είναι ουσιαστικά ένα πλήθος από καλώδια, που λειτουργεί ως δίαυλος για την μεταφορά δεδομένων μεταξύ των μονάδων μέσα στον υπολογιστή.

Το interrupt είναι μια τεχνική ελέγχου της δραστηριότητας εισόδου / εξόδου με την οποία μια περιφερειακή μονάδα ή τερματικό που χρειάζεται να στείλει ή να λάβει μια δεδομένων στέλνει ένα σήμα, το οποίο θα ενεργοποιήσει ένα πρόγραμμα interrupt. Αλληλένδετα με το συνολικό interrupt σύστημα, οι επεξεργαστές ενεργοποιούν μια υπηρεσία interrupt. Η λειτουργία της οποίας θα εξαρτηθεί από τα επίπεδα διακοπής που έχουν τεθεί στον επεξεργαστή και των προτεραιοτήτων που εφαρμόζει. Η τεχνική interrupt απαιτεί πιο πολύπλοκο υλικό και λογισμικό, αλλά κάνει πολύ πιο αποτελεσματική χρήση του χρόνου και των δυνατοτήτων του υπολογιστή.

Ο controller είναι ένα ψηφιακό κύκλωμα που διαχειρίζεται τη ροή δεδομένων προς και από την κύρια μνήμη του υπολογιστή. Ένας controller μπορεί να είναι ένα ξεχωριστό τσιπ ή να ενσωματωθεί σε ένα άλλο τσιπ, να τοποθετηθεί δηλαδή στην ίδια μήτρα ή να αποτελεί αναπόσπαστο τμήμα ενός μικροεπεξεργαστή.

**63. Τι είναι το ρολόι του υπολογιστή και σε τι χρησιμεύει;**

Οι επεξεργαστές εκτελούν διαδοχικές στοιχειώδεις λειτουργίες με τη χρήση ενός ηλεκτρικού σήματος τετραγωνικού παλμού. Αυτό το ηλεκτρικό σήμα συγχρονισμού καλείται σήμα ρολογιού, επειδή παράγεται εξωτερικά του επεξεργαστή, από ένα ταλαντωτή που ονομάζεται ρολόι (clock). Το σήμα ρολογιού (ή χρονισμού) διαδίδεται μέσα από το δίαυλο ελέγχου και εναλλάσσεται περιοδικά μεταξύ μηδέν και ένα. Ο χρόνος που χρειάζεται το ρολόι για να μεταπηδήσει από το μηδέν στο ένα και πίσω στο μηδέν, ονομάζεται περίοδος ή κύκλος του ρολογιού.

64. Να αναφέρετε δύο διαφορετικούς τρόπους εισαγωγής στοιχείων από το πλκτρολόγιο (Με έλεγχο του περιεχομένου και χωρίς έλεγχο) σε γλώσσα assembly.

Αν υποθέσουμε ότι θέλουμε να εισάγουμε τι τιμή 5 στον καταχωρητή \$t0 τότε θα μπορούσαμε να το κάνουμε με δύο τρόπους:

```
a) li $t0,5          # χωρίς έλεγχο περιεχομένου
b) cmp $t0,0         #με έλεγχο περιεχομένου
    je set
    set    li $t0,5
```

65. Να αναφέρετε δύο διαφορετικούς τρόπους απεικόνισης στοιχείων στην οθόνη με assembly.

Με την syscall 4 (print string) εκτυπώνουμε οποιαδήποτε συμβολοσειρά στην οθόνη.

```
Π.χ  li $v0, 4
      la $a0, msg
      syscall
```

Με την syscall 1 (print integer) εκτυπώνουμε οποιαδήποτε αριθμοσειρά στην οθόνη.

```
Π.χ. li $v0, 1
      move $a0, $t2
      syscall
```

66. Τί είναι ένας κατάλογος, ποια στοιχεία γνωρίζετε για αυτόν και ποιες οι υποστηριζόμενες λειτουργίες του;

Στους υπολογιστές καλούμε κατάλογο ή φάκελο ένα ειδικό αρχείο το οποίο περιέχει πληροφορίες για άλλα αρχεία και καταλόγους. Οι πληροφορίες αυτές είναι το όνομα, μέγεθος, τοποθεσία στην οποία βρίσκεται σε κάποιο δευτερεύον αποθηκευτικό μέσο, ημερομηνία δημιουργίας, δικαιώματα προσπέλασης, ιδιοκτήτης κτλ. Ένας κατάλογος που βρίσκεται καταχωρημένος σε έναν κατάλογο ονομάζεται υποκατάλογός του. Έτσι σχηματίζεται μια ιεραρχία καταλόγων, μια δένδροειδής δομή.

67. Τι είναι Διαδοχική οργάνωση αρχείων (sequential) και τι Άμεση οργάνωση αρχείων;

Στη διαδοχική (sequential) οργάνωση οι εγγραφές του αρχείου έχουν μία ορισμένη διάταξη, σύμφωνα με τα περιεχόμενα κάποιου πεδίου, που χαρακτηρίζεται ως κλειδί της εγγραφής. Για παράδειγμα, σ' ένα αρχείο μαθητών με διαδοχική οργάνωση, οι εγγραφές έχουν διάταξη αύξουσα σύμφωνα με το περιεχόμενο του κλειδιού κωδικός. Η εγγραφή με κωδικό 85001 προηγείται της 85501 και αυτή της 86001 κλπ. Εάν για κλειδί ληφθεί το επίθετο, τότε η εγγραφή που αρχίζει από Α προηγείται αυτής που αρχίζει από Β κλπ

Η άμεση (direct) οργάνωση επιχειρεί να εκμεταλλευτεί τις δυνατότητες που προσφέρει ο δίσκος ή άλλες παρόμοιες μονάδες, που έχουν άμεση προσπέλαση σε κάθε φυσική εγγραφή της οποίας η διεύθυνση είναι γνωστή. Στα αρχεία αυτά οι εγγραφές αποθηκεύονται τυχαία στο δίσκο γι' αυτό, πολλές φορές, λέγονται και τυχαία (random) αρχεία. Το κλειδί κάθε φυσικής εγγραφής μετατρέπεται, τυχαία, με κάποιο αλγόριθμο σε φυσική διεύθυνση, δηλαδή κύλινδρο, ίχνος και τομέα. Η διαδικασία αυτή είναι γνωστή ως παραγωγή διευθύνσεων.

**68. Τι είναι το αρχείο system.ini των windows;**

Το SYSTEM.INI ήταν ένα αρχείο αρχικοποίησης (INI) που χρησιμοποιούνταν στις πρώτες εκδόσεις των Microsoft Windows (από 1.01 έως και Me) για να φορτώσει τα προγράμματα οδήγησης συσκευών και το προεπιλεγμένο πυρήνα των Windows (πρόγραμμα διαχείρισης ή Windows Explorer). Πολλές από αυτές τις ρυθμίσεις διατηρήθηκαν στα Windows 9x (95, 98 και Me), αν και τα αρχεία INI είχαν αρχίσει να καταργούνται σταδιακά υπέρ της Registry των Windows. Τα Windows NT 4.0, 2000, XP και Server 2003 εξακολουθούν να αναγνωρίζουν ορισμένες καταχωρήσεις του SYSTEM.INI προκειμένου να παρέχουν συμβατότητα προς τα πίσω με παλαιότερες εφαρμογές 16 bit. Τα Windows Vista και τα υπόλοιπα διαθέτουν επίσης το System.ini.

**69. Ποια είναι τα δομικά μέρη του UNIX;**

Βασικό συστατικό στοιχείο του λειτουργικού συστήματος Linux είναι ο πυρήνας ή kernel. Το kernel είναι το ενδιάμεσο στρώμα ανάμεσα στο υλικό και τον χρήστη ή τις εφαρμογές που εκτελεί ο χρήστης. Διαχειρίζεται τις συσκευές, τη μνήμη RAM και την CPU με τέτοιο τρόπο, ώστε να γίνεται η καλύτερη δυνατή χρήση τους. Γύρω από το kernel, υπάρχουν τα κελύφη ή shells, τα οποία ανάλογα με τα προνόμια της εφαρμογής ή του χρήστη, του επιτρέπουν να πλησιάσει στις δυνατότητες που του παρέχει ο πυρήνας. Αυτό φυσικά γίνεται για λόγους ασφαλείας, ώστε μία λανθασμένη επιλογή να μην οδηγήσει το σύστημα σε κατάρρευση. Παράλληλα, είναι υπεύθυνο να ελέγχει και να δίνει προτεραιότητα στις διεργασίες, να μοιράζει τον χρόνο που θα έχει η καθεμία σε κάθε κύκλο στην CPU και ανάλογα με τις δυνατότητες του hardware να ελέγχει την πολυεπεξεργασία, δηλαδή την ταυτόχρονη εκτέλεση πολλαπλών εφαρμογών. Στο επίπεδο της μνήμης, παραχωρεί τον χώρο που χρειάζεται κάθε εφαρμογή και αποφασίζει τι θα στείλει στην εικονική μνήμη ή swap στον σκληρό δίσκο και τότε.

**70. Τι είναι το FAT και τι το NTFS σύστημα διαχείρισης αρχείων στο δίσκο;**

Το FAT είναι το καθιερωμένο σύστημα αρχείων του λειτουργικού συστήματος DOS της Microsoft, το οποίο έχει επεκταθεί και ενσωματωθεί και σε όλες τις νεότερες εκδόσεις λειτουργικών συστημάτων (σειρά Windows) της εταιρείας. Αποτελεί αρκτικόλεξο των λέξεων File Allocation Table (πίνακας κατανομής αρχείων).

Το FAT είναι το βασικό σύστημα αρχείων του MS-DOS, των Windows 3.11 και των Windows 95 στην πρώτη εκδόσή τους. Αρχικά χρησιμοποιούνταν στους εύκαμπτους δίσκους των 5.25" και μετέπειτα των 3.5". Αργότερα φορμάρονταν με αυτό και οι σκληροί δίσκοι. Είναι αρκετά απλό σύστημα αρχείων και διαμορφώθηκε με τα τεχνολογικά δεδομένα τις εποχής που σχεδιάστηκε. Το περισσότερα λειτουργικά συστήματα το υποστηρίζουν ακόμα, αλλά έχει ξεπεραστεί από τα νεότερα συστήματα αρχείων, όπως το NTFS. Κατά την μορφοποίηση του σκληρού δίσκου δημιουργούνται το Master Boot Record (MBR), το πρωτεύον διαμέρισμα (primary partition) και τα επεκταμένα διαμερίσματα (extended partitions).

Σε περιβάλλον MS-DOS υπήρχε η δυνατότητα του DoubleSpace, η συμπίεση δηλαδή του σκληρού δίσκου για εξοικονόμηση δεδομένων. Μέσα σε ένα αρχείο συμπιέζονταν όλα τα αρχεία. Αυτό ήταν αρκετά επικίνδυνο (για απώλεια δεδομένων λόγω αστοχίας) και αργό αλλά χρήσιμο, όταν ο χώρος αποθήκευσης δεδομένων ήταν περιορισμένος και ακριβός. Το

DoubleSpace εγκαταλείφθηκε από το FAT32 αλλά με το NTFS η ιδέα επανήλθε, φυσικά με άλλα χαρακτηριστικά.

Το NTFS είναι το καθιερωμένο σύστημα αρχείων του λειτουργικού συστήματος Windows NT. Έχει ενσωματωθεί και στις νεότερες εκδόσεις αυτού του λειτουργικού συστήματος, δηλαδή στα Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008 και Windows Vista κτλ. Αποτελεί αρκτικόλεξο των λέξεων New Technology File System.

Το σύστημα αρχείων NTFS έχει εκτοπίσει το προηγούμενο FAT ως το προτιμώμενο σύστημα αρχείων για τα λειτουργικά συστήματα Windows της Microsoft. Ενσωματώνει αρκετές βελτιώσεις σε σχέση με το FAT και το HPFS, όπως η καλύτερη υποστήριξη μεταδεδομένων και η χρήση προχωρημένων δομών δεδομένων με σκοπό τη βελτίωση της απόδοσης, της αξιοπιστίας και της εκμετάλλευσης του διαθέσιμου χώρου στο δίσκο, καθώς και επιπρόσθετες προσθήκες που αφορούν την ασφάλεια του συστήματος.

Τα πλεονεκτήματα χρήσης του σε σχέση με το FAT και FAT32 προκύπτουν από τις δυνατότητες του και είναι:

- ασφάλεια πρόσβασης με δικαιώματα σε φακέλους ή μεμονωμένα αρχεία.
- ανάκαμψη μετά από αστοχία γιατί η καταγραφή των transactions ολοκληρώνεται πριν την εκτέλεσή τους
- μικρότερος κατακερματισμός αρχείων.
- δεν μπορούμε να ξεκινήσουμε το σύστημα με δισκέτες DOS (το οποίο είναι και μειονέκτημα).

#### **71. Τι είναι ανάκτηση πληροφοριών και πόσα είδη υπάρχουν;**

Η ανάκτηση πληροφοριών ή ανάκτηση πληροφορίας είναι το επιστημονικό πεδίο της πληροφορικής που μελετά την αποδοτική αναζήτηση πληροφοριών και κάθε τύπου δεδομένων μέσα σε έγγραφα και μεταδεδομένα σχετικά με έγγραφα, όπως επίσης και με την αναζήτηση σε βάσεις δεδομένων και στον Παγκόσμιο Ιστό (WWW). Μία εννοιολογική σύγχυση και μία πολύ εκτενής επικάλυψη υφίστανται στη χρήση των όρων ανάκτηση δεδομένων, ανάκτηση εγγράφων, ανάκτηση πληροφοριών και ανάκτηση κειμένου. Ωστόσο, καθένας από αυτούς τους όρους έχει τη δική του θεωρία και τις δικές του τεχνολογίες. Η ανάκτηση πληροφοριών στηρίζεται στη θεωρία των βάσεων δεδομένων, σε κατάλληλα υπολογιστικά συστήματα και σε μαθηματικές μεθόδους της τεχνητής νοημοσύνης, ενώ εφαρμόζεται στην επιστήμη πληροφορικής και στον Παγκόσμιο Ιστό (μηχανές αναζήτησης).

Υπάρχουν διάφορα μοντέλα ανάκτησης πληροφοριών, τα πιο σημαντικά από αυτά είναι τα πιθανοτικά μοντέλα, τα μοντέλα ανεστραμμένο αρχείου και τα Boolean μοντέλα.

- 72. Τι είναι ένα κατακερματισμένο αρχείο και τι δεικτοδοτούμενο; Αναφέρατε τι γίνεται με τις εγγραφές των αρχείων σε κάθε μία από τις παραπάνω κατηγορίες.**

Τα κατακερματισμένα αρχεία (hash files) είναι μια αριθμητική αναπαράσταση δεδομένων και δεν είναι εύκολο για τον άνθρωπο να ερμηνεύσει. Ένα κατακερματισμένο αρχείο είναι ένα αρχείο που έχει μετατραπεί σε αριθμητική συμβολοσειρά με μαθηματικό αλγόριθμο. Αυτά τα δεδομένα μπορούν να γίνουν κατανοητά μόνο αφού έχουν αποκρυπτογραφηθεί με συγκεκριμένο κλειδί κατακερματισμού.

Η διαδικασία του hashing είναι η μαθηματική μετατροπή μιας αλυσίδας χαρακτήρων σε μια μικρότερη τιμή που συνήθως ονομάζεται κλειδί κατακερματισμού. Αυτή η νέα τιμή αντιπροσωπεύει την αρχική συμβολοσειρά χαρακτήρων μετά την κρυπτογράφηση της. Το Hashing χρησιμοποιείται συχνά σε βάσεις δεδομένων ως μέθοδος δημιουργίας ενός ευρετηρίου. Επειδή οι hashed τιμές είναι μικρότερες από τις αριθμοσειρές του αρχείου, η βάση δεδομένων μπορεί να εκτελεί ταχύτερα τις λειτουργίες ανάγνωσης και γραφής.

Το δεικτοδοτούμενο αρχείο είναι ένα αρχείο δεδομένων με συγκεκριμένους δείκτες που επιτρέπουν την εύκολη και γρήγορη τυχαία πρόσβαση σε οποιαδήποτε εγγραφή του αρχείου δεδομένου πάντα ενός συγκεκριμένου κλειδιού-δείκτη.

Το κλειδί πρέπει να είναι τέτοιο ώστε να αναγνωρίζει μονοσήμαντα μια εγγραφή. Εάν υπάρχουν περισσότεροι από ένα δείκτη, οι υπόλοιποι ονομάζονται εναλλακτικοί δείκτες. Οι εν λόγω δείκτες δημιουργούνται μαζί με το αρχείο και διατηρούνται από το σύστημα.

- 73. Πρέπει να ανακτήσετε το αρχείο mad.ben, σε ένα υπολογιστή με Λ/Σ Linux. Έχει κρατηθεί αντίγραφο ασφαλείας του στο αρχείο Backup.tar. Με τη χρήση ποιός/ποιών εντολής/ών θα το κάνετε;**

Η εντολή που πρέπει να χρησιμοποιηθεί είναι `tar -xf Backup.tar -C / mad.ben`

- 74. Ποιος είναι ο πιο εύκολος τρόπος να εισάγετε μία λίστα εντολών από τη γραμμή εντολών στα Linux;**

Ανοίγουμε ένα παράθυρο Τερματικού (terminal). Αν έχουμε Ubuntu, θα το βρούμε στο μενού Εφαρμογές > Βοηθήματα. Εναλλακτικά, στο γραφικό περιβάλλον Gnome, μπορούμε να τρέξουμε το Τερματικό πατώντας `Alt+F2` και γράφοντας: `gnome-terminal`.

- 75. Τί είναι το partial backup στα Linux;**

Στο partial Backup αντί να αντιγράφει ολόκληρα αρχεία, μπορεί κανείς να περιορίσει το αντίγραφο ασφαλείας μόνο σε μπλοκ ή bytes μέσα σε ένα αρχείο που έχει αλλάξει σε μια δεδομένη χρονική περίοδο. Αυτή η τεχνική μπορεί να χρησιμοποιήσει σημαντικά λιγότερο χώρο αποθήκευσης στο μέσο δημιουργίας αντιγράφων ασφαλείας, αλλά απαιτεί υψηλό επίπεδο πολυπλοκότητας για την ανασύσταση των αρχείων σε μια περίπτωση αποκατάστασης (recovery).

**76. Πόσα μέρη (partitions) χρειάζεται για να εγκαταστήσετε τα Linux; Ποια θα είναι αυτά και τί θα εγκατασταθεί στο κάθε ένα από αυτά;**

Χρειάζονται τουλάχιστον δύο partitions το primary partition (/) και τουλάχιστον όσο είναι η μνήμη RAM για το swap partition.

Το primary partition είναι υπεύθυνο για την εκκίνηση (boot) του λειτουργικού συστήματος και την ομαλή του λειτουργία, δεδομένου ότι εκεί είναι αποθηκευμένα τα βασικά του αρχεία (root directory).

Το swap partition λειτουργεί κυρίως σε περίπτωση υπερχειλίσης στη μνήμη (RAM). Εάν η κεντρική μνήμη γεμίσει εντελώς, τυχόν πρόσθετες εφαρμογές θα εκκινήσουν από το SWAP partition και όχι από τη μνήμη.

Αυτό ακούγεται σαν ένας εύκολος τρόπος για να αυξήσει κάποιος το ποσό της χρησιμοποιήσιμης μνήμης χωρίς να έχει περισσότερη μνήμη RAM, αλλά αυτό δεν συμβαίνει. Η RAM είναι το ιδανικό υλικό για μνήμη επειδή είναι εξαιρετικά γρήγορο, σε αντίθεση με τους σκληρούς δίσκους οι οποίοι, είναι σχετικά αργοί.

**77. Τι σημαίνει παροχέτευση (spooling) και ποια τα πλεονεκτήματα αυτής της τεχνικής στα Λειτουργικά Συστήματα;**

Όταν μια διεργασία εκτελείται διαβάζει από τον δίσκο και γράφει σε αυτόν (και όχι απευθείας στις συσκευές E/E). Ομοίως, όταν κάτι πρέπει να τυπωθεί στον εκτυπωτή τοποθετείται πρώτα στο δίσκο και από εκεί στέλνεται στον εκτυπωτή. Αυτή η τεχνική ονομάζεται spooling (Simultaneous Peripheral Operation On-Line).

Ο δίσκος χρησιμοποιείται ως ένας τεράστιος buffer. Οι λειτουργίες E/E μιας διεργασίας μπορεί να εκτελείται ενώ η cpu χρησιμοποιείται για την εκτέλεση μιας άλλης διεργασίας, το οποίο συνεπάγεται πιο αποδοτική χρήση της cpu και των συσκευών E/E.

Η τεχνική αυτή επιτρέπει επίσης την υλοποίηση αποδοτικών και καθιερωμένων (σήμερα) τεχνικών, όπως τη χρονοδρομολόγηση διεργασιών. Ένα σύνολο (pool) διεργασιών προς εκτέλεση είναι αποθηκευμένες στο δίσκο. Το σύστημα μπορεί να επιλέξει ποια θα εκτελέσει στη συνέχεια.

**78. Αναφέρετε τι γνωρίζετε για την προστασία φακέλων σε σύστημα UNIX. Πόσες κατηγορίες χρηστών υπάρχουν, πόσα ψηφία λαμβάνει κάθε πεδίο και τι σημαίνει το κάθε ψηφίο ανάλογα με την τιμή που λαμβάνει;**

Η φιλοσοφία της διαχείρισης δικαιωμάτων αρχείων και φακέλων σε σύστημα Unix διαχωρίζει τους χρήστες σε τρεις ομάδες:

- **Ιδιοκτήτης** (user ή owner): πρόκειται για το χρήστη που δημιούργησε το αρχείο (εκτός αν έχει γίνει αλλαγή ιδιοκτήτη).
- **Ομάδα** (group): πρόκειται για τους χρήστες που ανήκουν στην ίδια ομάδα με τον ιδιοκτήτη. Ο κάθε χρήστης μπορεί να ανήκει σε περισσότερες από μία ομάδες.
- **Άλλοι** (others): όλοι οι υπόλοιποι χρήστες.



Τα είδη των δικαιωμάτων είναι:

- **Ανάγνωση:** αυτό το δικαίωμα δίνει στο χρήστη που το έχει τη δυνατότητα να εμφανίσει τα περιεχόμενα του αρχείου ή του φακέλου. Επίσης, δίνει το δικαίωμα αντιγραφής σε άλλη τοποθεσία.
- **Εγγραφή:** οι χρήστες που έχουν δικαιώματα εγγραφής μπορούν να τροποποιήσουν τα περιεχόμενα του αρχείου ή του φακέλου (π.χ. να δημιουργήσουν ένα νέο αρχείο μέσα στο φάκελο). Επίσης, το δικαίωμα εγγραφής επιτρέπει τις λειτουργίες της μετακίνησης, της διαγραφής και της μετονομασίας.
- **Εκτέλεση:** αφορά εκτελέσιμα αρχεία (για παράδειγμα αρχεία που αντιστοιχούν σε εντολές).

Φυσικά, κάθε χρήστης μπορεί να αλλάξει τα δικαιώματα μόνο των αρχείων των οποίων είναι ιδιοκτήτης, με την εξαίρεση του χρήστη root ο οποίος είναι ο ανώτερος χρήστης του συστήματος και έχει δικαίωμα να τροποποιήσει τα πάντα.

**79. Μια εκτεταμένη λίστα πληροφοριών αρχείων - καταλόγων θα άρχιζε από drwxrwxrwx με πιθανότητα παύλας σε μερικούς από τους παραπάνω χαρακτήρες. Αναλύστε τι σημαίνουν.**

Η λίστα πιθανότατα θα εμφανιζόταν πιθανότατα κάπως έτσι d-rwxrwxrwx. Ουσιαστικά μας υποδεικνύει ότι σε αυτόν τον κατάλογο και οι τρεις ομάδες χρηστών έχουν και τα τρία είδη δικαιωμάτων. Δηλαδή όλοι μπορούν να κάνουν Ανάγνωση(r) – Εγγραφή(w) – Εκτέλεση(x) σε αυτόν το φάκελο ή αρχείο.

**80. Ποιο ρόλο επιτελεί το Λειτουργικό Σύστημα σε έναν υπολογιστή; Τι θα γινόταν αν δεν υπήρχε αυτό;**

Ως λειτουργικό σύστημα (ΛΣ) χαρακτηρίζεται μία συλλογή βασικών προγραμμάτων, η οποία ελέγχει τη λειτουργία του υπολογιστή συνολικά και χρησιμοποιείται ως υπόβαθρο για την εκτέλεση όλων των υπόλοιπων προγραμμάτων, τη διαχείριση των περιφερειακών συσκευών και την εξασφάλιση της επικοινωνίας μεταξύ χρήστη και υπολογιστή. Στην πράξη πρόκειται για ένα επίπεδο λογισμικού που μεσολαβεί μεταξύ του υλικού και των εκτελούμενων προγραμμάτων σε έναν ηλεκτρονικό υπολογιστή. Αποτελείται από ένα σύνολο μηχανισμών μέσω των οποίων επιτυγχάνεται αυτόματη διαχείριση των πόρων ενός υπολογιστή και ελεγχόμενη κατανομή τους στις εκτελούμενες εφαρμογές, έτσι ώστε οι τελευταίες να είναι σε θέση να προσπελάσουν εύκολα τους πόρους και τις συσκευές του συστήματος χωρίς να χρειάζεται να γνωρίζουν με ακρίβεια τη δομή του υποκείμενου υλικού, αλλά και ώστε πολλαπλές εφαρμογές να μπορούν να εκτελούνται ταυτόχρονα χωρίς να έρχονται σε διένεξη μεταξύ τους ή με τον υπολογιστή.

Αν δεν υπήρχε το λειτουργικό σύστημα τότε θα έπρεπε ο χρήστης, κάθε φορά που θέλει να χρησιμοποιήσει μια περιφερειακή συσκευή ή έναν πόρο ενός δικτύου ή ακόμα και ένα πρόγραμμα, να φορτώνει τα αντίστοιχα προγράμματα μόνος του στη μνήμη του υπολογιστή ή να υλοποιεί προγραμματιστικά τις κατάλληλες διασυνδέσεις. Κάτι τέτοιο συνέβαινε τη δεκαετία του '50 στους πρώτους υπολογιστές, όμως στις μέρες μας με την πληθώρα συσκευών και προγραμμάτων που χρησιμοποιούμε είναι κάτι ακατόρθωτο.

**81. Να αναφέρετε ονομαστικά τα κυριότερα μέρη ενός Λειτουργικού Συστήματος (ΛΣ).Τι γνωρίζετε για τον πυρήνα και το ρόλο του σε ένα ΛΣ;**

Ένα Λειτουργικό Σύστημα αποτελείται από τα εξής στοιχεία:

**A) Τον Πυρήνα (Kernel)**

Εάν το λειτουργικό σύστημα είναι ένας ενδιάμεσος μεταξύ του χρήστη και του υλικού, ο πυρήνας (Kernel) είναι το μέρος εκείνο που βρίσκεται πλησιέστερα στο υλικό. Επιτελεί δηλαδή τον ρόλο του ενδιάμεσου «κρίκου» μεταξύ του υλικού και του υπόλοιπου λειτουργικού συστήματος. Επειδή εξαρτάται από το υλικό, ένα μέρος του έχει γραφεί στη γλώσσα μηχανής που κατανοεί η ΚΜΕ του υπολογιστή. Είναι το πρόγραμμα που φορτώνεται πρώτο, όταν ξεκινάει η λειτουργία του υπολογιστικού συστήματος. Επίσης, με ειδικούς μηχανισμούς (σηματοφορείς, ουρές μηνυμάτων) ρυθμίζει την επικοινωνία και το συγχρονισμό μεταξύ των διεργασιών, τις οποίες και διανέμει στον ή στους κατάλληλους επεξεργαστές. Άλλωστε ο πυρήνας ονομάζεται και «ελεγκτής της κυκλοφορίας». Για παράδειγμα, αν δύο διεργασίες πρέπει να χρησιμοποιήσουν τον ίδιο πόρο -π.χ. να διαβάσουν ένα κοινό αρχείο- θα πρέπει να συντονίσουν την εκτέλεσή τους για να αποτραπεί εμπλοκή του συστήματος. Η διαχείριση και επικοινωνία των διεργασιών είναι απαραίτητη, ώστε όλα τα προγράμματα να εκτελούνται και να τρέχουν «δίκαια». Είναι ανεπιθύμητο το πρόγραμμα ενός χρήστη να εκτελείται με τρόπο που να παραμελούνται τα προγράμματα των άλλων χρηστών, οι οποίοι θα πρέπει να περιμένουν. Αντίθετα, κάποια προγράμματα -όπως π.χ. ο έλεγχος επικοινωνιών δικτύου- απαιτείται να εκτελούνται συχνότερα ή κατά προτεραιότητα από άλλα. Τα ζητήματα αυτά τα αντιμετωπίζει ο πυρήνας του λειτουργικού.

**B) Το Σύστημα Αρχείων (File System)**

**Γ) Τον Διερμηνευτή Εντολών (Command Interpreter) ή Φλοιό (Shell)**

Οι υπηρεσίες που προσφέρουν αυτά τα στοιχεία είναι:

- Έλεγχος και συντονισμός των διαδικασιών εισόδου, εξόδου και επεξεργασίας
- Διαχειρίζονται την ΚΜΕ και τη μνήμη του Υπολογιστή
- Ενεργοποιούν διάφορα προγράμματα (π.χ. τους μεταφραστές, προγράμματα οδήγησης)
- Διαμοιράζουν τους πόρους (υλικό και λογισμικό) αν αυτοί χρησιμοποιούνται από διάφορους υπολογιστές (π.χ. Printer)
- Δημιουργούν ένα φιλικό περιβάλλον επικοινωνίας ανάμεσα στο χρήστη και τον Η/Υ

**82. Αναλύστε τι είναι το FTP, τι το HTTP και ποια η χρήση τους στο Διαδίκτυο.**

Ο **File Transfer Protocol (FTP)**, (*Πρωτόκολλο Μεταφοράς Αρχείων*) είναι ένα ευρέως χρησιμοποιούμενο πρωτόκολλο σε δίκτυα τα οποία υποστηρίζουν το πρωτόκολλο TCP/IP (δίκτυα όπως internet ή intranet). Ο υπολογιστής που τρέχει εφαρμογή FTP client μόλις συνδεθεί με τον server μπορεί να εκτελέσει ένα πλήθος διεργασιών όπως ανέβασμα αρχείων στον server, κατέβασμα αρχείων από τον server, μετονομασία ή διαγραφή αρχείων από τον server κ.ο.κ. Το πρωτόκολλο είναι ένα ανοιχτό πρότυπο. Είναι δυνατό κάθε υπολογιστής που είναι συνδεδεμένος σε ένα δίκτυο, να διαχειρίζεται αρχεία σε ένα άλλο υπολογιστή του δικτύου, ακόμη και εάν ο δεύτερος διαθέτει διαφορετικό λειτουργικό σύστημα.

Το **Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol, HTTP)** είναι ένα πρωτόκολλο επικοινωνίας. Αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές



του Παγκοσμίου Ιστού για να μεταφέρει δεδομένα ανάμεσα σε έναν διακομιστή (server) και έναν πελάτη (client).

- 83. Να δώσετε τους ορισμούς για τα τρία είδη επικοινωνίας: απλή (simplex), αμφίδρομη μη ταυτόχρονη (half duplex) και αμφίδρομη ταυτόχρονη (full duplex). Ποιες είναι οι ομοιότητες και ποιες οι διαφορές τους;**

#### **Μονόδρομη μετάδοση (Simplex)**

Στη μονόδρομη μετάδοση η πληροφορία μεταφέρεται μόνο προς τη μία κατεύθυνση. Η ροή των δεδομένων είναι από τον πομπό προς το δέκτη και δεν μπορεί να αντιστραφεί. Τυπικά παραδείγματα μονόδρομης μετάδοσης είναι η αποστολή δεδομένων στον εκτυπωτή, η ραδιοφωνική εκπομπή, η τηλεοπτική εκπομπή, η αποστολή δεδομένων σε συστήματα καταγραφής (logging system) κλπ.

#### **Μη ταυτόχρονη αμφίδρομη μετάδοση (Half Duplex)**

Στην περίπτωση αυτή, η ροή της πληροφορίας είναι είτε προς τη μία είτε προς την άλλη κατεύθυνση, αλλά όχι ταυτόχρονα. Κάποια χρονική στιγμή  $t_0$  τα δεδομένα κινούνται από τον υπολογιστή Α προς τον υπολογιστή Β. Μια επόμενη χρονική στιγμή Γ, η ροή των δεδομένων είναι από τον υπολογιστή Β προς τον Α. Η μη ταυτόχρονη αμφίδρομη μετάδοση είναι ευρύτατα χρησιμοποιούμενη. Ο χρόνος καθυστέρησης για την αλλαγή της κατεύθυνσης ροής των δεδομένων ονομάζεται turnaround time.

#### **Αμφίδρομη μετάδοση (Full Duplex)**

Σ' αυτή τη μορφή της επικοινωνίας η πληροφορία κινείται ταυτόχρονα και προς τις δύο κατευθύνσεις. Δηλαδή, την ίδια χρονική στιγμή ο σταθμός λαμβάνει δεδομένα και συγχρόνως εκπέμπει δεδομένα. Σε ορισμένες περιπτώσεις υπάρχουν διαφορετικά κυκλώματα λήψης και εκπομπής. Σε άλλες περιπτώσεις δημιουργούνται λογικά κανάλια (για λήψη και εκπομπή) στο ίδιο μέσο μετάδοσης. Η αμφίδρομη μετάδοση είναι ευρύτατα διαδεδομένη στην επικοινωνία υπολογιστικών συστημάτων. Φυσικά, στην αμφίδρομη μετάδοση δεν υπάρχει χρονική καθυστέρηση για αλλαγή της κατεύθυνσης ροής των δεδομένων.

- 84. Να εξηγήσετε πώς λειτουργούν οι υπηρεσίες με σύνδεση (connection oriented services) και πώς οι υπηρεσίες χωρίς σύνδεση (connectionless services). Να δώσετε από ένα αντιπροσωπευτικό παράδειγμα για τον κάθε τύπο υπηρεσίας.**

Οι υπηρεσίες προσανατολισμένες στη σύνδεση, απαιτούν την αποκατάσταση της επικοινωνίας ανάμεσα στους δύο υπολογιστές, πριν την έναρξη της μετάδοσης δεδομένων. Αυτό σημαίνει πως πριν ο αποστολέας αρχίσει να στέλνει δεδομένα στον παραλήπτη, επιλέγεται και δεσμεύεται μια διαδρομή μέσα στο δίκτυο, και στη συνέχεια τα δεδομένα προς μετάδοση, φτάνουν στον παραλήπτη χρησιμοποιώντας αυτή τη διαδρομή. Με άλλα λόγια αυτό το είδος της σύνδεσης λειτουργεί ως ένας σωλήνας τον οποίο ο αποστολέας τροφοδοτεί με αντικείμενα από το ένα άκρο και ο παραλήπτης τα λαμβάνει στο άλλο άκρο. Αυτό φυσικά σημαίνει πως η σειρά παραλαβής των δεδομένων ταυτίζεται με τη σειρά αποστολής τους. Χαρακτηριστικό παράδειγμα είναι το τηλεφωνικό σύστημα.

Από την άλλη πλευρά, στην υπηρεσία χωρίς σύνδεση, δεν λαμβάνει χώρα αποκατάσταση της επικοινωνίας ανάμεσα στους δύο υπολογιστές, αλλά το κάθε πακέτο δρομολογείται από το σύστημα ανεξάρτητα από τα άλλα πακέτα, και φέροντας πάνω του τη διεύθυνση του παραλήπτη. Αυτή η δρομολόγηση του πακέτου, γίνεται χωρίς προηγουμένως να καθοριστεί κάποια διαδρομή μέσα στο δίκτυο, όπως γίνεται στην υπηρεσία με σύνδεση. Αυτό έχει ως αποτέλεσμα ορισμένα πακέτα που φεύγουν μετά από κάποια άλλα να παραλαμβάνονται πρώτα. Στην περίπτωση αυτή ο παραλήπτης θα

πρέπει να έχει τη δυνατότητα πριν τα συναρμολογήσει, να τα τοποθετήσει με τη σωστή σειρά. Χαρακτηριστικό παράδειγμα είναι το ταχυδρομικό σύστημα.

**85. Να δώσετε ορισμό της έννοιας "μεταγωγή πακέτου". Να περιγράψετε τις 2 μεθόδους προώθησης πακέτων.**

Αποτελεί παραλλαγή της μεταγωγής μηνύματος. Η πληροφορία τεμαχίζεται σε περισσότερα του ενός πακέτου μετάδοσης. Εμφανίζεται με δύο παραλλαγές στην τεχνική:

1. Χρήση αυτοδύναμου πακέτου (datagram). Οι ενδιαμέσοι κόμβοι ανταλλάσσουν κατάλληλα τα πακέτα που δρομολογούνται, ώστε αυτά να φτάσουν στον προορισμό τους έστω και από διαφορετικούς δρόμους. Δεν έχει αξιοπιστία (πχ. χρήση UDP πρωτοκόλλου).
2. Χρήση νοητού κυκλώματος (VC). Αποτελεί μια προδιαγραμμένη νοητή διαδρομή που ακολουθούν τα πακέτα μέσα σε ένα δίκτυο για να φτάσουν στον προορισμό τους.

**86. Τι είναι η τεχνική Frame Relay και ποιες ανάγκες καλύπτει;**

Το Frame Relay, είναι σύγχρονη τεχνολογία γρήγορης μεταγωγής πακέτων μεταβλητού μεγέθους. Σε αυτή την τεχνολογία έχουν αφαιρεθεί αρκετές λειτουργίες ελέγχου οι οποίες δεν είναι απαραίτητες σε αξιόπιστο και ασφαλές ψηφιακό περιβάλλον. Επίσης, έχει προδιαγραφεί η διεπαφή μεταξύ τερματικής συσκευής (DTE) και δικτύου (DCE). Τα δίκτυα τεχνολογίας Frame Relay είναι αρκετά δημοφιλή γιατί εκτελούν πολύ πιο γρήγορα από άλλα συστήματα μεταγωγής πακέτου βασικές λειτουργίες προώθησης πακέτων. Αυτό συμβαίνει επειδή με τη χρήση μόνιμων νοητών κυκλωμάτων είναι εκ των προτέρων καθορισμένη η διαδρομή που θα ακολουθήσουν τα πακέτα μιας σύνδεσης από άκρη σε άκρη. Δεν είναι ανάγκη να υπάρχουν συσκευές που να τεμαχίζουν και να επανασυναρμολογούν τα πακέτα ή να αποφασίζουν για την καλύτερη διαδρομή. Επίσης, τα δίκτυα Frame Relay είναι δυνατό να παρέχουν στους χρήστες τους εύρος ζώνης ανάλογα με τις ανάγκες τους. Υποστηρίζουν ταχύτητες από 64 Kbps έως 2,048 Mbps. Η τιμολόγηση της χρήσης του δικτύου Frame Relay εξαρτάται από το επιθυμητό εύρος ζώνης. Για την πρόσβαση τοπικού δικτύου σε δίκτυο Frame Relay απαιτείται μισθωμένη ψηφιακή γραμμή για τη σύνδεση με τον πλησιέστερο κόμβο, δρομολογητής με κάρτα Frame Relay και συσκευή CSU/DSU για τον μετασχηματισμό του ψηφιακού σήματος. Η υπηρεσία Frame Relay είναι οικονομικότερη από τη χρήση αφιερωμένων γραμμών, όταν πρόκειται να διασυνδεθούν αρκετά τοπικά δίκτυα σε πολλές απομακρυσμένες περιοχές. Απαιτείται λιγότερο υλικό στα κεντρικά γραφεία του χρήστη- πελάτη της υπηρεσίας, αφού αρκεί μία μόνο γραμμή E1/T1 (μεταξύ του πελάτη και του παροχέα της υπηρεσίας) για το σύνολο των απαιτούμενων γραμμών. Ακόμη, ο φορέας της υπηρεσίας είναι υπεύθυνος για τη διαχείριση και καλή λειτουργία του δικτύου Frame Relay, αντίθετα με τις αφιερωμένες γραμμές, όπου υπεύθυνος είναι ο χρήστης-πελάτης.

**87. Τι ονομάζουμε επικοινωνία δεδομένων;**

Μετάδοση δεδομένων, είναι η φυσική μεταφορά των δεδομένων σε απόσταση (ψηφιακή ροή bit) από σημείο-προς-σημείο ή από σημείο σε πολλά σημεία (point-to-multipoint) χρησιμοποιώντας ένα μέσο μετάδοσης. Παραδείγματα τέτοιων μέσων μετάδοσης είναι σύρματα χαλκού, οπτικές ίνες, ασύρματα μέσα επικοινωνίας, και μέσα αποθήκευσης. Τα δεδομένα μεταφέρονται χρησιμοποιώντας είτε ηλεκτρομαγνητική ακτινοβολία, ηλεκτρικά σήματα (μεταβολές τάσης), ραδιοκύματα, μικροκύματα ή υπέρυθρη ακτινοβολία. Το είδος δεδομένων που μπορούν να μεταδοθούν μπορεί να ανήκουν σε συνεχόμενο πεδίο τιμών - να είναι δηλαδή αναλογικά σήματα, είτε να έχουν διακριτές τιμές, να είναι ψηφιακά σήματα.

**88. Ποια είναι τα κύρια στοιχεία μετάδοσης των δεδομένων;**

Έχουμε την Σειριακή Μετάδοση. Εδώ τα bit στέλνονται το ένα μετά το άλλο μέσα από το κανάλι (μέσο) μετάδοσης.

Έχουμε την Παράλληλη Μετάδοση. Εδώ τα bit αποστέλλονται ταυτόχρονα (ομαδικά), μέσα από πολλαπλά κανάλια μετάδοσης.

Έχουμε την Ασύγχρονη Μετάδοση. Εδώ τα δεδομένα που αποστέλλονται έχουν την μορφή χαρακτήρων.

Τέλος έχουμε την Σύγχρονη Μετάδοση. Εδώ οι χαρακτήρες ομαδοποιούνται σε blocks.

**89. Ποιους βασικούς τύπους καλωδίων συνεστραμμένων ζευγών γνωρίζετε και ποιες είναι οι διαφορές τους;**

UTP: Αθωράκιστο καλώδιο συνεστραμμένων ζευγών. Αποτελείται από 4 ζεύγη συνεστραμμένων αγωγών που καλύπτονται μόνο από μονωτικό περίβλημα (εσωτερικό και εξωτερικό).

FTP: Θωρακισμένο με φύλλο αλουμινίου. Εδώ τα 4 ζεύγη αγωγών θωρακίζονται με φύλλο αλουμινίου κάτω από το εξωτερικό περίβλημα.

STP: Θωρακισμένο με μεταλλικό πλέγμα. Εδώ κάτω από το εξωτερικό περίβλημα υπάρχει μπλενταζ χαλκού για τη θωράκιση του καλωδίου

SFTP: Θωρακισμένο με μεταλλικό πλέγμα + με φύλλο αλουμινίου.

SSTP: Θωράκιση πλέγματος συνολικά και θωράκιση αλουμινίου σε κάθε ζεύγος.

**90. Ποιες είναι οι βασικές αιτίες θορύβου στην επικοινωνία δεδομένων;**

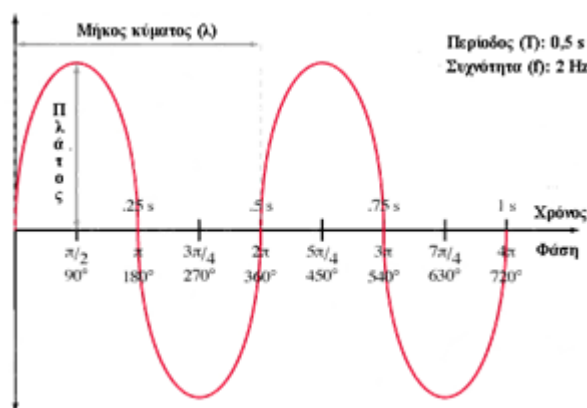
Ηλεκτρομαγνητικές παρεμβολές από γραμμές τροφοδοσίας συσκευών

Παρεμβολές ραδιοσυχνότητας από έντονα ηλεκτρομαγνητικά πεδία κεραιών εκπομπής

Ενδογενής θόρυβος από ατέλειες των συστημάτων μετάδοσης

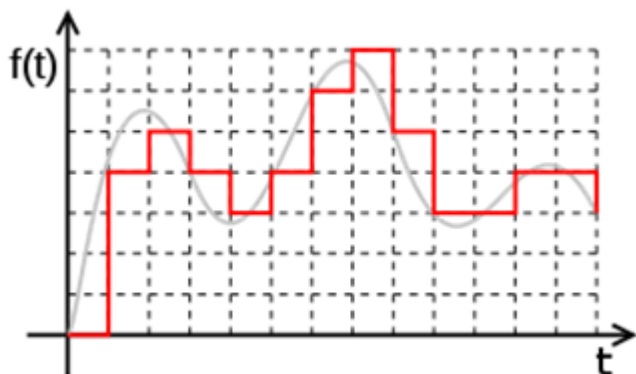
**91. Να αναφέρετε τα Αναλογικά Σήματα με τα χαρακτηριστικά τους.**

Αναλογικά είναι εκείνα τα σήματα που μεταβάλλονται συνεχώς στο χρόνο και μπορούν να πάρουν άπειρες τιμές κατά τη διάρκεια του χρόνου. Χαρακτηριστικό παράδειγμα αναλογικού σήματος είναι η φωνή μας που είναι απλές μεταβολές πίεσης του αέρα. Το αναλογικό σήμα είναι συνάρτηση του χρόνου, η οποία παίρνει τιμές από ένα συνεχόμενο πεδίο τιμών.



## 92. Να αναφέρετε τα Ψηφιακά Σήματα με τα χαρακτηριστικά τους.

Ψηφιακά είναι τα σήματα που λαμβάνουν μόνο διακριτές (συγκεκριμένες) τιμές στη διάρκεια του χρόνου. Το ψηφιακό σήμα ορίζεται σαν συνάρτηση του χρόνου που μπορεί να παίρνει τιμές από συγκεκριμένο σύνολο τιμών. Στο δυαδικό ψηφιακό σήμα μπορεί να παίρνει 2 τιμές



## 93. Σε ποια περίπτωση απαιτείται η μετατροπή του ψηφιακού σήματος σε αναλογικό και το αντίστροφο;

Το σήμα του Η/Υ είναι ψηφιακό. Το σήμα στις γραμμές του ΟΤΕ είναι αναλογικό. Από τον πομπό στο δέκτη χρειάζεται το σήμα να περάσει από τις τηλεφωνικές γραμμές του ΟΤΕ και να καταλήξει πάλι στον Η/Υ του παραλήπτη. Γι' αυτό χρειάζεται η μετατροπή του ψηφιακού σήματος σε αναλογικό και στο τέλος η επαναμετατροπή του αναλογικού σήματος σε ψηφιακό. Αυτή τη δουλειά κάνει το MODEM (Modulator – Demodulator / Διαποδιαμορφωτής)

## 94. Ποιες είναι οι βασικές διαφορές σύγχρονης και ασύγχρονης μετάδοσης δεδομένων;

1. Στην ασύγχρονη μετάδοση οι χαρακτήρες μεταδίδονται ανεξάρτητα ο ένας από τον άλλο, ενώ στην σύγχρονη οι χαρακτήρες ομαδοποιούνται σε Block.
2. Η σύγχρονη μετάδοση συνοδεύεται από ένα σήμα χρονισμού που συνοδεύει την Data και το λέμε Clock.
3. Στην σύγχρονη μετάδοση δεν υπάρχει κενός χρόνος (IDLE) μεταξύ του τελευταίου Bit και του επόμενου πρώτου Bit, όπως υπάρχει στην ασύγχρονη.

## 95. Ποιες είναι οι διαφορές μεταγωγής μηνύματος και μεταγωγής πακέτων;

Στην μεταγωγή μηνύματος η πληροφορία έχει οργανωθεί ολόκληρη σε μήνυμα, ενώ στην μεταγωγή πακέτων, η πληροφορία τεμαχίζεται σε πακέτα και μερικές φορές σε ακόμα μικρότερα μηνύματα.

Στην μεταγωγή μηνύματος, το μήνυμα πρέπει να παραλειφθεί ολόκληρο από κάποιο κόμβο και μετά να βρεθεί ο κατάλληλος κενός δρόμος για την συνέχιση της αποστολής του. Στην μεταγωγή πακέτου τα πακέτα ενώ έχουν τον ίδιο προορισμό δεν ακολουθούν όλα τον ίδιο δρόμο, αλλά χρησιμοποιούν το καθένα, το δικό του συντομότερο δρόμο.

## 96. Πώς ανταλλάσσουν τα δεδομένα ένας πομπός και ένας δέκτης, σε συνάρτηση με το χρόνο; Δώστε ένα παράδειγμα.

Τα λειτουργικά τμήματα που συνιστούν ένα απλό μοντέλο επικοινωνίας είναι οι πηγές της πληροφορίας, το μέσο μετάδοσης, και ο δέκτης της πληροφορίας. Οι πηγές της πληροφορίας μπορούν να διακριθούν σε δύο κατηγορίες, με κριτήριο τη φύση της πληροφορίας που εκπέμπουν, και η οποία μπορεί να είναι αναλογική ή ψηφιακή. Οι πηγές αναλογικής πληροφορίας (όπως είναι για παράδειγμα το μικρόφωνο ή η video κάμερα), δίνουν ένα ή περισσότερα σήματα, που μεταβάλλονται συνεχώς σε συνάρτηση με το χρόνο. Αντίθετα η

πληροφορία που εκπέμπεται από ψηφιακές πηγές (όπως είναι για παράδειγμα το τηλέτυπο) αποτελείται από μια σειρά διακριτών συμβόλων ή γραμμάτων. Το μέσο μετάδοσης, πραγματοποιεί τη φυσική (ηλεκτρική) σύνδεση ανάμεσα στην πηγή και στον προορισμό της πληροφορίας. Η γραμμή επικοινωνίας μπορεί να είναι ένα ζεύγος συρμάτων ή ένα τηλεφωνικό καλώδιο, ή μια οπτική ίνα ή ακόμα ο ελεύθερος χώρος μέσα στον οποίο μεταδίδεται το σήμα που μεταφέρει την πληροφορία. Τέλος, ο δέκτης, αφού πάρει το εξασθενημένο σήμα από το μέσο μετάδοσης, το επαναφέρει στην πραγματική του κατάσταση. Επιτελεί τη λειτουργία αυτή με τη διαδικασία της αποδιαμόρφωσης, η οποία είναι η αντίστροφη της διαδικασίας διαμόρφωσης, που πραγματοποιείται στον πομπό. Λόγω της παρουσίας θορύβου και άλλων παραμορφωτικών σημάτων, ο δέκτης δεν μπορεί να επαναφέρει τέλεια το σήμα του μηνύματος. Εκτός από την αποδιαμόρφωση, ο δέκτης κάνει συνήθως ενίσχυση και φιλτράρισμα του σήματος. Οι χρήστες εναλλάσσονται κυκλικά και καθένας παίρνει όλο το εύρος ζώνης του φυσικού καναλιού για σύντομο και καθορισμένο χρονικό διάστημα.

- 97. Ποια ετικέτα θα χρησιμοποιήσω για να κάνω τη λέξη Επικοινωνία σύνδεσμο για την αποστολή μηνυμάτων μέσω ηλεκτρονικού ταχυδρομείου στην ηλεκτρονική διεύθυνση [webmaster@test.gr](mailto:webmaster@test.gr);**

```
<a href="webmaster@test.gr">Επικοινωνία</a>
```

- 98. Τι θα εμφανίσει ο παρακάτω κώδικας:<HTML> <HEAD> <TITLE> Άσκηση </TITLE> <SCRIPT LANGUAGE="JavaScript"> window.alert ("Αυτό είναι το μήνυμα 1");**

Ο εν λόγω κώδικας θα ανοίξει μια σελίδα ενός φυλλομετρητή (browser), όπου θα τοποθετήσει στη γραμμή τίτλου τη λέξη «Άσκηση» και αυτόματα θα εμφανίσει ένα pop-up παράθυρο που θα γράφει «Αυτό είναι το μήνυμα1».

- 99. Πώς εισάγουμε γραφικά με την HTML; Πρέπει το "αρχείο γραφικού" να βρίσκεται στο ίδιο σύστημα ή μπορεί να είναι εξωτερικό; Τι επιτυγχάνουν οι παράμετροι ALIGN, HEIGHT, WIDTH, CAPTION;**

Για την παρουσίαση ενός γραφικού αρχείου σε μία ιστοσελίδα χρησιμοποιείται η ετικέτα <IMG, SRC= "μονοπάτι/αρχείο εικόνας">. Η ετικέτα αυτή περιέχει πάντα την παράμετρο SRC= "μονοπάτι/αρχείο εικόνας", η οποία και δηλώνει το μονοπάτι όπου είναι αποθηκευμένη η εικόνα, καθώς και το αρχείο ψηφιακής εικόνας που θα χρησιμοποιηθεί.

Οι παράμετροι width και height ορίζουν τις διαστάσεις της εικόνας σε pixels. Συνήθως ταυτίζονται με τις πραγματικές διαστάσεις της εικόνας. Όμως αν χρειάζεται η εικόνα να εμφανιστεί με μικρότερες ή μεγαλύτερες διαστάσεις προσαρμόζονται ανάλογα οι ιδιότητες width και height. Η παράμετρος align Καθορίζει την στοίχιση της εικόνας με το κείμενο που είναι δίπλα σε αυτήν. Οι τιμές που μπορεί να πάρει είναι οι εξής: baseline, top, middle, bottom, absmiddle, absbottom, left, right. Με την παράμετρο caption τοποθετούμε λεζάντα στο γραφικό μας αρχείο.

- 100. Με ποιούς τρόπους μπορούμε να εισάγουμε κείμενο σε μια ιστοσελίδα; (Αναφέρατε τουλάχιστον δύο τρόπους).**

Γράφοντας απευθείας από το πληκτρολόγιο στο επίπεδο του <body> ή κάνοντας αντιγραφή από κάποιο άλλο έγγραφο και επικόλληση στην ιστοσελίδα που μας ενδιαφέρει.



**101. Εξηγήστε τι λειτουργίες εκτελούν οι παρακάτω HTML εντολές/παράμετροι (όταν αναφέρονται σε HTML πίνακες) <CAPTION>, <TH>, COLSPAN, ROWSPAN, ALIGN, VALIGN.**

<CAPTION>: Μας δίνει τη δυνατότητα να τοποθετήσουμε λεζάντα στο πίνακά μας.

<TH>: ορίζουμε επικεφαλίδες στον πίνακα. Το κείμενο που βρίσκετε μέσα στις ετικέτες <th> και </th> εμφανίζεται με bold χαρακτήρες.

COLSPAN : Δημιουργεί συγχώνευση στηλών.

ROWSPAN : Δημιουργεί συγχώνευση γραμμών.

ALIGN: Καθορίζει την στοίχιση των δεδομένων ενός κελιού ή μιας στήλης ενός πίνακα.

VALIGN: Δημιουργεί κάθετη στοίχιση των δεδομένων ενός κελιού ή μιας στήλης ενός πίνακα.

**102. Τί είναι οι σημάνσεις επικεφαλίδας; Ποιό είναι, ως προεπιλογή, το μεγαλύτερο μέγεθος επικεφαλίδας; Το Heading 1 ή το Heading 6;**

Οι HTML επικεφαλίδες είναι κείμενο που εμφανίζετε με μεγάλα γράμματα και χρησιμοποιούνται για την μορφοποίηση κειμένου. Οι επικεφαλίδες ορίζονται από τις ετικέτες <h1> , <h2> , <h3> , <h4> , <h5> και <h6> . Με την <h1> ορίζουμε την μεγαλύτερη ετικέτα ενώ με την <h6> την μικρότερη.

**103. Αναφέρατε δύο τεχνολογίες κατασκευής στατικών ιστοσελίδων (Static web pages) και τέσσερις δυναμικών ιστοσελίδων (Dynamic web pages). Ποιά είναι η διαφορά των στατικών ιστοσελίδων από τις δυναμικές ιστοσελίδες;**

Οι στατικές ιστοσελίδες είναι ουσιαστικά απλά ηλεκτρονικά "έγγραφα". Μπορούν να περιέχουν κείμενα, φωτογραφίες, συνδέσμους (links), αρχεία για download, κινούμενα σχέδια (animated graphics). Είναι κατάλληλες κυρίως για την δημιουργία "μόνιμων/στατικών παρουσιάσεων", όπου δεν υπάρχει συχνά η ανάγκη να τροποποιείται το περιεχόμενό τους. Δηλαδή, θα μπορούσε να χρησιμοποιηθεί μια στατική ιστοσελίδα για την γενική παρουσίαση μιας επιχείρησης και των προϊόντων ή υπηρεσιών της. Οι τεχνολογίες που χρησιμοποιούνται είναι συνήθως η HTML και η XML.

Οι δυναμικές ιστοσελίδες, μπορεί στην εμφάνιση, σε πολλές περιπτώσεις, να μην έχουν μεγάλη διαφορά με τις στατικές, όμως οι δυνατότητές είναι πολύ περισσότερες, από πολλές πλευρές, καθώς στην περίπτωση αυτή ουσιαστικά πρόκειται για μία εφαρμογή (πρόγραμμα), και όχι ένα απλό ηλεκτρονικό έγγραφο.

Συνήθως, οι δυναμικές ιστοσελίδες, χρησιμοποιούν κάποια βάση δεδομένων (database), όπου αποθηκεύουν πληροφορίες και απ' όπου αντλούν το περιεχόμενό τους, ανάλογα με το τι ζητάει ο χρήστης/επισκέπτης σε κάθε του "κλικ". Η χρήση των βάσεων δεδομένων, είναι αυτή που επιτρέπει την εύκολη προσθαφαίρεση περιεχομένου στις δυναμικές ιστοσελίδες, καθώς δεν απαιτείται να επεξεργάζεται κανείς κάθε φορά την ίδια την ιστοσελίδα, αλλά απλά να διαχειρίζεται έμμεσα το περιεχόμενο στην βάση δεδομένων και οι υπόλοιπες διαδικασίες γίνονται αυτοματοποιημένα από τον "μηχανισμό" της ιστοσελίδας.

Γενικά, αυτό που προσφέρουν οι δυναμικές ιστοσελίδες, είναι μεγαλύτερη αλληλεπίδραση του χρήστη με την σελίδα π.χ. να προσθέτει τα σχόλια του στην σελίδα, αλλά και πολλές ευκολίες στον διαχειριστή του περιεχομένου της ιστοσελίδας π.χ. τον ιδιοκτήτη της σελίδας.

Οι τεχνολογίες που χρησιμοποιούνται στις δυναμικές ιστοσελίδες είναι η PHP, η Flash, η .NET, MySQL κ.α.

**104. Με ποιόν τρόπο μπορώ να προσθέσω μια γραμμή θέματος (Subject) σε μια σύνδεση email (mailto);**

Το subject για να προστεθεί σε μια σύνδεση email θα πρέπει να χρησιμοποιηθεί ως παράμετρος της εντολής mailto. Για παράδειγμα αν θέλουμε να ορίσουμε το θέμα (subject) test θα πρέπει να γράψουμε ως εξής: `<a href="mailto:emailname@yourdomain.com?subject=test">The text of your link</a>`

**105. Ποιες ετικέτες θα χρησιμοποιήσουμε σε κάθε μία από τις παρακάτω περιπτώσεις για να εμφανιστεί η λέξη ""Παράδειγμα"": i. μόνο έντονα γραμμένη ii. έντονα και υπογραμμισμένη iii. μόνο υπογραμμισμένη iv. έντονα, πλάγια και υπογραμμισμένη.**

Για το παράδειγμα που δίνεται θα χρησιμοποιήσουμε τις εξής ετικέτες:

- i. `<b>Παράδειγμα</b>`
- ii. `<b><u> Παράδειγμα </b></u>`
- iii. `<u> Παράδειγμα </u>`
- iv. `<b><i><u> Παράδειγμα </b></i></u>`

**106. Τι κάνει η εντολή <BODY BACKGROUND = "bgimage.gif">; Πώς μπορούμε να αλλάξουμε το χρώμα τού φόντου σε ένα HTML κείμενο; Δώστε την εντολή που κάνει το φόντο πράσινο. Δώστε επίσης την εντολή που κάνει το φόντο 50% πράσινο και 50% μπλέ.**

Η εντολή `<BODY BACKGROUND = "bgimage.gif">` θα τοποθετήσει στο φόντο της ιστοσελίδας τη φωτογραφία από το αρχείο bgimage.gif. Για να αλλάξουμε χρώμα φόντου σε ένα HTML κείμενο χρησιμοποιούμε την εντολή `<BODY BGCOLOR=όνομα χρώματος>`. Για να κάνουμε το φόντο 50% πράσινο και 50% μπλε θα χρησιμοποιήσουμε την εντολή `<BODY bgcolor="rgb(0,127,127)">`.

**107. Ποια είναι η λειτουργία της ετικέτας <div>; Τι λειτουργίες κάνουν οι ιδιότητές της align, class και id;**

Η ετικέτα `<DIV>...</DIV>` (Document Division element), λειτουργεί με την παράμετρο ALIGN και αναφέρεται σε ένα τμήμα σελίδας, δηλαδή σε μια σειρά από επικεφαλίδες ή παραγράφους συνολικά (δηλαδή δεν χρειάζεται να επαναλάβουμε την παράμετρο ALIGN σε κάθε μια από τις ετικέτες). Το τμήμα της σελίδας που θα στοιχηθεί με την `<DIV>` μπορεί να περιέχει και ετικέτες εισαγωγής εικόνων.

Το id χρησιμοποιείται για να καθορίσει ένα στυλ για ένα και μοναδικό στοιχείο. Ο επιλογέας id χρησιμοποιεί το χαρακτηριστικό id του στοιχείου στην HTML και ορίζεται με "#". Το id είναι μοναδικό για κάθε στοιχείο και θα είναι συγκεκριμένο σε αυτό το στοιχείο.



Ενώ το class χρησιμοποιείται για να καθορίσει ένα στυλ για μια ομάδα στοιχείων. Μπορείτε να χρησιμοποιήσετε το ίδιο class σε διαφορετικά στοιχεία που σημαίνει ότι μπορείτε να εφαρμόσετε λίγες ιδιότητες του css σε περισσότερα από ένα στοιχεία χρησιμοποιώντας το ίδιο class.

108. Δώστε τις HTML εντολές που υλοποιούν τον παρακάτω πίνακα:

Ιανουάριος	1		
Φεβρουάριος		2	
Μάρτιος			3

```
<table height="200" width="500" border="1">
<tr>
<td width="500" align="center" valign="top">Ιανουάριος</td>
<td width="500" align="center" valign="top">1</td>
<td width="500"></td>
<td width="500"></td>
</tr>
<tr>
<td align="center" valign="top">Φεβρουάριος</td>
<td></td>
<td align="center" valign="top">2</td>
<td></td>
</tr>
<tr>
<td align="center" valign="top">Μάρτιος</td>
<td></td>
<td></td>
<td align="center" valign="top">3</td>
</tr>
</table>
```

109. Τι είναι οι επώνυμες άγκυρες (Named Anchors), και σε ποιούς τύπους ιστοσελίδων χρησιμοποιούνται; Αναφέρατε ένα παράδειγμα.

Ένα στοιχείο ονομάζεται άγκυρα επειδή οι web designers μπορούν να το χρησιμοποιήσουν για να «αγκυροβολήσουν» μια διεύθυνση URL σε κάποιο κείμενο σε μια ιστοσελίδα. Όταν οι χρήστες προβάλλουν την ιστοσελίδα σε ένα πρόγραμμα περιήγησης, μπορούν να κάνουν κλικ στο κείμενο αυτό για να ενεργοποιήσουν το σύνδεσμο και να επισκεφτούν τη σελίδα του οποίου η διεύθυνση URL βρίσκεται στη σύνδεση.

Στην HTML, μια άγκυρα μπορεί να είναι είτε η προέλευση (το κείμενο της άγκυρας) είτε το τέλος (προορισμού) μιας υπερσύνδεσης. Με τις ιδιότητες HTML name ή id, το στοιχείο στο οποίο έχουμε θέσει μια άγκυρα γίνεται στόχος. Ένα URL μπορεί να συνδεθεί με αυτόν τον στόχο μέσω ενός αναγνωριστικού θραύσματος.

**110. Με ποιόν τρόπο μπορούμε να τοποθετήσουμε κείμενο πάνω σε μια εικόνα;**

Υπάρχουν ορισμένοι browsers που δεν υποστηρίζουν την εμφάνιση γραφικών με αποτέλεσμα να μην εμφανίζονται οι εικόνες που τοποθετούμε στις σελίδες μας. Η χρήση της ιδιότητας alt έχει σαν αποτέλεσμα σε έναν τέτοιο browser να εμφανίζετε αντί της εικόνας, το κείμενο το οποίο ορίζεται με την ιδιότητα. Συνήθως το κείμενο αυτό περιγράφει την εικόνα έτσι ώστε ο χρήστης που δεν μπορεί να την δει, να πάρει μια ιδέα για το τι απεικονίζετε σε αυτήν. Το alt είναι τα τρία πρώτα γράμματα από την λέξη alternative.

Αν θέλουμε να βάλουμε κείμενο πάνω στην εικόνα και αυτό να εμφανίζεται πάντα τότε θα πρέπει να χρησιμοποιήσουμε κάποιο πρόγραμμα επεξεργασίας εικόνων. Το πιο εύκολο και γρήγορο είναι το paint των windows. Ανοίγουμε την εικόνα που θέλουμε μέσα από το εν λόγω πρόγραμμα και το μόνο που μας μένει να κάνουμε είναι να πατήσουμε από τα εργαλεία που βρίσκονται αριστερά, το κουμπί με το γράμμα A ή αλλιώς την επιλογή κείμενο. Γράφουμε ότι θέλουμε μέσα στο πλαίσιο που θα επιλέξουμε μόλις πατήσουμε το A και στην συνέχεια πατάμε από το μενού "Αρχείο" και "Αποθήκευση".

**111. Τι είναι το CSS (Cascading Style Sheet) και με ποιούς τρόπους μπορεί να ενσωματωθεί σε μια ιστοσελίδα; (αναφέρατε τουλάχιστον δύο τρόπους)**

Η CSS (Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ) ή (αλληλουχία φύλλων στυλ) είναι μια γλώσσα υπολογιστή που χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη. Δύο τρόποι σύνδεσής τους με την HTML είναι οι εξής:

**Μέθοδος 1: Εξωτερική σύνδεση - External (link to a style sheet)**

Ο πρώτος τρόπος, που είναι αυτός που συνιστάται να χρησιμοποιούμε, είναι ένας σύνδεσμος (link) προς ένα εξωτερικό αρχείο φύλλου στυλ (external stylesheet). Ένα εξωτερικό αρχείο φύλλου στυλ (external style sheet) είναι απλά ένα αρχείο κειμένου (text file) που έχει επέκταση (extension) .css. Μπορούμε να το τοποθετήσουμε (upload) στον Web server, όπως όλα τα άλλα αρχεία.

Για παράδειγμα, ας υποθέσουμε ότι το αρχείο μας style sheet έχει όνομα style.css και βρίσκεται στον φάκελο style. Για να δημιουργήσουμε έναν σύνδεσμο (link) από το HTML έγγραφο, που είναι το default.htm, προς το αρχείο του style sheet, που είναι το style.css, θα πρέπει να γράψουμε τον εξής κώδικα :

```
<link rel="stylesheet" type="text/css" href="style/style.css" />
```

**Μέθοδος 2 : In-line (attribute style)**

Ο άλλος τρόπος με τον οποίο μπορούμε να εφαρμόσουμε ένα CSS σ' ένα HTML κείμενο είναι να χρησιμοποιήσουμε την ιδιότητα (attribute) style της HTML. Το είδαμε ήδη προηγουμένως με το κόκκινο χρώμα φόντου και μπορεί να εφαρμοσθεί ως εξής :

```
<html>
<head>
<title>Παράδειγμα</title>
</head>
<body style="background-color: #FF0000;">
<p>Αυτή είναι μια κόκκινη σελίδα</p>
</body>
</html>
```

**112. Δώστε τον ορισμό του Hyperlink (Υπερσύνδεση) και εξηγήστε ποιό θα είναι το αποτέλεσμα του ορισμού των κάτωθι target: \_self, \_parent, \_blank και \_top όταν αυτά θα ενσωματωθούν σε ένα Hyperlink.**

Σε ηλεκτρονικό περιβάλλον ως σύνδεσμος (link) ή και υπερσύνδεσμος (hyperlink) ονομάζεται η σύνδεση (αναφορά ή παραπομπή) μεταξύ των κόμβων του υπερκειμένου. Συνήθως είναι μια λέξη ή φράση, μαρκαρισμένη με χρωματισμό μπλε και με υπογράμμιση, ή ακόμη και μια εικόνα, που εντοπίζεται στο ηλεκτρονικό κείμενο μιας ιστοσελίδας στο διαδίκτυο και περιέχει την εντολή να «μεταφέρει» τον χρήστη σε άλλη σχετική ιστοσελίδα, από τον κόμβο προέλευσης στον κόμβο προορισμού.

Όταν ορίζουμε στην HTML την εντολή για έναν υπερσύνδεσμο έχουμε τη δυνατότητα να της αποδώσουμε τις εξής ιδιότητες:

Self: Ανοίγει την σελίδα στο ίδιο πλαίσιο το οποίο έγινε κλικ.

Blank: Ανοίγει την σελίδα σε νέο παράθυρο. Αφού πατήσετε τον σύνδεσμο, ανοίγει ένα νέο παράθυρο του browser στο οποίο θα εμφανιστεί την σελίδα.

Parent: Ανοίγει την σελίδα στο "πατρικό πλαίσιο".

Top: ανοίγει την σελίδα στο ίδιο παράθυρο αγνοώντας τα πλαίσια που τυχόν υπάρχουν.

**113. Αναφέρατε τον ορισμό του Χάρτη Εικόνας (Image Map). Ποιο εργαλείο χάρτη εικόνας σάς επιτρέπει να σχεδιάσετε ακανόνιστα σχήματα;**

Στην HTML και XHTML, ένας image map είναι ένας κατάλογος συντεταγμένων που σχετίζονται με μια συγκεκριμένη εικόνα και δημιουργήθηκε για να συνδέσει περιοχές της εικόνας σε διαφορετικούς προορισμούς ή κείμενα (σε αντίθεση με έναν κανονικό σύνδεσμο εικόνας, στον οποίο συνδέεται ολόκληρη η περιοχή της εικόνας ένα μόνο προορισμό). Κατάλληλα εργαλεία για τη δημιουργία Χαρτών Εικόνας είναι τα: Adobe Dreamweaver, το KImageMapEditor (ή KDE), και το imagemap plugin που έχει η εφαρμογή GIMP.

**114. Ποια είναι η χρήση της παραμέτρου Cell Padding για έναν πίνακα μιας ιστοσελίδας και ποια της παραμέτρου Cell Spacing.**

Το CELLSPACING είναι το πλάτος των εικονοστοιχείων (pixel) μεταξύ των επιμέρους κελιών δεδομένων στον πίνακα. (Το πάχος των γραμμών που παράγει το πλέγμα του πίνακα). Η προεπιλογή είναι μηδέν. Εάν το BORDER είναι ρυθμισμένο στο 0, οι γραμμές CELLSPACING θα είναι αόρατες.

Το CELLPADDING αναφέρεται στην απόσταση εικονοστοιχείων μεταξύ των περιεχομένων των κελιών και του πλαισίου του πίνακα. Η προεπιλογή για αυτήν την ιδιότητα είναι επίσης μηδέν. Αυτή η λειτουργία δεν χρησιμοποιείται συχνά, αλλά μερικές φορές είναι χρήσιμη όταν έχουν ενεργοποιηθεί και είναι εμφανή τα πλαίσια και θέλετε το περιεχόμενο να είναι "μακριά" από το περίγραμμα για μια πιο ευδιάκριτη προβολή. Το CELLPADDING είναι αόρατο, ακόμα και με την ιδιότητα BORDER ενεργοποιημένη.

**115. Οι περισσότερες ιστοσελίδες χρησιμοποιούν εικόνες και γραφικά για την αναπαράσταση πληροφοριών. Περιγράψτε τα τρία βασικότερα είδη εικόνων (GIF, JPEG, PNG) και εξηγήστε για το καθένα απ' αυτά τα πλεονεκτήματα χρήσης του σε μια ιστοσελίδα.**

Κάθε είδος εικόνας έχει τα πλεονεκτήματά του και επιλέγεται κάθε φορά ανάλογα με τη χρήση που θέλουμε να κάνουμε γενικότερα αλλά και ειδικότερα στις ιστοσελίδες μας.

Η τεχνολογία JPEG είναι ο κατάλληλος τύπος αρχείου για αποθήκευση φωτογραφιών ή εικόνων που αποτελούνται από πολλά χρώματα. Χρησιμοποιεί πολύπλοκο αλγόριθμο συμπίεσης που επιτρέπει διαβαθμισμένα να μικραίνει το μέγεθος του αρχείου με την ανάλογη (μικρή) απώλεια στην ποιότητα της εικόνας.

Δεν είναι κατάλληλο για γραφικά, banner, εικόνες με κείμενο ή ξεκάθαρα σχήματα αφού η μικρή κατά τα άλλα απώλεια θα φαίνεται έντονα στα όρια των γραμμών και της εναλλαγής σχημάτων. Προγράμματα όπως το Adobe Photoshop μπορούν εύκολα να παράγουν το αρχείο που θέλουμε.

Τα αρχεία GIF χρησιμοποιούνται όταν η εικόνα αποτελείται από ευδιάκριτα σχήματα και έχει λίγα χρώματα. Έτσι και αλλιώς τα αρχεία GIF υποστηρίζουν μέχρι 256 χρώματα. Από αυτό το στοιχείο και μόνο καταλαβαίνουμε πως δεν είναι κατάλληλα αρχεία για φωτογραφίες. Επίσης ένας άλλος λόγος είναι ότι οι φωτογραφίες έχουν πολλά (άπειρα) πιθανά χρώματα και αποχρώσεις ενώ δεν έχουν ευδιάκριτα σχήματα. Συμπερασματικά λοιπόν προκύπτει πως μια φωτογραφία δεν πρέπει να αποθηκευτεί ως αρχείο GIF. Ένα άλλο γραφικό όμως π.χ. ένα διαφημιστικό banner το οποίο είναι απλό και έχει ένα απλό φόντο και ένα κείμενο θα έχει μικρότερο μέγεθος ως αρχείο GIF. Ο αλγόριθμος συμπίεσεως είναι πολύ απλός, όταν όμως συναντήσει μια μικρή παλέτα από χρώματα γίνεται πολύ αποδοτικός.

Το βασικό χαρακτηριστικό – πλεονέκτημα των αρχείων GIF δεν είναι το μέγεθος. Μπορεί να συνδυάσει – συμπεριλάβει περισσότερες των 1 εικόνες και να δημιουργήσει αυτό που λέγεται animated GIF , δηλαδή μια κινούμενη εικόνα. Όλοι έχουν δει στο διαδίκτυο εικόνες που κινούνται ή εναλλάσσονται πιο σωστά. Ένα τρόπος για να δημιουργηθούν είναι τα προγράμματα που παράγουν animated GIF αρχεία.

Οι εικόνες GIF υποστηρίζουν και την διαφάνεια ( transparency ) . Δηλαδή ένα μέρος της εικόνας να μην έχει περιεχόμενο και στην θέση του να φαίνεται ότι υπάρχει από πίσω.

Οι εικόνες PNG αναπτύχθηκαν με σκοπό παρόμοιο με αυτόν των αρχείων GIF. Έχουν μικρό μέγεθος σε ξεκάθαρα σχήματα, δεν είναι κατάλληλη μορφή για αποθήκευση φωτογραφιών αλλά υπερτερούν όσον αφορά την δυνατότητα της διαφάνειας. Η δυνατότητα των αρχείων GIF για τέλεια αποτύπωση διαφανών εικόνων είναι περιορισμένη ενώ αντίστοιχα το αποτέλεσμα με την χρήση των αρχείων PNG μπορεί να είναι τέλειο. Άρα το δυνατό σημείο των αρχείων PNG είναι η ικανότητα στον τομέα της διαφάνειας ( transparency ).

**116. Διατυπώστε τη συνάρτηση swap\_image() σε Javascript που δέχεται ως ορίσματα α) το id του αντικειμένου που θέλουμε να διαχειριστούμε και β) το path της εικόνας "pic2.jpg" που θα χρησιμοποιήσουμε ως rollover image. Προσδιορίστε τι είναι το HTML αντικείμενο που καλεί τη συνάρτηση.**

Αρχικά πρέπει να ορίσουμε το script στο head της HTML.

```
<head>
```

```
<script language="javascript">
```

```
function swap_image() {
```

```
document.getElementById("myID").src = "path/pic2.jpg";
```

```
</script>
```

```
</head>
```

Έπειτα στο body δηλώνουμε το HTML αντικείμενο που θα κάνει rollover

```
<body>
```

```

```

```
</body>
```

**117. Διατυπώστε τη συνάρτηση check\_form() σε Javascript, που ελέγχει εάν τα πεδία username και password της HTML φόρμας "form1" δεν είναι κενά. Εάν κάποιο από αυτά είναι κενό, να ανοίγει ένα message box με την αντίστοιχη ειδοποίηση στον client.**

```
function check_form() {
```

```
if(document.form1.username.value== "" ||
```

```
document.form1.password.value==) {
```

```
    window.alert("Username or password is empty");
```

```
    }
```

```
}
```

**118. Διατυπώστε τον κώδικα για να διαβάσετε το cookie "user" και εάν αυτό υπάρχει, τυπώστε στο browser "Hello..." και ό,τι περιέχει το cookie, αλλιώς αποθηκεύσατε το cookie "user" με τιμή "Unknown visitor" και διάρκεια ζωής 1 ώρα.**

```
<?php
if(isset($_COOKIE["user"])){
echo "Hello..." .$_COOKIE["user"];
}else{
    setcookie("user", "Unknown visitor",time()+3600,"/");
}
?>
```

**119. Διατυπώστε τον κώδικα για ένα script που θα τυπώσει ένα HTML table με 7 σειρές και 6 στήλες. Σε κάθε κελί θα υπάρχει και ένας αριθμός σε αύξουσα σειρά από το 1 ως το 42. Ο πίνακας θα είναι κεντραρισμένος στην σελίδα, με border 1 και padding 4. Η στοίχιση μέσα στα κελιά θα είναι κεντρική και κάθε έβδομος αριθμός (7,14,21...) θα είναι έντονος (bold) και θα έχει χρώμα κόκκινο.**

```
<script>
function makeTableHTML() {
    var result = "<table align="center" border= "1" cellpadding="4" cellspacing="4">";
    var sum=1;
    for(var i=1; i<=7; i++) {
        result += "<tr>";
        for(var j=1; j<=6 j++){
            if (sum%7==0){
                result += "<td bgcolor="#FF0000" align="center"><b>"+ sum + "</b></td>";
                sum++;}
            else{
                result += "<td align= "center">"+ sum + "</td>";
                sum++;}
        }
        result += "</tr>";
    }
    result += "</table>";
    return result;
}
</script>
```

**120. Στείλτε ένα email σε όλες τις εγγραφές του πίνακα subscribers χρησιμοποιώντας τη συνάρτηση mail (to, subject, body). Το email θα έχει θέμα "Login info" και περιεχόμενο "Your username is <το username χρήστη> and your password is <το password χρήστη> ". Δεν απαιτείται ο κώδικας σύνδεσης με τον (My)Sql server. Ο πίνακας subscribers έχει τα πεδία id, username, password, email.**

```
$emailsql = "SELECT * FROM subscribers";
$email_query = mysqli_query($con,$emailsql);
while($emailrow = mysqli_fetch_array ($email_query)){
    $name = $emailrow["username"];
    $pass = $emailrow["password"];
    $to = $emailrow["email"];
    $subject = "Login Info";
    $message = "Your username is ".$name. "and your password
is ".$pass.;
    mail($to, $subject, $message);
}
```

**121. Ποιές είναι οι σημάνσεις HTML (<σήμανση>), όπως εμφανίζονται στον επιλογέα σήμανσης του Dreamweaver, α) για ένα πίνακα, β) για μια γραμμή πίνακα, γ) για ένα κελί πίνακα, και δ) για μια κεφαλίδα πίνακα; Αναφέρατε μόνο τον κώδικό σήμανσης για την κάθε μια από τις τέσσερις περιπτώσεις.**

```
<table>
<tr>
<td>
<th>
```

**122. Γιατί είναι προτιμότερο να ορίζω το στυλ των ιστοσελίδων μου (CSS) σε ένα εξωτερικό αρχείο παρά να εισάγω το στυλ εσωτερικά σε κάθε μια ιστοσελίδα; Ποιό είναι το πρόβλημα με την χρήση εσωτερικών (inline) στυλ;**

Είναι προτιμότερο να ορίζω το στυλ των ιστοσελίδων μου (CSS) σε ένα εξωτερικό αρχείο γιατί έχουμε:

- Πλήρη έλεγχο της δομής της σελίδας.
- Ευκολότερη συντήρηση της ιστοσελίδας μας.
- Λιγότερο χρόνο loading της σελίδας λόγω του ότι το css γίνεται cached στον browser.
- Μεγαλύτερο page ranking(SEO Friendly).

Τα προβλήματα που προκύπτουν με την χρήση εσωτερικού στυλ είναι ότι:

- Μπορεί να κάνει override που εμείς δεν θέλουμε λόγω αλληλουχίας
- Πρέπει να γράφουμε το στυλ για κάθε element που θέλουμε να προσθέσουμε.
- Δεν μπορούμε να βάλουμε στυλ σε ψευδο-elements.
- Μεγαλώνει πολύ το html αρχείο μας.



**123. Ποιοί είναι οι τύποι επιλογέων (Selector Types) που χρησιμοποιούνται για τον ορισμό στυλ ενός CSS (Cascading Style Sheet); Περιγράψτε ποιος είναι ο ρόλος του κάθε τύπου. Δώστε ένα παράδειγμα για κάθε τύπο.**

- **HTML Selector:** Για να καθορίσουμε τον τρόπο με τον οποίο μια συγκεκριμένη ετικέτα HTML θα πρέπει να είναι (στυλ) σε ολόκληρη την ιστοσελίδα μας, χρησιμοποιήστε τον επιλογέα HTML. Για παράδειγμα, εάν θέλουμε όλες οι ετικέτες επιπέδου κεφαλίδας 2 (h2) πρέπει να είναι κόκκινες: `h2{color:red}`.
- **Class Selector:** Αν δεν θέλουμε όλες οι ετικέτες μας να εμφανίζονται ακριβώς ίδιες, χρειαζόμαστε έναν επιλογέα που μπορεί να εφαρμοστεί σε οποιαδήποτε ετικέτα HTML. Αυτός είναι ο επιλογέας class. Κατά τον ορισμό μιας class, τοποθετούμε μια τελεία αμέσως πριν από το όνομα της class για να γνωρίζει το πρόγραμμα περιήγησης ότι πρόκειται για class selector. Αν θέλαμε να κάνουμε το background-color κίτρινο χρησιμοποιώντας class θα γράφαμε:

**CSS**

```
.example{background-color:yellow;}
```

**HTML**

```
<h2 class="example">text</h2>
```

- **ID Selector:** Το πρώτο πράγμα που πρέπει να γνωρίζουμε για τον ID Selector είναι ότι, χρησιμοποιείται αυτός ο επιλογέας μόνο μια φορά σε μια ετικέτα HTML και λειτουργεί ακριβώς όπως ο Class Selector. Η μόνη προφανής διαφορά είναι ότι, στην αρχή, χρησιμοποιείτε ένα # για να το δηλώσουμε, αντί μιας τελείας. Αν θέλαμε ένα text να έχει χρώμα πράσινο με την χρήση του ID Selector θα γράφαμε:

**CSS**

```
#example{color:green;}
```

**HTML**

```
<div id="example">text</div>
```

**124. Περιγράψτε πώς, με τη βοήθεια σχετικού εργαλείου, μπορείτε να εισάγετε σε μια ιστοσελίδα έναν πίνακα δύο στηλών (η πρώτη πλάτους 100 pixels και η δεύτερη πλάτους 300 pixels) και τριών γραμμών, στοιχισμένο στο κέντρο.**

**1<sup>ος</sup> τρόπος: Χρησιμοποιούμε editor**

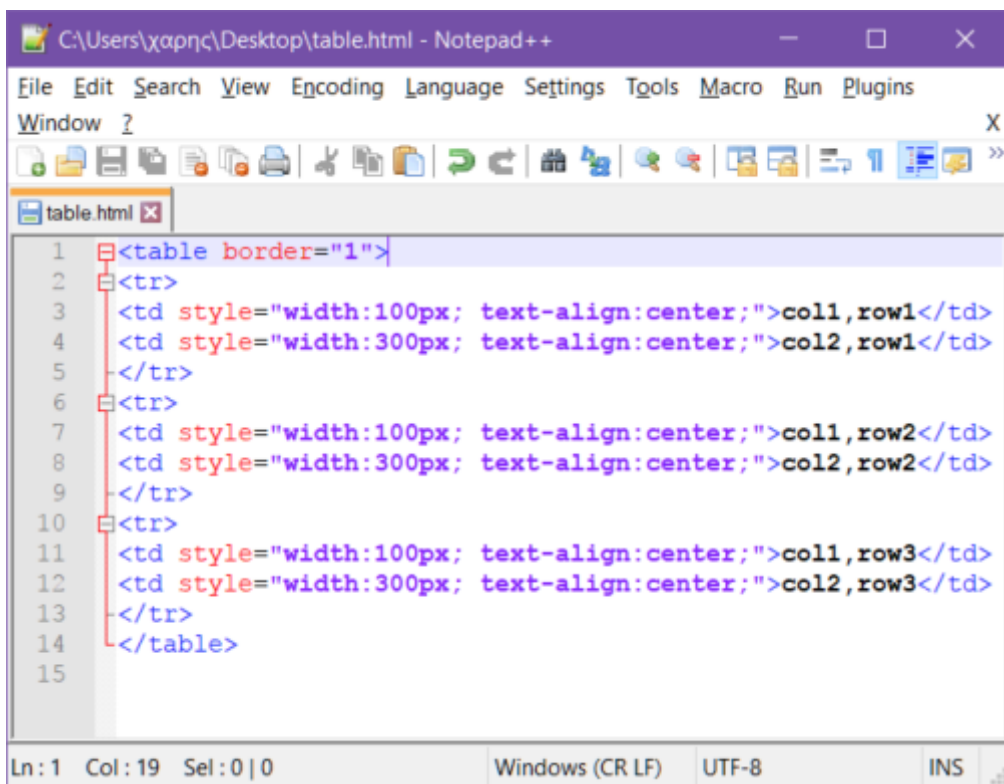
Μπορούμε να χρησιμοποιήσουμε το εργαλείο Notepad++, και ακολουθούμε τα βήματα:

- File->New
- Γράφουμε τον παρακάτω κώδικα:

```
<table border="1">
<tr>
<td style="width:100px; text-align:center;">col1,row1</td>
<td style="width:300px; text-align:center;">col2,row1</td>
</tr>
<tr>
<td style="width:100px; text-align:center;">col1,row2</td>
<td style="width:300px; text-align:center;">col2,row2</td>
</tr>
```

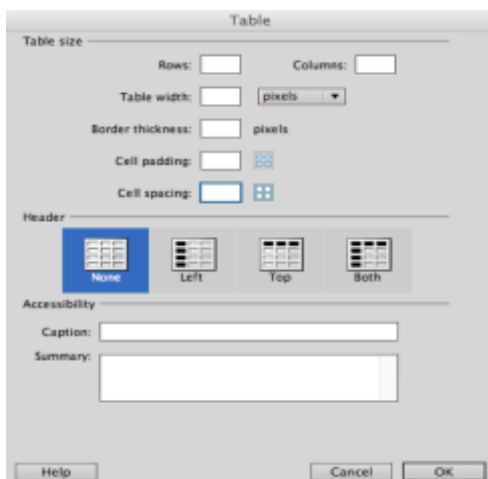
```
<tr>
<td style="width:100px; text-align:center;">col1,row3</td>
<td style="width:300px; text-align:center;">col2,row3</td>
</tr>
```

- Αποθηκεύουμε το αρχείο με κατάληξη .html (πχ. test.html)
- Για να δούμε το αποτέλεσμα ανοίγουμε το αρχείο html με ένα browser(chrome,mozilla,IE)



## 2<sup>ος</sup> τρόπος:Χρησιμοποιούμε dreamweaver

Ανοίγουμε το dreamweaver,από το menu πατάμε Εισαγωγή->πίνακας και στο παράθυρο που ανοίγει βάζουμε τις τιμές που επιθυμούμε.



**125. Περιγράψτε πώς με τη χρήση σχετικού εργαλείου, μπορείτε να δημιουργήσετε IMAGE HOTSPOTS, να εισάγετε ένα JAVA APPLET και πώς μπορείτε να εισάγετε μια αριθμημένη λίστα σε μια ιστοσελίδα. Ποιους άλλους τύπους λίστας μπορείτε να εισάγετε και με ποιο τρόπο;**

Για να δημιουργήσουμε ένα **image hotspot** μέσα στο tag <img> βάζουμε το attribute usemap="#name"

Μετά ορίζουμε το map μας:

```
<map name="name">  
<area shape="circle" coords="90,58,3 href="example.htm" alt="Example"  
</map>
```

Για να εισάγουμε ένα **java applet** στην σελίδα μας χρησιμοποιούμε το tag <applet>Για να εισάγουμε ένα applet το οποίο θα ζωγραφίζει φούσκες ο κώδικας μας θα ήταν έτσι:<applet code="Bubbles.class" width="400px" height="200">Example applet that draws bubbles</applet>

Για την εισαγωγή **αριθμημένης λίστας** χρησιμοποιούμε το attribute type στο tag <ol>:

```
<ol type="1">  
<li>Example1</li>  
<li>Example2</li>  
</ol>
```

Ο κώδικας αυτός θα δημιουργήσει: 1.Example1  
2.Example2

**Άλλοι τύποι λιστών** είναι: Αρίθμηση με κεφαλαία γράμματα(type="A"),αρίθμηση με πεζά γράμματα(type="a"),αρίθμηση με κεφαλαίους λατινικούς αριθμούς(type="I") και αρίθμηση με πεζούς λατινικούς αριθμούς(type="i"). Επίσης άλλος τρόπος εισαγωγής αποτελεί η λίστα περιγραφής (Description List). Η ετικέτα <dl> ορίζει τη λίστα περιγραφής, η ετικέτα <dt> ορίζει τον όρο (όνομα) και η ετικέτα <dd> περιγράφει κάθε όρο:

```
<dl>  
<dt>Example</dt>  
<dd>this is an example</dd>  
</dl>
```

**126. Τι είναι τα «ερωτήματα» (queries) στο σχεσιακό μοντέλο Βάσεων δεδομένων;**

Τα ερωτήματα είναι εντολές που υποβάλλονται προς τη βάση δεδομένων. Χρησιμοποιούνται για την ενημέρωση και διαχείριση των σχεσιακών βάσεων δεδομένων. Για τη σύνταξη των ερωτημάτων χρησιμοποιείται η γλώσσα SQL.

**127. Τι είναι ένα σύστημα διαχείρισης βάσης δεδομένων;**

Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS) είναι ένα σύνολο προγραμμάτων που επιτρέπουν το χειρισμό των πληροφοριών μιας ή περισσότερων Βάσεων Δεδομένων που δημιουργήθηκαν από το ίδιο σύστημα. Τα συστήματα αυτά, δίνουν γενικής χρήσης εργαλεία που επιτρέπουν τη δημιουργία και τον χειρισμό πληροφοριών.

**128. Τι χρησιμοποιούμε προκειμένου να απεικονίσουμε τον πραγματικό κόσμο σ' ένα μοντέλο, το οποίο εύκολα μεταφράζεται σε Σύστημα Διαχείρισης Βάσης Δεδομένων (ΣΔΒΔ);**

Στην περίπτωση των σχεσιακών βάσεων δεδομένων, οι πληροφορίες συνδέονται μεταξύ τους με ΣΧΕΣΕΙΣ που απορρέουν από τα κοινά πεδία διαφορετικών αρχείων. Σε αυτό το είδος Βάσεων Δεδομένων, τα αρχεία λέγονται “tables” (πίνακες), τα records “rows” (γραμμές) και τα πεδία “columns” (στήλες).

**129. Εξηγήστε τι σημαίνει ότι «μία σχέση οντοτήτων είναι ένα προς πολλά». Δώστε ένα παράδειγμα.**

Μία συσχέτιση είναι “Ένα – προς - Πολλά” όταν κάθε μέλος της πρώτης συνδέεται με οποιοδήποτε αριθμό μελών της δεύτερης, ενώ κάθε μέλος της δεύτερης συνδέεται το πολύ με ένα μέλος της πρώτης. Έτσι σχέση “Ένα – προς - Πολλά” έχουμε όταν το πρωτεύον κλειδί ενός πίνακα αντιστοιχεί σε πολλές εγγραφές σε έναν άλλο πίνακα.

Παραδείγματα σχέσης “Ένα – προς - Πολλά” είναι:

Πελάτης – Παραγγελίες του πελάτη

Μαθητής – Βαθμοί του μαθητή

Κάθε προμηθευτής προμηθεύει πολλά προϊόντα και κάθε προϊόν προμηθεύεται από έναν μόνο προμηθευτή.

**130. Τι είναι λειτουργικές εξαρτήσεις (functional dependencies); Ποιες μορφές λειτουργικών εξαρτήσεων γνωρίζετε;**

Μια λειτουργική εξάρτηση σημαίνει ότι χρησιμοποιώντας τη γνωστή τιμή μίας στήλης η αντίστοιχη τιμή ή σύνολο τιμών μιας άλλης στήλης ή ομάδας στηλών μπορεί να καθορίζεται πάντα μοναδικά.

Τετριμμένη λειτουργική Εξάρτηση ( $\rightarrow$ ) έχουμε όταν μία στήλη εξαρτάται από το κλειδί ενός πίνακα.

Π.χ. Από τον κωδικό του υπαλλήλου καθορίζονται **μοναδικά** το επίθετο και το όνομα του υπαλλήλου.

$Emp\_no \rightarrow emp\_lname$

Εξάρτηση πολλαπλών τιμών ( $\rightarrow \rightarrow$ ) έχουμε όταν η τιμή μίας στήλης μπορεί να καθορίζει το **αντίστοιχο σύνολο τιμών** άλλης στήλης ή στηλών.

Π.χ. Ο κωδικός ISBN του βιβλίου καθορίζει πάντα όλους τους συγγραφείς του. Έτσι η ιδιότητα author έχει εξάρτηση πολλαπλών τιμών από την ιδιότητα isbn.

$Isbn \rightarrow \rightarrow authors$

Η θεωρία των λογικών εξαρτήσεων χρησιμοποιείται για την εξάλειψη του πλεονασμού δεδομένων σε έναν πίνακα (κανονικοποίηση).

**131. Ποια είναι τα στάδια ανάπτυξης μίας βάσης δεδομένων;**

- Ανάλυση απαιτήσεων
- Εννοιολογική σχεδίαση
- Λογική Σχεδίαση Βάσης
- Εκλέπτυνση Σχήματος
- Φυσικός Σχεδιασμός Βάσης
- Υλοποίηση Εφαρμογής και Μηχανισμών Ασφάλειας

**132. Τι σημαίνει ο ορισμός Supertype και τι Subtype; Δώστε ένα παράδειγμα για τον κάθε ορισμό.**

Το **Supertype** είναι ένας τύπος οντότητας που έχει σχέσεις (σχέση parent-child) με έναν ή περισσότερους **subtypes** και περιέχει ιδιότητες που είναι κοινές μεταξύ τους.

Τα **Subtype** είναι υποομάδες της οντότητας Supertype και έχουμε μοναδικά χαρακτηριστικά αλλά είναι ξεχωριστά από κάθε subtype.

**133. Τι ονομάζουμε κανονικοποίηση μιας βάσης δεδομένων και πότε λέμε ότι μία σχέση ή πίνακας βρίσκεται σε πρώτη κανονική μορφή;**

**Κανονικοποίηση δεδομένων** είναι μια διαδικασία κατά την οποία οι πίνακες αναδομούνται σε πολλαπλούς μέχρι να εξαλειφθούν οι εξαρτήσεις μεταξύ των στηλών του. Στόχος της κανονικοποίησης είναι η εξάλειψη του πλεονασμού των δεδομένων σε έναν πίνακα.

Ένας πίνακας βρίσκεται σε 1<sup>η</sup> Κανονική Μορφή (1<sup>st</sup> Normal Form) όταν:

- Όλα τα records έχουν τα ίδια πεδία.
- Κανένα πεδίο δεν περιλαμβάνει υπο – πεδία.
- Δεν υπάρχουν επαναλήψεις πεδίων.
- Τα παραπάνω τα επιβάλει το ΣΔΒΔ (DBMS).

**134. Εξηγήστε την οργάνωση αρχείων με δείκτη «index» σε ένα σύστημα διαχείρισης βάσεων δεδομένων.**

Τα ευρετήρια επιτρέπουν την άμεση πρόσβαση στην πληροφορία χρησιμοποιώντας δείκτες προς τις εγγραφές και εξαλείφουν την ανάγκη για εξερεύνηση σειριακά της βάσης δεδομένων (σάρωση πινάκων) ή για ανακατασκευή των αρχείων της.

Για κάθε εγγραφή στη βάση, δημιουργείται μία αντίστοιχη εγγραφή στο ευρετήριο για τη θέση της στο αρχείο. Τα ευρετήρια είναι σε δομές δένδρου για γρήγορη αναζήτηση και ενημερώνονται αυτόματα από το σύστημα της βάσης δεδομένων.

Μπορούμε να έχουμε πολλά ευρετήρια για τον ίδιο πίνακα ανάλογα με την ιδιότητα που θέλουμε να έχουμε άμεση προσπέλαση.

**135. Τι είναι «πρωτεύον κλειδί» (primary key) και τί «εξωτερικό κλειδί» (foreign key) στο σχεσιακό μοντέλο βάσεων δεδομένων;**

Το **πρωτεύον κλειδί (Primary Key)**, είναι ένα πεδίο ή ο ελάχιστος αριθμός πεδίων που μας επιτρέπουν τη μονοσήμαντη αναγνώριση κάθε πλειάδας σε έναν πίνακα. Για παράδειγμα, σε έναν

πίνακα που περιέχει τα στοιχεία των μαθητών ενός σχολείου, το πρωτεύον κλειδί είναι ο αριθμός του μητρώου κάθε μαθητή, αφού βάσει αυτού μπορούμε να αναφερθούμε μονοσήμαντα σε κάθε μαθητή.

Το **εξωτερικό κλειδί (Foreign Key)**, είναι ένα πεδίο ή συνδυασμός πεδίων ενός πίνακα που αντιστοιχούν στο πεδίο ή στα πεδία του πρωτεύοντος κλειδιού ενός άλλου πίνακα. Το ξένο κλειδί καθορίζει το είδος της συσχέτισης των πινάκων, αφού τα δεδομένα του ξένου κλειδιού πρέπει να ταιριάζουν με τα δεδομένα του πρωτεύοντος.

**136. Ποιους τύπους δεδομένων υποστηρίζει η γλώσσα προγραμματισμού σχεσιακών βάσεων δεδομένων SQL;**

NUMBER  
VARCHAR2  
DATE

**137. Εξηγήστε πότε λέμε ένα ότι ένα σύστημα βάσεων δεδομένων λειτουργεί σε περιβάλλον πολλαπλών χρηστών.**

Ένα σύστημα βάσεων δεδομένων λειτουργεί σε περιβάλλον πολλαπλών χρηστών όταν:

- Εξυπηρετεί πολλούς χρήστες ταυτόχρονα (multi-user).
- Λειτουργεί σε περιβάλλον δικτύου και στο μοντέλο πελάτη-εξυπηρετητή (client-server).
- Αποθηκεύει τα δεδομένα σε ένα server ή σε καταναμημένους servers που είναι προσβάσιμοι από όλους τους χρήστες.
- Διασφαλίζεται η εξουσιοδοτημένη πρόσβαση και η ακεραιότητα των δεδομένων.

**138. Εξηγήστε τι είναι τα καταναμημένα συστήματα βάσεων δεδομένων (distributed database systems).**

Οι καταναμημένες Βάσεις Δεδομένων είναι ειδικές Β.Δ., στις οποίες τα δεδομένα είναι αποθηκευμένα σε ξεχωριστές τοπικές Βάσεις που επικοινωνούν με μια κεντρική βάση. Στην κεντρική βάση βρίσκεται η συνολική πληροφορία για τη δομή της ως αυτόνομη οντότητα, ωστόσο η πληροφορία βρίσκεται στις τοπικές βάσεις δεδομένων. Σε κάθε ερώτημα προς την κεντρική βάση, αυτή επικοινωνεί με τις τοπικές βάσεις, προκειμένου να συλλέξει τη ζητούμενη πληροφορία. Οι τοπικές βάσεις μπορεί να είναι του ίδιου τύπου (ομοιογενείς ΚΒΔ) ή διαφορετικού τύπου (ετερογενείς ΚΒΔ).

Πλεονεκτήματα:

- Κατανομή του έργου σε πολλούς υπολογιστές
- Ασφάλεια σε περίπτωση βλάβης.
- Μικρός χρόνος απόκρισης για τα τοπικά δεδομένα.

Μειονεκτήματα:

- Δεν υπάρχει κεντρικός έλεγχος.
- Κίνδυνος να κλαπούν δεδομένα.
- Μεγάλος χρόνος απόκρισης για μη τοπικά δεδομένα.

**139. Εξηγήστε τι ονομάζουμε ανοικτά συστήματα διασύνδεσης βάσεων δεδομένων, όπως τα συστήματα που υποστηρίζουν το «Open Database Connectivity (ODBC)».**

Ο όρος «ανοικτά συστήματα» υποδηλώνει κατά βάση την ανεξαρτησία από συγκεκριμένο προμηθευτή και την υποχρεωτική χρήση προτύπων (Standards) που διασφαλίζουν:

- την αρμονική συνεργασία και λειτουργία μεταξύ συστημάτων και λειτουργικών εφαρμογών διαφορετικών προμηθευτών
- την μέσω δικτύων συνεργασία και εφαρμογών που βρίσκονται σε διαφορετικά υπολογιστικά συστήματα
- τη μεταφερσιμότητα των εφαρμογών
- τη δυνατότητα αύξησης του μεγέθους των μηχανογραφικών συστημάτων χωρίς αλλαγές στη δομή και τη φιλοσοφία.

**140. Τι σημαίνει «Ακεραιότητας Αναφορών-Referential Integrity»; Δώστε ένα παράδειγμα.**

Ακεραιότητα Αναφοράς (Referential Integrity) είναι ο κανόνας των σχεσιακών βάσεων δεδομένων που επιβάλλει ότι όταν γίνονται μεταβολές στα περιεχόμενα της βάσης, δε διαταράσσεται η ακεραιότητα των υφιστάμενων σχέσεων μεταξύ των πινάκων της βάσης. Έτσι για παράδειγμα, όταν σβήνεται ένα πρωτεύον κλειδί, πρέπει να σβηστούν και οι εγγραφές που το περιέχουν σαν εξωτερικό κλειδί, αλλιώς οι εγγραφές αυτές θα είναι ασύνδετες.

Παραδείγματα:

Έχουμε την οντότητα πελάτες και την οντότητα παραγγελίες. Κάθε παραγγελία περιέχει τον κωδικό του πελάτη που έκανε την παραγγελία.

Όταν καταχωρείται η παραγγελία θα πρέπει να ελέγχεται αν έχει ήδη καταχωρηθεί ο πελάτης, αλλιώς να μη γίνεται δεκτή.

Αν διαγραφεί ο πελάτης, τότε οι παραγγελίες του θα περιέχουν έναν κωδικό πελάτη που δεν αντιστοιχεί σε κανέναν υπαρκτό πελάτη. Αυτό μπορεί να δημιουργήσει πολλά προβλήματα, όταν για παράδειγμα τυπωθεί το τιμολόγιο της παραγγελίας, ενώ δε θα υπάρχουν τα στοιχεία του πελάτη. Οι λύσεις είναι δύο, ή δε θα σβήνονται οι πελάτες, ή όταν σβήνονται, θα πρέπει ταυτόχρονα να σβήνονται και οι παραγγελίες τους.

**141. Να αναφέρετε τα μοντέλα βάσεων δεδομένων που γνωρίζετε και ποιες είναι οι διαφορές τους.**

Τα τρία μοντέλα Βάσεων Δεδομένων είναι:

Hierarchical (Ιεραρχικό), Network (Δικτυωτό), Relational (Σχεσιακό)

ΙΕΡΑΡΧΙΚΟ: Σε αυτές τις ΒΔ οι πληροφορίες αναπαρίστανται με Δενδρικής μορφής δομές δεδομένων και συνδέονται μεταξύ τους με συνδέσμους “links”.

ΔΙΚΤΥΩΤΟ: Η διαφορά των Δικτυωτών βάσεων από τις Ιεραρχικές είναι ότι στις πρώτες κάποιο record μπορεί να έχει ακαθόριστο αριθμό πατρικών (parent) records.

ΣΧΕΣΙΑΚΟ: Στην περίπτωση των σχεσιακών βάσεων δεδομένων, οι πληροφορίες συνδέονται μεταξύ τους με σχέσεις που απορρέουν από τα κοινά πεδία διαφορετικών αρχείων. Σε αυτό το είδος Βάσεων Δεδομένων, τα αρχεία λέγονται “tables” (πίνακες), τα records “rows” (γραμμές) και τα πεδία “columns” (στήλες).



**142. Εξηγήστε τι είναι «γλώσσα χειρισμού δεδομένων» (Data Manipulation Language) και τι «γλώσσα ορισμού δεδομένων» (Data Definition Language), τα δυο βασικά τμήματα που χωρίζεται η γλώσσα διαχείρισης βάσεων δεδομένων SQL.**

**Γλώσσα χειρισμού δεδομένων (DML)** είναι το σύνολο των εντολών της SQL που αφορούν τη διαχείριση των δεδομένων μας. Αυτές οι εντολές αφορούν την αναζήτηση εγγραφών / πεδίων (select), τη διαγραφή (delete), την ενημέρωση (update) και την εισαγωγή (insert).

**Γλώσσα ορισμού δεδομένων (DDL)** είναι το σύνολο των εντολών της SQL που αφορούν τη δημιουργία του σχήματος της βάσης. Αυτές οι εντολές, αφορούν τη δημιουργία / τροποποίηση / διαγραφή πινάκων (create table / alter table / drop table), συνόλων, όψεων, χρηστών κ.α.

**143. Ποιοι είναι οι τρεις πρώτοι κανόνες κανονικοποίησης; Δώστε ένα παράδειγμα για τον κάθε ένα από αυτούς.**

Για το παράδειγμα θα χρησιμοποιήσουμε έναν πίνακα με τα πεδία:

ΚΩΔΙΚΟΣ            ΜΑΘΗΤΗ,ΟΝΟΜΑ,ΕΠΩΝΥΜΟ,ΗΜ.            ΓΕΝΝΗΣΗΣ,ΣΥΜΒΟΥΛΟΣ,ΤΗΛ  
ΣΥΜΒΟΥΛΟΥ,ΚΩΔΙΚΟΣ   ΜΑΘΗΜΑΤΟΣ   1,ΠΕΡΙΓΡΑΦΗ   ΜΑΘΗΜΑΤΟΣ   1,ΚΩΔΙΚΟΣ  
ΜΑΘΗΜΑΤΟΣ2,ΠΕΡΙΓΡΑΦΗ ΜΑΘΗΜΑΤΟΣ2.

**1<sup>ος</sup> κανόνας κανονικοποίησης:** Στην πρώτη μορφή κανονικοποίησης βρίσκεται μια σχέση όταν: κάθε όνομα γνωρίσματος(στήλης) είναι μοναδικό, όλες οι τιμές κάθε γνωρίσματος είναι ίδιου τύπου, κάθε τιμή γνωρίσματος είναι ατομική,τα γνωρίσματα δεν έχουν διάταξη από αριστερά προς τα δεξιά, οι συστοιχίες δεν έχουν διάταξη από πάνω προς τα κάτω, δεν υπάρχουν 2 ίδιες συστοιχίες σε μία σχέση. Έστω ότι έχουμε έναν πίνακα ΜΑΘΗΜΑΤΑ ο οποίος είναι μέσα σε έναν άλλον πίνακα ΜΑΘΗΤΕΣ. Σε αυτόν τον κανόνα,κάθε τιμή γνωρίσματος είναι ατομική αλλά επειδή πολλά πεδία είναι επαναλαμβανόμενα καταχωρούνται σε ένα πεδίο το ΜΑΘΗΜΑ(ομάδα πολλών τιμών).Έτσι αυτά έρχονται σε σύγκρουση με τον πρώτο κανόνα κανονικοποίησης άρα,διασπάμε τον αρχικό πίνακα σε δυο ξεχωριστούς(ΜΑΘΗΜΑ,ΜΑΘΗΤΕΣ)

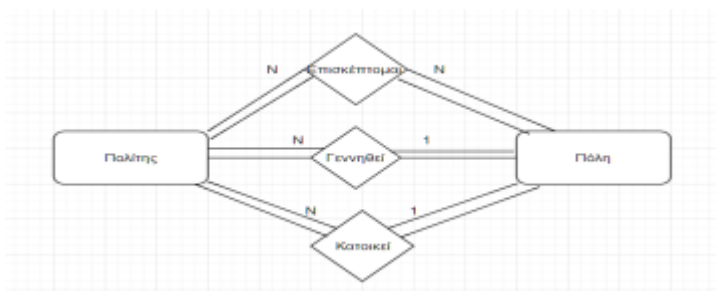
**2<sup>ος</sup> κανόνας κανονικοποίησης:** Σε 2<sup>η</sup> μορφή κανονικοποίησης βρίσκεται μια σχέση εάν είναι σε 1<sup>η</sup> μορφή κανονικοποίησης και επιπλέον κάθε γνώρισμα που δεν είναι κλειδί εξαρτάται ολόκληρο από το πρωτεύον κλειδί και όχι από ένα τμήμα του. Στο παράδειγμα μας με τους πίνακες ΜΑΘΗΤΕΣ, ΜΑΘΗΜΑ,στο πίνακα μαθημάτων θα δημιουργήσουμε ένα πρωτεύον κλειδί που θα συνδυάζει τον ΚΩΔΙΚΟ ΜΑΘΗΤΗ+ΚΩΔΙΚΟΣ ΜΑΘΗΜΑΤΟΣ. Ο ΚΩΔΙΚΟΣ ΜΑΘΗΤΗ δεν θα είναι μοναδικό κλειδί από μόνο του στον πίνακα γιατί ένας μαθητής μπορεί να έχει πολλά μαθήματα. Ανάλογα τον ΚΩΔΙΚΟ ΜΑΘΗΜΑΤΟΣ δεν θα είναι μοναδικό κλειδί από μόνο του σε αυτόν τον πίνακα,γιατί ένα μάθημα μπορεί να το έχουν πολλοί μαθητές. Έτσι τα πεδία μας ΠΕΡΙΓΡΑΦΗ ΜΑΘΗΜΑΤΟΣ,ΕΚΠΑΙΔΕΥΤΗΣ ΜΑΘΗΜΑΤΟΣ είναι τα ίδια για κάθε μάθημα(εξαρτώνται μόνο από τον κωδικό Μαθητή) ανεξάρτητα από τον σπουδαστή που παίρνει το μάθημα.

**3<sup>ος</sup> κανόνας κανονικοποίησης:** Μια σχέση βρίσκεται σε 3<sup>η</sup> μορφή κανονικοποίησης εάν είναι σε δεύτερη και δεν περιέχει μεταβατικές συναρτήσεις. Το πεδίο ΤΗΛ ΣΥΜΒΟΥΛΟΥ δεν περιέχει πληροφορίες για τον μαθητή άρα πρέπει να δημιουργήσουμε ξεχωριστό πίνακα για τους Σύμβουλους.

**144. Τι είναι το Μοντέλο Οντοτήτων Συσχετίσεων(ΜΟΣ); Παρουσιάστε σχηματικά πώς απεικονίζονται στο ΜΟΣ οι οντότητες (entity), οι ιδιότητες των οντοτήτων (attributes), τα πρωτεύοντα κλειδιά (primary key) και οι σχέσεις (relation type και identifying relationship type).**

Είναι ένα μοντέλο σχεδίασης το οποίο το χρησιμοποιούμε για να περιγράψουμε με γραφικά σύμβολα τα δεδομένα ως οντότητες, συσχετίσεις και γνωρίσματα αυτών. Χρησιμοποιείται κατά το στάδιο του εννοιολογικού σχεδιασμού της ΒΔ. Σχηματικά, οι οντότητες «μπαίνουν» σε ένα παραλληλόγραμμο, οι ιδιότητες αυτών σε έναν κύκλο, τα πρωτεύοντα κλειδιά υπογραμμίζονται και οι σχέσεις «μπαίνουν» σε ένα ρόμβο.

**145. Έχουμε δύο οντότητες: Η μία είναι Πολίτης και η άλλη Πόλη. Για αυτές τις δύο οντότητες αναπαραστήστε σχηματικά, με τη βοήθεια της γλώσσας ERDish, τις ακόλουθες σχέσεις: 1. Κάθε Άτομο πρέπει να έχει γεννηθεί μόνο σε μία Πόλη / Κάθε Πόλη μπορεί να είναι γενέτειρα ενός ή περισσότερων Πολιτών 2. Κάθε Πολίτης πρέπει να έχει μόνιμη κατοικία σε μία μόνο Πόλη / Κάθε Πόλη μπορεί να είναι ο τόπος κατοικίας ενός ή περισσότερων Πολιτών 3. Κάθε Πολίτης μπορεί να είναι επισκέπτης μίας ή περισσότερων Πόλεων / Κάθε Πόλη πρέπει να την έχει επισκεφθεί ένας ή περισσότεροι Πολίτες.**



**146. Αναλύστε τις βασικές διαφορές μεταξύ C και C++, όσον αφορά στα χαρακτηριστικά που προσθέτει η C++.**

- Ορισμοί μεταβλητών μπορούν να δοθούν σε οποιοδήποτε σημείο
- Σχόλια της μιας γραμμής μπορούν να οριστούν με τους χαρακτήρες //
- Μεταβλητές μπορούν να οριστούν ως αναφορά
- Συναρτήσεις μπορούν να οριστούν πολυμορφικά
- Οι δηλώσεις δομών ορίζουν αυτόματα και έναν νέο τύπο
- Οι τελεστές new και delete[] δεσμεύουν και ελευθερώνουν και μνήμη
- Στα ορίσματα των συναρτήσεων μπορούν να δοθούν αρχικές τιμές
- Ορίζεται ο τύπος bool για λογικές μεταβλητές

**147. Ποια είναι η διαφορά μεταξύ του αντικειμένου και της κλάσης; Δώστε ένα παράδειγμα.**

Η διαφορά είναι ότι το αντικείμενο (object) είναι το στιγμιότυπο μίας κλάσης, δηλαδή αυτή καθαυτή η δομή δεδομένων (με αποκλειστικά δεσμευμένο χώρο στη μνήμη) βασισμένη στο «καλούπι» που προσφέρει η κλάση. Παραδείγματος χάρη, σε μία αντικειμενοστρεφή γλώσσα προγραμματισμού θα μπορούσαμε να ορίσουμε κάποια κλάση ονόματι BankAccount, η οποία αναπαριστά έναν τραπεζικό λογαριασμό, και να δηλώσουμε ένα αντικείμενο της με όνομα MyAccount. Το αντικείμενο αυτό θα

έχει δεσμεύσει χώρο στη μνήμη με βάση τις μεταβλητές και τις μεθόδους που περιγράψαμε όταν δηλώσαμε την κλάση. Έτσι, στο αντικείμενο θα μπορούσε να περιέχεται ένα γνώρισμα Balance (=υπόλοιπο) και μία μέθοδος GetBalance (=επίστρεψε το υπόλοιπο). Ακολούθως θα μπορούσαμε να δημιουργήσουμε ακόμα ένα ή περισσότερα αντικείμενα της ίδιας κλάσης τα οποία θα είναι διαφορετικές δομές δεδομένων (διαφορετικοί τραπεζικοί λογαριασμοί στο παράδειγμα). Τα αντικείμενα μίας κλάσης μπορούν να προσπελάσουν τα ιδιωτικά περιεχόμενα άλλων αντικειμένων της ίδιας κλάσης.

**148. Ποιες είναι οι διαφορές μεταξύ του δομημένου προγραμματισμού και του προγραμματισμού οδηγούμενου από γεγονότα;**

Σε μια εφαρμογή που έχει αναπτυχθεί με τη φιλοσοφία του δομημένου προγραμματισμού, η εκτέλεσή της ξεκινά από την αρχική εντολή του προ- γράμματος και η ροή εκτέλεσής της είναι καθορισμένη από τις διαδικασίες και τις συναρτήσεις που περιλαμβάνει το πρόγραμμα. Σύμφωνα με τα παραπάνω το δομικό στοιχείο του προγράμματος αποτελούν οι διαδικασίες και οι συναρτήσεις. Σε όλη τη διάρκεια της εκτέλεση της εφαρμογής, το πρόγραμμα διατηρεί τον έλεγχο της, ενώ στο χρήστη έχει ανατεθεί δευτερεύων ρόλος και απλά πληκτρολογεί κάποια δεδομένα, όταν το πρόγραμμα το απαιτεί. Αντίθετα με τον οδηγούμενο από γεγονότα προγραμματισμό, ο χρήστης με την έναρξη της εκτέλεσης της εφαρμογής αποκτά τον έλεγχο της και αποφασίζει, χρησιμοποιώντας το πληκτρολόγιο ή το ποντίκι, ποιο τμήμα του προγράμματος θα εκτελεστεί. Κάθε ενέργεια του χρήστη δημιουργεί μηνύματα με τη μορφή γεγονότων, τα οποία αντιλαμβάνεται το πρόγραμμα και ανταποκρίνεται σε αυτά.

**149. Αναφέρετε τι καθορίζει η κλάση (class) ενός αντικειμένου και τι το στιγμιότυπο (instance) ενός αντικειμένου σε μια αντικειμενοστραφή γλώσσα προγραμματισμού.**

Στον αντικειμενοστραφή προγραμματισμό όλα είναι αντικείμενα και χρησιμοποιούνται για αποθήκευση δεδομένων όπως και για την εκτέλεση υπολογισμών πάνω σε δεδομένα.. Τα αντικείμενα λένε το ένα στο άλλο τι να κάνει μέσω ανταλλαγής μηνυμάτων. Για να εγερθεί ένα αίτημα σε κάποιο αντικείμενο στέλνουμε ένα μήνυμα προς αυτό με αποτέλεσμα την κλήση μιας συνάρτησης του αντικειμένου. Κάθε αντικείμενο έχει τη δική του μνήμη η οποία αποτελείται από άλλα αντικείμενα. Κάθε αντικείμενο έχει τύπο: Κάθε αντικείμενο είναι στιγμιότυπο μιας κλάσης (κλάση  $\approx$  τύπος). Η κλάση καθορίζει τα χαρακτηριστικά και τις συμπεριφορές αντικειμένων: αντικείμενα της ίδιας κλάσης μπορούν να αποδεχθούν τα ίδια μηνύματα.

**150. Τι είναι ο constructor και τι ο destructor σε μια κλάση; Ποιος είναι ο λόγος χρήσης τους;**

Σε κάθε κλάση μπορεί να οριστεί μια συνάρτηση κατασκευής (constructor) με όνομα ίδιο με αυτό της κλάσης και μια συνάρτηση καταστροφής (destructor) με όνομα το όνομα της κλάσης με το πρόθεμα ~. Η συνάρτηση κατασκευής καλείται κάθε φορά που δημιουργείται ένα νέο αντικείμενο (σε επίπεδο καθολικό, τοπικό, ή με new). Η συνάρτηση καταστροφής καλείται κάθε φορά που παύει να υπάρχει ένα αντικείμενο δηλαδή αντίστοιχα όταν τελειώνει το πρόγραμμα, όταν η ροή βγαίνει από το τοπικό τμήμα, ή καλείται η delete. Το όρισμα που δηλώνουμε στη συνάρτηση κατασκευής επιτρέπει προσδιορισμό ιδιοτήτων του αντικειμένου που δημιουργούμε (π.χ. τον αριθμό στοιχείων σε μια στοίβα) ή αρχικών τιμών. Η συνάρτηση καταστροφής δε δέχεται κάποιο όρισμα. Οι συναρτήσεις αυτές μπορούν να χρησιμοποιηθούν για να διαχειριστούν τη μνήμη αντικειμένων που απαιτούν τη χρήση δυναμικής μνήμης με κατάλληλες κλήσεις στις new και delete.

**151. Δώστε τον ορισμό των πιο σημαντικών χαρακτηριστικών του αντικειμενοστραφούς προγραμματισμού: Ενθυλάκωση (Encapsulation), Πολυμορφισμός (Polymorphism), Κληρονομικότητα (Inheritance)**

Κληρονομικότητα στον αντικειμενοστραφή προγραμματισμό αναφέρεται στη δυνατότητα επαναχρησιμοποίησης κώδικα από ήδη υπάρχοντα αντικείμενα, στη δημιουργία νέων αντικειμένων τα οποία μοιράζονται ιδιότητες με ήδη υπάρχοντα ή και τα δύο ανάλογα τη γλώσσα προγραμματισμού. Ένα αντικείμενο αντικείμενο αποτελεί ένα στιγμιότυπο μιας κλάσης, οι κλάσεις μπορούν να κληρονομούν από άλλες κλάσεις τόσο τα γνωρίσματα όσο και της μεθόδους. Ενθυλάκωση δεδομένων (data encapsulation) καλείται η ιδιότητα που προσφέρουν οι κλάσεις να «κρύβουν» τα ιδιωτικά δεδομένα τους από το υπόλοιπο πρόγραμμα και να εξασφαλίζουν πως μόνο μέσω των δημόσιων μεθόδων τους θα μπορούν αυτά να προσπελαστούν. Αυτή η τακτική παρουσιάζει μόνο οφέλη καθώς εξαναγκάζει κάθε εξωτερικό πρόγραμμα να φιλτράρει το χειρισμό που επιθυμεί να κάνει στα πεδία μίας κλάσης μέσω των ελέγχων που μπορούν να περιέχονται στις δημόσιες μεθόδους της κλάσης. Πολυμορφισμός είναι η δυνατότητα εκτέλεσης διαφορετικών λειτουργιών μιας μεθόδου που είναι κοινή σε διαφορετικές κλάσεις (συνήθως κλάση-γονέα και κλάση-παιδί), ανάλογα με το αντικείμενο της κλάσης που την καλεί.

**152. Αναλύστε τις διαφορετικές τεχνικές προγραμματισμού procedural (διαδικαστικός), structured (δομημένος) και object-oriented (αντικειμενοστραφή).**

Δομημένος προγραμματισμός (structured programming) ή Διαδικαστικός προγραμματισμός (procedural programming) είναι μία προσέγγιση στον προγραμματισμό, η οποία βασίζεται στην έννοια της κλήσης διαδικασίας. Η διαδικασία, γνωστή επίσης και ως ρουτίνα, υπορουτίνα, μέθοδος ή συνάρτηση (δεν σχετίζεται άμεσα με τη μαθηματική έννοια της συνάρτησης), είναι απλά ένα αυτοτελές σύνολο εντολών προς εκτέλεση. Ο δομημένος προγραμματισμός βασίζεται στην αρχή του διαίρει και βασίλευε, καθώς διασπά το βασικό πρόβλημα σε μικρότερα υποπροβλήματα (γνωστά επίσης και ως εργασίες). Κάθε εργασία με πολύπλοκη περιγραφή διαιρείται σε μικρότερες, έως ότου οι εργασίες να είναι αρκετά μικρές, περιεκτικές και εύκολες προς κατανόηση.

Αντικειμενοστραφή προγραμματισμό (object-oriented programming), ονομάζουμε ένα προγραμματιστικό υπόδειγμα το οποίο εμφανίστηκε στα τέλη της δεκαετίας του 1960 και καθιερώθηκε κατά τη δεκαετία του 1990, αντικαθιστώντας σε μεγάλο βαθμό το παραδοσιακό υπόδειγμα του δομημένου προγραμματισμού. Πρόκειται για μία μεθοδολογία ανάπτυξης προγραμμάτων, υποστηριζόμενη από κατάλληλες γλώσσες προγραμματισμού, όπου ο χειρισμός σχετιζόμενων δεδομένων και των διαδικασιών που επενεργούν σε αυτά γίνεται από κοινού, μέσω μίας δομής δεδομένων που τα περιβάλλει ως αυτόνομη οντότητα με ταυτότητα και δικά της χαρακτηριστικά. Αυτή η δομή δεδομένων καλείται αντικείμενο και αποτελεί πραγματικό στιγμιότυπο στη μνήμη ενός σύνθετου, και πιθανώς οριζόμενου από τον χρήστη, τύπου δεδομένων ονόματι κλάση. Η κλάση προδιαγράφει τόσο δεδομένα όσο και τις διαδικασίες οι οποίες επιδρούν επάνω τους· αυτή υπήρξε η πρωταρχική καινοτομία του αντικειμενοστραφή προγραμματισμού

**153. Αναλύστε ποιός είναι ο ρόλος της οδηγίας #include <iostream>.**

Το `iostream` είναι μέρος της πρότυπης βιβλιοθήκης της γλώσσας προγραμματισμού `C++` για την είσοδο/έξοδο δεδομένων. Στην πραγματικότητα είναι ένα ψευδώνυμο (typedef) για την εξειδίκευση του πρωτύπου κλάσης `basic_iostream` για είσοδο - έξοδο δεδομένων σε μορφή `char`. Παράγεται με διπλή κληρονομικότητα από τα πρότυπα κλάσεων `basic_ostream` και `basic_istream`. Τα αρχικά `io` στο όνομα της βιβλιοθήκης αναφέρονται στις αγγλικές λέξεις `Input/Output Stream` που μεταφράζονται ως είσοδος/έξοδος ροής (stream) δεδομένων. Η βιβλιοθήκη `iostream` παρέχει την βασική είσοδο/έξοδο σε ένα πρόγραμμα της `C++`. Το `cin` είναι αντικείμενο της `istream` (παράφραση του αγγλικού `see-in` που σημαίνει δεξ είσοδο), και τα `cout` (παράφραση του αγγλικού `see-out` που σημαίνει δεξ έξοδο), `cerr` (παράφραση του αγγλικού `see-error` που σημαίνει δεξ έξοδο λαθών), και `clog` (παράφραση του αγγλικού `see-log` που σημαίνει δεξ έξοδο αρχείου καταγραφής) αντικείμενα της `ostream` για

την ροή δεδομένων στην τυπική έξοδο, την έξοδο λαθών και την έξοδο στο αρχείο καταγραφής αντίστοιχα. Όλα αυτά τα αντικείμενα βρίσκονται κάτω από το χώρο ονομάτων (namespace) std

**154. Δώστε ένα παράδειγμα σε C++ που να εκτυπώνει τα παρακάτω μηνύματα το ένα κάτω από το άλλο: "Ένα απλό Πρόγραμμα" , "Είναι γραμμένο σε C++" , "Η C++ είναι αντικειμενοστραφής γλώσσα προγραμματισμού" .**

```
#include <iostream>

int main() {
    cout << "Ένα απλό πρόγραμμα\n";
    cout << "Είναι γραμμένο σε C++\n";
    cout << "Η C++ είναι αντικειμενοστραφής γλώσσα προγραμματισμού \n";

    return 0;
}
```

**155. Αναλύστε τι είναι το λάθος σε χρόνο εκτέλεσης και τι λάθος κατά την μεταγλώττιση. Να αναφέρετε από δύο αιτίες που προκαλούν αυτά τα λάθη.**

Τα λάθη του προγράμματος είναι γενικά δύο ειδών, λογικά και συντακτικά. Τα λογικά λάθη εμφανίζονται μόνο στην εκτέλεση, ενώ τα συντακτικά λάθη στο στάδιο της μεταγλώττισης. Τα λογικά λάθη που είναι τα πλέον σοβαρά και δύσκολα στη διόρθωσή τους, οφείλονται σε σφάλματα κατά την υλοποίηση του αλγορίθμου, ενώ τα συντακτικά οφείλονται σε αναγραμματισμούς ονομάτων εντολών, παράληψη δήλωσης δεδομένων και πρέπει πάντα να διορθωθούν, ώστε να παραχθεί το τελικό εκτελέσιμο πρόγραμμα. Ο μεταγλωττιστής ανιχνεύει λοιπόν τα συντακτικά λάθη και εμφανίζει κατάλληλα διαγνωστικά μηνύματα. Το στάδιο που ακολουθεί είναι η διόρθωση των λαθών. Το διορθωμένο πρόγραμμα επαναυποβάλλεται για μεταγλώττιση και η διαδικασία αυτή επαναλαμβάνεται, μέχρις ότου εξαληφθούν πλήρως όλα τα λάθη.

**156. Αναλύστε τις διαφορές μεταξύ if και else if όσον αφορά στον έλεγχο και στη χρήση τους. Δώστε παράδειγμα.**

Η εντολή if χρησιμοποιείται όταν θέλουμε να εκτελέσουμε κάποιες εντολές μόνο όταν ικανοποιείται κάποια συνθήκη:

```
if (συνθήκη)
{ εντολές; }
else
{ εντολές; }
```

Μπορούν να συνδυαστούν πολλές εντολές if μεταξύ τους, όταν θέλουμε να ελέγξουμε περισσότερες από δύο περιπτώσεις, όπως στο ακόλουθο παράδειγμα:

```
if (x == 1) {
    cout << "x is one." << endl;
}
else if (x == 2) {
    cout << "x is two." << endl;
}
else if (x == 3) {
    cout << "x is three." << endl;
}
else if (x == 4) {
```

```
cout << "x is four." << endl;
}
else {
cout << "x is not between 1-4." << endl;
}
```

### 157. Αναλύστε τη χρήση των πινάκων χαρακτήρων string στον προγραμματισμό με C++.

Αντίθετα με τα strings σε άλλες γλώσσες προγραμματισμού (C, PASCAL) που είναι απλώς πίνακες χαρακτήρων, στη C++ τα strings είναι κανονικά αντικείμενα, που υλοποιούνται με την κλάση string (στο namespace std). Φυσικά, για λόγους συμβατότητας με τη C, υπάρχει πλήρης υποστήριξη των strings υπό την μορφή πινάκων χαρακτήρων (char \* ή char []).

### 158. Αναλύστε τις διαφορές μεταξύ for, while και do while, όσον αφορά στον έλεγχο των συνθηκών τους. Δώστε από ένα παράδειγμα σε C++.

Η for εκτελείται για προκαθορισμένο αριθμό επαναλήψεων

```
for (int i = 1; i < 10; i += 3)
{
cout << i << endl;
}
```

Η while εκτελείται μόνο εφόσον η συνθήκη της είναι TRUE

```
int x = 1;
while (x <10) {
cout << x << endl;
x+=3;
}
```

Η do while εκτελείται τουλάχιστον μία φορά ανεξαρτήτως της συνθήκης της

```
int x = 1;
do {
cout << x << endl;
x+=3;
}
while (x <10);
```

### 159. Αναλύστε τη χρήση της υπερφόρτωσης σε συναρτήσεις και κατασκευαστές.

Με την υπερφόρτωση συναρτήσεων μπορούμε να ορίσουμε πολλές συναρτήσεις με το ίδιο όνομα, έχοντας διαφορετικά σύνολα παραμέτρων (διαφορετικούς τύπους ή διαφορετικό αριθμό ορισμάτων). Το ίδιο ισχύει και για την υπερφόρτωση κατασκευαστών, μπορούμε να ορίσουμε διαφορετικούς κατασκευαστές για ένα αντικείμενο έχοντας διαφορετικά σύνολα παραμέτρων (διαφορετικούς τύπους ή διαφορετικό αριθμό ορισμάτων).



**160. Τι είναι ένα string στην C++ και σε τι διαφέρει από ένα αλφαριθμητικό C-string;**

Στο C, οι συμβολοσειρές είναι απλά συστοιχίες char, οι οποίες, κατά σύμβαση, τελειώνουν με ένα byte NULL. Στη C++, οι συμβολοσειρές (std :: string) είναι αντικείμενα με όλη τη σχετική αυτοματοποιημένη διαχείριση μνήμης και ελέγχου που τα κάνει πολύ ασφαλέστερα και ευκολότερα στη χρήση.

**161. Αναλύστε τη χρήση των πινάκων χαρακτήρων string στον προγραμματισμό με C++.**

Στην C++ χρησιμοποιούμε πίνακες χαρακτήρων string για την αποθήκευση μιας συλλογής δεδομένων. Κάθε string-κελί του πίνακα το αντιμετωπίζουμε ως ένα object αυξάνοντας έτσι την ευκολία χρησιμοποίησης τους από το χρήστη.

**162. Δημιουργήστε σε C++ ένα πρόγραμμα το οποίο θα δέχεται από τον χρήστη ένα αριθμό και θα επιστρέφει ως τιμή το παραγοντικό του αριθμού. Χρησιμοποιήστε συναρτήσεις αναδρομής. Παράδειγμα το παραγοντικό του 7! είναι: 5040 (1\*2\*3\*4\*5\*6\*7). Γενικά:  $N! = (1*2*3*...*N)$  και  $0! = 1$** 

```
#include <iostream>
#include <string>
using namespace std;

int main(){
    string num;
    int x;
    cout<<("Dwse arithmo");
    cin>>num;
    x=atoi(num);
    x=factorial(num);
    cout<<"To paragwntiko tou arithmou pou dwsate einai:"<<x;

int factorial(int n)
{
    if(n > 1)
        return n * factorial(n - 1);
    else
        return 1;
}
```



- 163. Δημιουργήστε το πρόγραμμα εύρεσης λέξης string σε C++. Το πρόγραμμα δημιουργεί δύο πίνακες string που ο πρώτος περιέχει την πρόταση «C++ is an object oriented Language and is a superset of C». Ο δεύτερος περιέχει την λέξη «is». Το πρόγραμμα πρέπει να ψάχνει στον πρώτο πίνακα να βρει εάν υπάρχει η λέξη που περιέχεται στο δεύτερο πίνακα και να επιστρέφει τη θέση στην οποία βρίσκεται καθώς και πόσες φορές επαναλαμβάνεται**

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string first []="C++ is an object oriented Language and is a superset of C ";
    string second[]="is";
    int found=first[0].find(second[0]);
    int sum=0;
    while (found!=string::npos){
        cout<<"found at:"<<found<<endl;
        sum+=1;
        found=first[0].find(second[0],found+1);
        if(found==string::npos)
            cout<<"The string doesnt exists\n"<<endl;
    }

    cout<<sum;

    return 0;
}
```

- 164. Γράψτε δυο συναρτήσεις που θα υπολογίζουν τον Μέγιστο Κοινό Διαιρέτη (ΜΚΔ) και το Ελάχιστο κοινό πολλαπλάσιο (ΕΚΠ) δυο ακέραιων αριθμών. ΜΚΔ είναι ο μεγαλύτερος από τους κοινούς διαιρέτες δηλαδή τους αριθμούς που τους διαιρούν ακριβώς. Κοινό πολλαπλάσιο (Κ.Π.) δύο ή περισσότερων αριθμών λέγεται κάθε ακέραιος, εκτός από το 0, που το συναντάμε κοινό στα πολλαπλάσια των δύο αριθμών. Ο μικρότερος από τα κοινά πολλαπλάσια κάποιων ακέραιων αριθμών λέγεται Ελάχιστο Κοινό Πολλαπλάσιο (Ε.Κ.Π.).**

```
void megistos_koinos_diairetis(int a,int b){
    int m;
    if(b>a){
        m=a;
        a=b;
        b=m;
    }
    if(b!=0){
        do
        {
            m=a-(a/b)* b;
            if(m==0)
                break;
            a=b;
            b=m;
        }while(m>0);
        cout<<"Megistos Koinos Diairetis = "<<b;
    }
}
```

```
}
else
{
cout<<"megistos koinos diairetis = "<<a;
}

}

void megistos_koinos_diairetis(int a,int b){
int x,y,max,min,ekp,c;

do
{
    x=a;
    y=b;
}while(!(x>0 && y>0));
if(x>y){
max=x;
min=y;
}
else{
max=y;
min=x;
}

c = 1;

do{
ekp=max*c;
c++;
} while ((ekp%min)!=0) ;

cout<<"\n\nEKP=%d\n---"<<ekp) ;

cout<<("\n\n");
}
```

**165. Αναλύστε τους όρους "κληρονομικότητα" και "πολυεπίπεδη κληρονομικότητα" (multi-level inheritance). Δώστε παραδείγματα σε C++ κώδικα.**

Η κληρονομικότητα είναι ένα από τα βασικά χαρακτηριστικά μιας αντικειμενοστραφούς γλώσσας προγραμματισμού. Επιτρέπει στους προγραμματιστές να αντλούν μια νέα κλάση από την υπάρχουσα κλάση. Η παράγωγη κλάση κληρονομεί τα χαρακτηριστικά της βασικής κλάσης. Υπάρχουν διάφορα μοντέλα κληρονομιάς στον προγραμματισμό C++.

Στην C ++, μπορούμε να αντλήσουμε μια κλάση από την κλάση βάσης(base class) αλλά μπορούμε επίσης να αντλήσουμε μια κλάση από την παράγωγη κλάση. Αυτή η μορφή κληρονομικότητας είναι γνωστή ως πολυεπίπεδη κληρονομικότητα

Παράδειγμα:

```
class A
{
    class A
    ... ..
};
class B: public A
{
    ... ..
};
class C: public B
{
    ... ..
};
```

Εδώ, η κλάση B προέρχεται από την κλάση βάσης A και η κλάση C προέρχεται από την παράγωγη κλάση B.

Άλλο παράδειγμα κληρονομικότητας:

```
#include <iostream>

using namespace std;

// Base class
class Shape {
public:
    void setWidth(int w) {
        width = w;
    }
    void setHeight(int h) {
        height = h;
    }

protected:
    int width;
    int height;
};

// Derived class
class Rectangle: public Shape {
public:
    int getArea() {
        return (width * height);
    }
};
```

```
    }  
};  
  
int main(void) {  
    Rectangle Rect;  
  
    Rect.setWidth(5);  
    Rect.setHeight(7);  
  
    // Print the area of the object.  
    cout << "Total area: " << Rect.getArea() << endl;  
  
    return 0;  
}
```

Σε αυτό το παράδειγμα έχουμε μια βασική κλάση Shape και από αυτή κληρονομεί τα βασικά χαρακτηριστικά της κλάσης μια νέα κλάση με όνομα Rectangle. Η Rectangle, έχει όλα τα γνωρίσματα του γονέα της και μια επιπλέον συνάρτηση που υπολογίζει το εμβαδόν του ορθογωνίου.

**166. Ποια χρωματικά μοντέλα υπάρχουν; Δώστε μια σύντομη περιγραφή για τη λειτουργία τους και για τα χρώματα που χρησιμοποιούνται στο κάθε ένα.**

### Χρωματικό μοντέλο RGB

Η ονομασία RGB του χρωματικού μοντέλου προέρχεται από τα αρχικά Red Green Blue (Κόκκινο Πράσινο Μπλέ). Με τα βασικά αυτά χρώματα κωδικοποιούνται όλα τα χρώματα που μπορεί να εμφανιστούν σε μία οθόνη υπολογιστή. Στην 8bit μορφή του χρωματικού αυτού μοντέλου κάθε χρώμα μπορεί να παρασταθεί με μία τριάδα αριθμών από 0 έως 255. Το μοντέλο βασίζεται στο γεγονός ότι, όταν μία οθόνη δεν εκπέμπει φως, εμφανίζεται μαύρη. Τα υπόλοιπα χρώματα δημιουργούνται με υπέρθεση - ανάμιξη των τριών βασικών με συγκεκριμένη αναλογία.

Μερικά παραδείγματα χρωμάτων δίνονται παρακάτω στην 8bit αυτή έκδοση του μοντέλου:

- Μαύρο: (0,0,0)
- Λευκό: (255,255,255)
- Κόκκινο: (255,0,0)
- Πράσινο: (0,255,0)
- Πορτοκαλί: (255,102,0)

### Χρωματικό μοντέλο CMYK (Χρώμα CMYK)

Πολλές φορές θα έχετε παρατηρήσει ότι όταν τυπώνετε μια έγχρωμη εικόνα από τον υπολογιστή, η εικόνα που τυπώνεται αποκλίνει χρωματικά σε πολλά σημεία από αυτή που βλέπετε στον υπολογιστή. Κι αυτό γιατί δεν μπορούν να χρησιμοποιηθούν στην εκτύπωση μελάνια των τριών χρωμάτων που χρησιμοποιείται στο RGB(κόκκινο, πράσινο, μπλε). Η επικάλυψη δύο και μόνο εξ αυτών δημιουργεί μαύρο. Χρειάζεται επομένως μια διαφορετική προσέγγιση. Στην εκτύπωση των εντύπων χρησιμοποιείται ευρέως το σύστημα C=Cyan M=Magenta Y=Yellow που είναι συμπληρωματικό μοντέλο του RGB. Με τα τρία αυτά χρώματα δημιουργούνται τα δευτερογενή Κόκκινο – Πράσινο – Μπλε ως εξής:

- Κόκκινο: Ματζέντα + Κίτρινο

- Πράσινο: Κίτρινο + Γαλάζιο
- Μπλε: Γαλάζιο + Ματζέντα

Το μοντέλο αυτό βασίζεται στο γεγονός ότι το υπόβαθρο της εκτύπωσης είναι το λευκό χαρτί που ανακλά όλα τα χρώματα (μήκη κύματος). Κάθε βασικό χρώμα που προστίθεται με ένα [μελάνι](#) απορροφά ορισμένα χρώματα και αποδίδει τα υπόλοιπα. Για παράδειγμα το κίτρινο μελάνι απορροφά το μπλε χρώμα και αφήνει το πράσινο και το κόκκινο να ανακλαστεί. Εδώ ο συνδυασμός των τριών βασικών χρωμάτων δίνει το μαύρο χρώμα (πλήρης απορρόφηση των ακτινοβολιών). Για το λόγο αυτό το μοντέλο CMY χαρακτηρίζεται και ως "αφαιρετικό". Μαύρο χρώμα, επίσης, προκύπτει από το συνδυασμό ενός βασικού και του αντίθετου δευτερογενούς:

- Μαύρο: Γαλάζιο + Ματζέντα + Κίτρινο
- Μαύρο: Γαλάζιο + Κόκκινο
- Μαύρο: Ματζέντα + Πράσινο
- Μαύρο: Κίτρινο + Μπλε

Τα μελάνια, όμως, από τη φύση τους δεν μπορούν να αποδώσουν συγκεκριμένα μήκη κύματος – χρώματα (όπως τα εικονοστοιχεία (pixels) μίας οθόνης) αλλά, μία πιο ευρεία περιοχή του χρωματικού φάσματος. Το αποτέλεσμα είναι ο συνδυασμός των τριών βασικών χρωμάτων να δίνει ένα καφετί χρώμα αντί για το μαύρο. Για το λόγο αυτό προστέθηκε στο μοντέλο CMY και το μαύρο μελάνι, με αποτέλεσμα να προκύψει το χρωματικό μοντέλο CMYK (Cyan – Magenta – Yellow – Black). Πρακτικά στην εκτύπωση δεν χρησιμοποιείται σήμερα το CMY μοντέλο αλλά το CMYK.

### **Χρωματικό μοντέλο HSB**

Αναλύει το χρώμα σε τρεις παραμέτρους:

#### **Απόχρωση (Hue)**

Το χρώμα που αντανακλάται ή μεταδίδεται διαμέσου ενός αντικειμένου. Μετράται ως περιοχή στον τυπικό δίσκο χρωμάτων που εκφράζεται ως μοίρα μεταξύ του 0° και του 360°. Σε κοινή χρήση, η απόχρωση αναγνωρίζεται από το όνομα του χρώματος όπως κόκκινη, πορτοκαλί ή πράσινη.

#### **Κορεσμός (Saturation)**

Ισχύς ή καθαρότητα του χρώματος (ορισμένες φορές αποκαλείται chroma). Ο κορεσμός αντιπροσωπεύει την ποσότητα του γκρι σε αναλογία με την απόχρωση και μετράται ως ποσοστό από 0% (γκρι) έως 100% (πλήρης κορεσμός). Στον τυπικό δίσκο χρωμάτων ο κορεσμός αυξάνεται από το κέντρο προς την άκρη.

#### **Φωτεινότητα (Brightness)**

Η σχετική φωτεινότητα ή σκοτεινία του χρώματος μετράται συνήθως ως ποσοστό από το 0% (μαύρο) έως το 100% (λευκό). Διαδικτυακά Χρώματα

Η γλώσσα γραφής των ιστοσελίδων HTML, χρησιμοποιεί για την απεικόνιση των χρωμάτων το μοντέλο RGB στην δεκαεξαδική του μορφή.

### **Χρωματικό μοντέλο Lab**

Το μοντέλο αυτό αναλύει το χρώμα του κάθε εικονοστοιχείου σε τρεις παραμέτρους L,a,b:

- Ένταση φωτεινότητας (Luminosity).

- Χρωματική θέση ανάμεσα στο κόκκινο και το συμπληρωματικό του πράσινο (a).
- Χρωματική θέση ανάμεσα μπλε και το συμπληρωματικό του κίτρινο (b).

**167. Ποιο χρώμα ονομάζουμε πρωτεύον συμπληρωματικό και ποια δευτερεύοντα; Δώστε δύο (2) παραδείγματα, καθώς και αιτιολογία για το λόγο χρήσης τους.**

Τα **βασικά ή πρωτεύοντα χρώματα** είναι τρία, **το κόκκινο, το μπλε και το κίτρινο**.

Αυτά χρησιμοποιούνται ως βάση για την παραγωγή των άλλων χρωμάτων και τη δημιουργία αρμονικών χρωματικών συνθέσεων. Ονομάζονται βασικά επειδή δεν μπορούν να παραχθούν από την μείξη άλλων χρωμάτων.

Η ανάμειξη των βασικών χρωμάτων ανά δύο παράγει τα **σύνθετα ή δευτερεύοντα χρώματα**. Η ανάμειξη του κόκκινου και του κίτρινου δίνει το **πορτοκαλί**, η ανάμειξη του μπλε και του κόκκινου δίνει το **μωβ** και η ανάμειξη του κίτρινου και του μπλε δίνει το **πράσινο**. Η ανάμειξη και των τριών παράγει ένα καφέ-γκρίζο χρώμα.

Τι είναι όμως τα **συμπληρωματικά ή αντίθετα χρώματα**; Για κάθε βασικό χρώμα υπάρχει ένα δεύτερο που βρίσκεται σε ιδιαίτερη σχέση μαζί του. Είναι το χρώμα που δημιουργήθηκε χωρίς τη συμμετοχή αυτού του βασικού. Έτσι π. χ. για το κόκκινο έχουμε το πράσινο, που δημιουργείται με την ανάμειξη των άλλων δύο βασικών χρωμάτων του κίτρινου και του μπλε. Με τον τρόπο αυτό λοιπόν, δημιουργούνται τρία ζευγάρια χρωμάτων: **κόκκινο-πράσινο, κίτρινο-μωβ, μπλε-πορτοκαλί**. Τα χρώματα αυτών των ζευγαριών είναι μεταξύ τους αντίθετα και συμπληρωματικά. Όταν δηλ. τοποθετήσουμε ένα κόκκινο αντικείμενο σε πράσινο φόντο η εντύπωση που δημιουργείται είναι η αντίθεση. Φαντάζομαι μπορείς να το αντιληφθείς και μόνο απ' τις χρωματιστές λέξεις βλέποντάς τις δίπλα-δίπλα. Αυτός ο λόγος για τον οποίο το κόκκινο με το πράσινο, το κίτρινο με το μωβ, και το μπλε με το πορτοκαλί ονομάζονται αντίθετα χρώματα.

Συμπληρωματικά απ' την άλλη ονομάζονται γιατί σαν ζευγάρι το κάθε ένα από αυτά συμπληρώνει το χρωματικό φάσμα. Όπως έγραψα παραπάνω το κόκκινο, το κίτρινο και το μπλε μπορούν να δημιουργήσουν όλα τα υπόλοιπα χρώματα, ολόκληρο δηλ. το χρωματικό φάσμα. Παίρνοντας ένα ζευγάρι συμπληρωματικών είναι σαν να παίρνουμε και τα τρία βασικά χρώματα, το ένα μόνο του και τα άλλα δύο σε ανάμειξη:

$KOKKINO + ΠΡΑΣΙΝΟ = KOKKINO + (ΜΠΛΕ + ΚΙΤΡΙΝΟ)$

$ΚΙΤΡΙΝΟ + ΜΩΒ = ΚΙΤΡΙΝΟ + (ΚΟΚΚΙΝΟ + ΜΠΛΕ)$

$ΜΠΛΕ + ΠΟΡΤΟΚΑΛΙ = ΜΠΛΕ + (ΚΟΚΚΙΝΟ + ΚΙΤΡΙΝΟ)$

**168. Ποια είναι τα χαρακτηριστικά μιας ψηφιογραφικής εικόνας (bitmap);**

Τα χαρακτηριστικά μια ψηφιογραφικής εικόνας (bitmap) είναι:

- Η ανάλυση ή διακριτική ικανότητα (resolution), που εκφράζει την πυκνότητα των εικονοστοιχείων (pixels) του γραφικού bitmap σε κάθε διάσταση και μετριέται σε κουκίδες ανά ίντσα (dots per inch, dpi).
- Βάθος χρώματος (color depth) ονομάζεται ο αριθμός των bits που χρησιμοποιούνται για την κωδικοποίηση της πληροφορίας της χρωματικής απόχρωσης κάθε εικονοστοιχείου. Αν το βάθος χρώματος είναι n bits, μια εικόνα μπορεί να διαθέτει μέχρι  $2^n$  διαφορετικά χρώματα.
- Το μέγεθός της (size), που εκφράζει την απαιτούμενη χωρητικότητα για την αποθήκευση του γραφικού bitmap σε κάποιο αποθηκευτικό μέσο.

**169. Ποιο είναι το μέγεθος μίας ασυμπίεστης εικόνας bitmap (σε bytes), διαστάσεων 100x100 pixels, αν είναι :α) true-color; β) grayscale;**

Το μέγεθος σε bits μιας εικόνας προκύπτει αν πολλαπλασιάσουμε την ανάλυση με το βάθος χρώματος. Στη συνέχεια πρέπει να διαιρέσουμε το αποτέλεσμα με το 8 για να το μετατρέψουμε σε bytes. Ο τύπος υπολογισμού είναι:

$(\text{Ύψος} \times \text{Πλάτος} \times \text{Βάθος χρώματος}) / 8$ .

Στην περίπτωση του **true-color** το βάθος χρώματος είναι 24bit. Συνεπώς το μέγεθος της ασυμπίεστης εικόνας bitmap, διαστάσεων 100x100 pixels είναι:

$(100 \times 100 \text{ pixels} \times 24 \text{ bits}) / 8 = 30000 \text{ bytes}$ .

Το βάθος χρώματος στην περίπτωση του **grayscale** είναι 8 bits. Συνεπώς το μέγεθος είναι:

$(100 \times 100 \text{ pixels} \times 8 \text{ bits}) / 8 = 10000 \text{ bytes}$ .

**170. Να αναφέρετε τέσσερις μορφές αρχείων ψηφιογραφικών εικόνων (bitmap) και πού χρησιμοποιούνται, συνήθως, αντίστοιχα.**

Οι πιο συχνά χρησιμοποιούμενες μορφές αρχείων γραφικών bitmap είναι:

- **Bmp**: αποτελεί πρότυπο γραφικών bitmap και παλαιότερα χρησιμοποιούνταν πολύ, σε πληθώρα προγραμμάτων ζωγραφικής και επεξεργασίας εικόνας.
- **Gif**: χρησιμοποιούνται στη διακίνηση εικόνων κινούμενων εικόνων στο Διαδίκτυο.
- **Jpg**: χρησιμοποιούνται στη διακίνηση εικόνων στα δίκτυα υπολογιστών και το Διαδίκτυο, λόγω της καλής συμπίεσης που επιτυγχάνουν.
- **Tiff**: χρησιμοποιείται στην αποθήκευση εικόνων που εισάγονται στον υπολογιστή για επεξεργασία και είναι η πλέον διαδεδομένη μορφή αποθήκευσης για τις γραφικές τέχνες.
- **Pcx**: χρησιμοποιείται στο πακέτο ζωγραφικής Paintbrush των Windows και σε πληθώρα προγραμμάτων ζωγραφικής και επεξεργασίας εικόνας.

**171. Ποια η χρησιμότητα μάσκας κατά την επεξεργασία εικόνας;**

Οι μάσκες επιτρέπουν την απομόνωση και προστασία περιοχών της εικόνας κατά την εφαρμογή χρωματικών αλλαγών, φίλτρων ή άλλων επιδράσεων στην υπόλοιπη εικόνα. Όταν επιλέγουμε μια περιοχή της εικόνας, η μη επιλεγμένη περιοχή χαρακτηρίζεται προστατευμένη από αλλαγές. Οι μάσκες μπορούν να χρησιμοποιηθούν επίσης σε σύνθετες ενέργειες επί των εικόνων, όπως η σταδιακή εφαρμογή χρωματικών αλλαγών ή επιδράσεων φίλτρων κλπ. Μπορούμε να θεωρήσουμε τις μάσκες σαν ένα εναλλακτικό τρόπο επιλογής εικόνας ή τμήματος αυτής, με τις ίδιες ακριβώς ιδιότητες.



**172. Τι είναι πρότυπο συμπίεσης εικόνας JPEG; Ποια είναι τα μειονεκτήματα και τα πλεονεκτήματα του;**

Η τύπου **JPEG** αποθήκευση σχεδιάστηκε από το Joint Photographic Expert Group και είχε ως στόχο να επιτύχει τη μέγιστη δυνατή συμπίεση μιας εικόνας, χρησιμοποιώντας τεχνικές συμπίεσης με απώλειες. Αυτό σημαίνει την οριστική απώλεια πληροφορίας, δηλαδή από τη στιγμή που μια εικόνα συμπιεστεί και μετά αποσυμπιεστεί, η παραγόμενη εικόνα δεν είναι ακριβώς ίδια με την αρχική. Παρ' όλα αυτά, συνήθως οι απώλειες δε γίνονται αντιληπτές από το ανθρώπινο μάτι. Τα πλεονεκτήματα εδώ είναι, ότι ενώ οι περισσότερες από τις άλλες μεθόδους πετυχαίνουν συμπίεση 3:1, η JPEG τεχνική πετυχαίνει λόγους συμπίεσης 20:1 ή και περισσότερο. Τα μειονεκτήματα είναι ότι η ευελιξία αυτού του τύπου αρχείων μπορεί να οδηγήσει σε προβλήματα ασυμβατότητας. Καθώς η εικόνα συμπιέζεται όταν αποθηκεύεται, ο περαιτέρω χειρισμός εικόνων σε JPEG μορφή, μπορεί να οδηγήσει σε χειροτέρευση του ποσοστού αλλοίωσης της ποιότητας.

**173. Τι ονομάζουμε Bit Depth, τι ονομάζουμε Color Depth και ποιες κατηγορίες εικόνων γνωρίζετε με βάση το bit depth;**

**Βάθος χρώματος (Color Depth)**, είναι το εύρος των δυαδικών ψηφίων που θα χρησιμοποιήσει ένας υπολογιστής για να αναπαραστήσει το χρώμα κάθε εικονοστοιχείου (πίξελ, pixel) μιας εικόνας. Το εύρος αυτό εκφράζεται ως δύναμη του 2 (επειδή η αναπαράσταση στον υπολογιστή είναι δυαδική) και, κατά συνέπεια, μια εικόνα μπορεί να έχει βάθος χρώματος:

- $2^1 = 2$ : Ασπρόμαυρη εικόνα (χωρίς διαβαθμίσεις γκριζου).
- $2^8 = 256$  χρώματα (ή αποχρώσεις του γκριζου).
- $2^{16} = 65536$  χρώματα. Η εικόνα με αυτό το βάθος χρώματος αναφέρεται και ως Highcolor.
- $2^{24} = 16.777.216$  χρώματα. Η εικόνα με αυτό το βάθος χρώματος αναφέρεται και ως Truicolor.
- $2^{48}$  = Αυτό το βάθος χρώματος υπερβαίνει την διακριτική ικανότητα του ανθρώπινου οφθαλμού. Χρησιμοποιείται, ωστόσο, για πρακτικούς λόγους, από πολλούς σαρωτές.

**Βάθος μπιτ (Bit Depth)**, είναι ακριβώς το ίδιο με το Color Depth, απεικονίζει δηλαδή τον αριθμό των bits που χρησιμοποιούνται για να αναπαραστήσουν το pixel μιας εικόνας.

**174. Ποια είναι τα πλεονεκτήματα και τα μειονεκτήματα των ψηφιογραφικών εικόνων bitmap;**

Τα μειονεκτήματα των γραφικών bitmap είναι:

- Μεγαλύτερο μέγεθος αρχείων σε σχέση με τα διανυσματικά γραφικά.
- Δεν παρέχουν δυνατότητες γεωμετρικών μετασχηματισμών (μετακίνηση, μεγέθυνση, σμίκρυνση, περιστροφή, διάτμηση και αναστροφή) των σχημάτων χωρίς να προκληθούν αλλοιώσεις.
- Το γραφικό bitmap αποτελεί ένα ενιαίο αντικείμενο και δεν επιτρέπει τον άμεσο διαχωρισμό των συστατικών του στοιχείων ούτε και την ανεξάρτητη επεξεργασία του καθενός.
- Η επεξεργασία τους είναι πιο δύσκολη και πιο αργή σε σχέση με τα διανυσματικά γραφικά.

Το πλεονέκτημα των γραφικών bitmap είναι ότι είναι κατάλληλα για φωτορεαλιστικές και τρισδιάστατες απεικονίσεις, διότι προσφέρουν μεγάλο εύρος χρωμάτων, λεπτομερειών και φωτοσκιάσεων.

**175. Αναφέρετε τους τρόπους λήψης και ψηφιοποίησης ψηφιογραφικής εικόνας.**

Μερικοί τρόποι λήψης/ψηφιοποίησης εικόνων bitmap είναι με:

- Προγράμματα δημιουργίας και επεξεργασίας εικόνας.
- Ψηφιοποίηση εικόνων με σαρωτή.
- Σύλληψη εικόνων από την οθόνη του υπολογιστή μέσω κατάλληλου λογισμικού.
- Χρήση ψηφιακής φωτογραφικής μηχανής.
- Ψηφιοποίηση εικόνας βίντεο ή τηλεόρασης μέσω ειδικής κάρτας σύλληψης εικόνας βίντεο.

**176. Με ποια εντολή δημιουργούμε, διαγράφουμε έναν πίνακα και με ποια εντολή διαγράφουμε τα δεδομένα ενός πίνακα;**

## Δημιουργία πίνακα: CREATE TABLE

```
π.χ. CREATE TABLE Persons (
```

```
PersonID int,  
LastName varchar(255),  
FirstName varchar(255),  
Address varchar(255),  
City varchar(255)
```

$$);$$

### Διαγραφή πίνακα: DROP TABLE

$\pi.\chi$ . DROP TABLE Persons;

**Διαγραφή δεδομένων πίνακα:** DELETE FROM TABLE

WHERE *συνθήκη*

**$\pi.\chi$ . DELETE FROM Persons**

WHERE LastName= 'Petrou':

**177. Τι εννοούμε όταν αναφερόμαστε στις όψεις; Ορίστε την "όψη" στην SQL.**

Η **όψη (view)** είναι μια σχέση μιας βάσης δεδομένων, που δεν έχει δημιουργηθεί με κάποια εντολή create table. Με τις όψεις μπορούμε να εμφανίζουμε ορισμένα μόνο πεδία μιας σχέσης ή και κάποιες άλλες πληροφορίες που αν δεν ήταν οι όψεις θα έπρεπε να δίνουμε κάθε φορά πολύπλοκες εντολές για να δούμε τις πληροφορίες που θέλουμε. Μπορούμε να δημιουργούμε, να τροποποιούμε και να διαγράφουμε όσες όψεις θέλουμε, χωρίς να επηρεάζονται καθόλου τα δεδομένα των σχέσεων στις οποίες αναφέρονται οι όψεις. Η όψη μπορεί να χρησιμοποιηθεί ως πίνακας σε ερωτήματα ή υποερώτημα SELECT.

Για να δημιουργήσουμε μια όψη, δίνουμε την εντολή *create view as*, ως εξής :

**Create view  $A\Theta\Lambda HTH\Sigma$  1 as**

(*Select Κωδικός\_Αθλητή, Επώνυμο, Ημνία\_Γέννησης*  
*From ΑΘΛΗΤΗΣ*);

**178. Τί είναι ένα ευρετήριο (index); Δώστε ένα παράδειγμα με χρήση κώδικα SQL.**

Ένα ευρετήριο (index), είναι μια βοηθητική δομή αρχείου που κάνει πιο αποδοτική την αναζήτηση μιας εγγραφής σε ένα αρχείο. Το ευρετήριο καθορίζεται (συνήθως) σε ένα γνώρισμα του αρχείου που καλείται πεδίο ευρετηριοποίησης (indexing field).

π.χ. 

```
CREATE INDEX idx_lastname  
ON Persons (LastName);
```

Παραπάνω δημιουργήσαμε ένα ευρετήριο με όνομα idx\_lastname, πάνω στο πεδίο LastName του πίνακα Persons.

**179. Ποια είναι η χρήση των stored procedures; Δώστε δύο παραδείγματα για τη δομή τους.**

Μια stored procedure (αποθηκευμένη διαδικασία) είναι ένα τμήμα προγράμματος που αποθηκεύεται στον κατάλογο της Βάσης Δεδομένων και στη συνέχεια μπορεί να κληθεί από ένα πρόγραμμα ή μια άλλη stored procedure.

Τα πλεονεκτήματα μίας stored procedure είναι τα εξής:

- Βελτιώνει την απόδοση μιας εφαρμογής. Όταν δημιουργηθεί, μεταγλωττίζεται και αποθηκεύεται στον κατάλογο της Βάσης Δεδομένων. Εκτελείται πιο γρήγορα από απλές SQL εντολές που δεν έχουν μεταγλωτιστεί.
- Μειώνει το επικοινωνιακό φορτίο μεταξύ της εφαρμογής και του εξυπηρετητή Βάσεων Δεδομένων. Αυτό συμβαίνει γιατί, αντί να στέλνονται μη μεταγλωτισμένες SQL εντολές, η εφαρμογή χρειάζεται να στείλει στον εξυπηρετητή μόνο το όνομα της stored procedure για να πάρει τα αποτελέσματα.
- Είναι επαναχρησιμοποιήσιμη για κάθε εφαρμογή. Διευκολύνει τον προγραμματιστή, ο οποίος δε χρειάζεται να δημιουργήσει συναρτήσεις για λειτουργίες που έχουν ήδη υλοποιηθεί σε μια stored procedure. Χρειάζεται απλώς να καλέσει την κατάλληλη stored procedure από το πρόγραμμά του.
- Είναι ασφαλής. Ο διαχειριστής του εξυπηρετή Βάσεων Δεδομένων μπορεί να δώσει δικαιώματα χρήσης σε stored procedures που υπάρχουν αντί να δώσει δικαιώματα επεξεργασίας σε πίνακες της Βάσης Δεδομένων.

Τα μειονεκτήματα των stored procedures είναι τα εξής:

- Αυξάνουν τον φόρτο του εξυπηρετητή Βάσεων Δεδομένων, τόσο από πλευράς επεξεργασίας, όσο και από πλευράς μνήμης. Αντί ο εξυπηρετής να χρησιμοποιείται μόνο για την αποθήκευση και την ανάκτηση δεδομένων, θα πρέπει να εκτελεί ένα πλήθος από απλές έως πολύπλοκες εντολές των stored procedures.
- Περιέχουν ένα σύνολο δηλωτικών εντολών σε SQL και δεν μπορούν να χρησιμοποιηθούν για πολύπλοκα προγράμματα που γίνονται παραδοσιακά με χρήση γλωσσών προγραμματισμού όπως C++, Java κτλ.
- Η ανίχνευση λαθών (debugging) σε μια stored procedure είναι δύσκολη. Κατά τη μεταγλώττιση δεν εμφανίζονται τα γνωστά μηνύματα λάθους που εμφανίζει ένας τυπικός μεταγλωττιστής.

## Παραδείγματα

### 1) DELIMITER //

```
CREATE PROCEDURE GetAllMovies()  
BEGIN  
    SELECT * FROM MOVIES;  
END //  
DELIMITER ;
```

Η πρώτη εντολή είναι η εντολή DELIMITER //. Η εντολή αυτή δε σχετίζεται με τη stored procedure. Η εντολή DELIMITER στη MySQL χρησιμοποιείται για να αλλάξουμε το βασικό delimiter της MySQL που είναι το ελληνικό ερωτηματικό (;) σε κάποιον άλλο. Σε αυτή την περίπτωση ο delimiter αλλάζει από ελληνικό ερωτηματικό (;) σε //, ώστε να μπορούμε να έχουμε πολλαπλές SQL εντολές σε μια stored procedure που να διαχωρίζονται με το ελληνικό ερωτηματικό (;).

Για να δημιουργήσουμε μια νέα stored procedure χρησιμοποιούμε την εντολή CREATE PROCEDURE. Μετά την εντολή CREATE PROCEDURE γράφουμε το όνομα της stored procedure, το οποίο στην περίπτωσή μας είναι το GetAllMovies. Το κυρίως τμήμα της stored procedure είναι ένα block που αρχίζει με την εντολή BEGIN και τελειώνει με την εντολή END. Σε αυτό το τμήμα μπορούμε να γράψουμε SQL εντολές. Μετά την εντολή END χρησιμοποιούμε το delimiter // για να δείξουμε το τέλος της stored procedure. Μετά την εντολή END της stored procedure θα πρέπει οπωσδήποτε να αλλάξουμε πάλι τον delimiter στο ελληνικό ερωτηματικό (;) με χρήση της εντολής DELIMITER ;

### 2) DELIMITER //

```
CREATE PROCEDURE GetMoviesByYear (IN inputYear INT)  
BEGIN  
    SELECT id, title  
    FROM MOVIES  
    WHERE year = inputYear;  
END //  
DELIMITER ;
```

Η παραπάνω stored procedure, είναι procedure με παραμέτρους. Ο τύπος IN είναι ο default τύπος παραμέτρου. Όταν χρησιμοποιούμε το IN για μια παράμετρο, τότε η τιμή της παραμέτρου περνάει στη stored procedure αλλά οποιαδήποτε αλλαγή στην τιμή της παραμέτρου μέσα στη stored procedure δεν αλλάζει την αρχική τιμή της παραμέτρου. Χρησιμοποιήσαμε ως IN παράμετρο το *inputYear* με τύπο δεδομένων INT. Στο block της stored procedure που βρίσκεται ανάμεσα στις εντολές BEGIN και END, γράψαμε μια SQL ερώτηση για να ανακτήσουμε όλες τις ταινίες οι οποίες δημιουργήθηκαν το έτος *inputYear*.

Αν υποθέσουμε ότι θέλουμε να ανακτήσουμε όλες τις ταινίες που δημιουργήθηκαν το έτος 2010. Τότε απλώς περνάμε ως παράμετρο το έτος 2010 στη stored procedure και την καλούμε ως εξής:

```
CALL GetMoviesByYear(2010);
```

**180. Με τη χρήση κώδικα SQL δημιουργήστε ένα πίνακα ο οποίος στην πρώτη στήλη με τίτλο "Κωδικός" και τιμές 053, 109, 027, στη δεύτερη την ιδιότητα με τίτλο: "Όνομα" και τιμές: Παπαδόπουλου, Χατζηκώστα, Καρανίκα, στη τρίτη με τίτλο "Τηλέφωνο" και τιμές 21041111, 21033099, 21023322. Η κάθε στήλη θα έχει και τον αντίστοιχο τύπο δεδομένων.**

```
CREATE TABLE Persons (  
    Κωδικός int,  
    Όνομα varchar(25),  
    Τηλέφωνο varchar(15),  
);  
  
INSERT INTO Persons (Κωδικός, Όνομα, Τηλέφωνο)  
VALUES  
    (053,'Παπαδοπούλου' 21041111),  
    (109,'Χατζηκώστα' 21033099),  
    (027,'Καρανίκα' 21023322);
```

**181. Τί είναι τα ψευδώνυμα και πού τα χρησιμοποιούμε; Εξηγήστε τον ακόλουθο κώδικα:**  
**SELECT E.NAME, S.NAME FROM EMPLOYEE E, EMPLOYEE S WHERE**  
**E.SUPERSN=S.SN**

Τα ψευδώνυμα χρησιμοποιούνται για λόγους συντόμευσης της γραφής της γλώσσας SQL. Ουσιαστικά μπορούμε να δώσουμε σε έναν πίνακα ή ακόμα και σε ένα πεδίο ψευδώνυμο για να μπορούμε να αναφερόμαστε σε αυτά με μεγαλύτερη ευκολία. Είναι δηλαδή, ένας προσωρινός τρόπος μετονομασίας, χωρίς το πραγματικό όνομα του πίνακα ή του πεδίου να αλλάζει στη βάση.

Στο παραπάνω ερώτημα, έχουμε δώσει 2 ψευδώνυμα στον πίνακα EMPLOYEE (E και S). Μετέπειτα, ζητάμε δύο φορές το πεδίο NAME, μια φορά με το ψευδώνυμο E και μια φορά με το ψευδώνυμο S, κάνοντας join στον εαυτό του στα πεδία E.SUPERSN και S.SN.

**182. Αν στο αποτέλεσμα ενός ερωτήματος πάρουμε ως απάντηση πολλαπλές πλειάδες πώς τις εξαλείφουμε; Δώστε ένα παράδειγμα σε SQL.**

Χρησιμοποιούμε την εντολή **DISTINCT**.

π.χ. **SELECT DISTINCT Country**  
**FROM Customers;**

Με το παραπάνω ερώτημα ζητάμε να μας φέρει όλες τις χώρες (Country) που βρίσκονται στον πίνακα Customers, χωρίς όμως να τις επαναλαμβάνει. Αν για παράδειγμα, στο πεδίο Country είχαμε καταχωρημένη πολλές φορές την Ελλάδα, χρησιμοποιώντας το DISTINCT μας τη φέρνει μόνο μία φορά.

**183. Πώς επιτυγχάνουμε τη διάταξη των αποτελεσμάτων του ερωτήματος; Δώστε ένα παράδειγμα όπου θα επιλέγεται από ένα πίνακα το ονοματεπώνυμο ενός σπουδαστή και θα εμφανίζονται τα αποτελέσματα ταξινομημένα κατά αύξουσα αλφαβητική σειρά στο όνομα και κατά φθίνουσα αλφαβητική σειρά στο επώνυμο.**

Με την εντολή **ORDER BY** για **αύξουσα** σειρά και την εντολή **ORDER BY DESC** για **φθίνουσα** σειρά.

π.χ. **Αύξουσα αλφαβητική σειρά**

```
SELECT * FROM Students  
ORDER BY Name, Surname;
```

**Φθίνουσα αλφαβητική σειρά**

```
SELECT * FROM Students  
ORDER BY Surname DESC;
```

**184. Εξηγήστε τον ακόλουθο κώδικα:**

```
SELECT NAME  
FROM WORKER W WHERE EXISTS  
(SELECT * FROM CONNECT C WHERE ID=C.CONNECT_ID AND  
W.Name=C.CONNECT_Name);
```

Το παραπάνω query μας επιστρέφει τα ονόματα των εργατών όπου υπάρχει το id του εργάτη στον πίνακα CONNECT και το όνομα του υπάρχει και στον πίνακα WORKER και στον CONNECT.

**185. Δώστε 3 τύπους συνένωσης στην SQL (όχι το απλό JOIN) και για κάθε έναν από αυτούς δώστε ένα παράδειγμα με χρήση κώδικα SQL.**

- **INNER JOIN:** Το INNER JOIN επιλέγει όλες τις σειρές και από τους δύο πίνακες, εφόσον υπάρχει αντιστοιχία μεταξύ των στηλών. Εάν υπάρχουν εγγραφές στον πίνακα "Orders" που δεν έχουν αντιστοιχίες στους "Customers", αυτές οι παραγγελίες δεν θα εμφανιστούν.  
SELECT Orders.OrderID, Customers.CustomerName  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
- **LEFT JOIN:** Το LEFT JOIN επιστρέφει όλες τις εγγραφές από τον αριστερό πίνακα (Customers), ακόμη και αν δεν υπάρχουν αντιστοιχίσεις στο δεξί πίνακα (Orders).  
SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
ORDER BY Customers.CustomerName;

- **RIGHT JOIN:** Το RIGHT JOIN επιστρέφει όλες τις εγγραφές από το δεξιό πίνακα (Orders), ακόμα και αν δεν υπάρχουν αντιστοιχίσεις στον αριστερό πίνακα (Customers).  
SELECT Orders.OrderID, Customers.LastName, Customers.FirstName  
FROM Orders  
RIGHT JOIN Orders ON Orders.CustomerID = Customers.CustomerID  
ORDER BY Orders.OrderID;

**186. Αναλύστε τη χρήση των συναρτήσεων στον αντικειμενοστραφή προγραμματισμό. Δώστε παράδειγμα πως οι συναρτήσεις καλούνται ως μηνύματα σύνδεσης μεταξύ τάξεων.**

Η συνάρτηση (function) είναι ένας συνδυασμός οδηγιών/εντολών συνδυασμένων για να επιτευχθεί κάποιο αποτέλεσμα. Οι συναρτήσεις μπορούν να δεχθούν δεδομένα εισόδου και να δημιουργήσουν αποτέλεσμα επιστροφής. Εάν μια συνάρτηση δεν επιστρέφει ένα αποτέλεσμα, συνήθως ονομάζεται διαδικασία (procedure).

Απ την άλλη, οι μέθοδοι ("member functions") είναι παρόμοιες με τις συναρτήσεις, ανήκουν σε τάξεις ή αντικείμενα και συνήθως εκφράζουν τις λειτουργίες των αντικειμένων / τάξεων. Για παράδειγμα, ένα αντικείμενο τύπου Window συνήθως θα έχει δημόσιες (public) και ιδιωτικές (private) μεθόδους οι οποίες αντιστοιχούν στις λειτουργίες του αντικειμένου που ανήκουν.

Στον αντικειμενοστραφή προγραμματισμό οι μεταβλητές και οι συναρτήσεις που επενεργούν πάνω σε αυτές τις μεταβλητές ομαδοποιούνται σε μια αφαίρεση που ονομάζουμε κλάση.

Παρακάτω δίνεται η κλάση Box και στη συνέχεια φαίνεται πως οι μέθοδοί της αξιοποιούνται στην ανταλλαγή μηνυμάτων – επικοινωνία μεταξύ κλάσεων από την κλάση BoxDemo.

```
class Box {  
    double width;  
    double height;  
    double depth;  
  
    Box(double x, double y, double z) {  
        width = x;  
        height = y;  
        depth = z;  
    }  
  
    double volume() {  
        return width * height * depth;  
    }  
}
```



```

class BoxDemo {
    Box mybox1 = new Box(10, 20, 15);
    Box mybox2 = new Box(3, 6, 9);
    double vol;
    vol = mybox1.volume();
    System.out.println("Volume is " + vol);
    vol = mybox2.volume();
    System.out.println("Volume is " + vol);
}

```

**187. Ποια από τα ακόλουθα είναι σωστά και ποια είναι λάθος; Αιτιολογήστε την απάντησή σας για κάθε περίπτωση.**

```

Person p1 = new Student();
Person p2 = new PhDStudent();
PhDStudent phd1 = new Student();
Prof t1 = new Person();
Student s1 = new PhDStudent();

```

Προκύπτει λογικά ότι η συσχέτιση μεταξύ των ανωτέρω κλάσεων είναι η εξής:

Student *ISA* Person

Prof *ISA* Person

Phd *ISA* Student (επειδή ισχύει Student *ISA* Person, ως εκ τούτου ισχύει Phd *ISA* Person)

Έτσι, λόγω της κληρονομικότητας οι προτάσεις PhDStudent phd1 = new Student(); και Prof t1 = new Person(); είναι σωστές. Απεναντίας, οι προτάσεις Student s1 = new PhDStudent();, Person p1 = new Student(); και Person p2 = new PhDStudent(); είναι λανθασμένες καθώς αντικείμενα κατώτερης τάξης δε μπορούν να καλέσουν constructors ανώτερων τάξεων.

**188. Γιατί δεν μπορούμε να αφαιρέσουμε πεδία από μια subclass ενώ μπορούμε να προσθέσουμε;**

Από τον ορισμό της υποκλάσης (μια υποκλάση *κληρονομεί και επεκτείνει* τα πεδία και τις μεθόδους της μητέρας κλάσης) δεν είναι δυνατόν η υποκλάση να μην αποτελέσει υπερσύνολο, αναφορικά με τις μεθόδους και τα πεδία της μητέρας κλάσης.

Αρκεί να αναλογιστούμε για παράδειγμα την περίπτωση του *τετραγώνου* και του *κύβου*, όπου εν προκειμένω το *τετράγωνο* είναι η μητέρα κλάση (με δυο τουλάχιστον πεδία μήκος και ύψος) και από την άλλη πλευρά η υποκλάση *κύβος* που επεκτείνει ουσιαστικά το την μητέρα κλάση *τετράγωνο* να μην έχει τουλάχιστον τις δυο επίπεδες διαστάσεις - πεδία μήκος και ύψος αντίστοιχα. Τόσο κατά την αρχική υπόθεση όπου είχε αφαιρεθεί το ένα πεδίο της μητέρας κλάσης (μη κληρονομία) *τετράγωνο* όπου για παράδειγμα η κλάση *κύβος* είχε μόνο 2 πεδία π.χ. μήκος –

βάθος, όσο και στην περίπτωση που δε προστίθενται περεταίρω πεδία (μη επέκταση) η κλάση *κύβος* θα ενέπιπτε στη κλάση *τετράγωνο* και ως εκ τούτου δε θα θεωρείτο υποκλάση.

**189. Ποια είναι η διαφορά μεταξύ ενός Frame και ενός Dialog;**

Ενώ αμφότερα το JFrame και το JDialog κληρονομούν τις ιδιότητες της κλάσης Window, ώστε να μοιράζονται πολλές λειτουργίες, το JFrame είναι ένα κανονικό παράθυρο με (προαιρετικά) κουμπιά και τις διακοσμήσεις, ενώ το JDialog από την άλλη πλευρά δεν έχει κουμπιά μεγιστοποίησης και ελαχιστοποίησης και συνήθως δημιουργείται με στατικές μεθόδους JOptionPane και είναι καλύτερα προσαρμοσμένο στις τροποποιήσεις του αναφορικά με το ότι μπλοκάρει άλλα components μέχρι ο χρήστης να το κλείσει.

**190. Ποια είναι η διαφορά μεταξύ ενός αντικείμενου Menu και ενός MenuBar (σε Java);**

Το Menu και το MenuBar είναι συστατικά στοιχεία πλοήγησης που υποστηρίζουν διάφορες λειτουργίες και εμφανίζουν επιλογές. Μπορούν να έχουν πολλαπλά επίπεδα υπο-Menu και τα στοιχεία Menu μπορούν είτε να ενεργοποιήσουν τις ενέργειες διακομιστή είτε να χρησιμεύσουν ως σύνδεσμοι σε εξωτερικές διευθύνσεις URL. Η κύρια διαφορά μεταξύ του Menu και του MenuBar είναι ότι το Menu εμφανίζεται κάθετα, ενώ το MenuBar εμφανίζεται οριζόντια. Ακόμη από κατασκευαστικής άποψης το Menu είναι υποκλάση του MenuBar. Έτσι, το MenuBar μπορεί να φιλοξενήσει πολλά Menu και κατ επέκταση MenuItems.

**191. Ποια είναι η διαφορά μεταξύ ενός Menu και ενός MenuItem (σε Java);**

Το Menu περιέχει τα στοιχεία του μενού. Το MenuItem εμφανίζει την πραγματική επιλογή που μπορεί να επιλέξει ο χρήστης μέσα σε ένα Menu. Τα στοιχεία του Menu (MenuItem-s) προστίθενται στο μενού με τη μέθοδο addItem ().

**192. Ποια είναι η διαφορά μεταξύ της επανεκκίνησης και της επαναφόρτωσης ενός Java applet;**

Οι λειτουργίες επανεκκίνησης (Restart) και ανανέωσης (Reload) χρησιμοποιούνται για την επανεκκίνηση της εκτέλεσης της εφαρμογής. Η διαφορά μεταξύ αυτών των δύο επιλογών είναι ότι η επαναφόρτωση (Reload) εκφορτώνει την μικροεφαρμογή πριν την επανεκκίνησή της, ενώ η επανεκκίνηση (Restart) όχι. Η επιλογή επαναφόρτωση (Reload) ισοδυναμεί με το κλείσιμο του προγράμματος προβολής μικροεφαρμογών και το άνοιγμα ξανά στην ίδια ιστοσελίδα.

**193. Τι είναι οι constructors στην JAVA; Δώστε ένα παράδειγμα.**

Μια κλάση περιέχει constructors που καλούνται να δημιουργήσουν αντικείμενα από το πρότυπο της κλάσης. Οι δηλώσεις των constructors μοιάζουν με δηλώσεις μεθόδου - εκτός από το ότι χρησιμοποιούν το όνομα της κλάσης και δεν έχουν τύπο επιστροφής. Για παράδειγμα, το ποδήλατο έχει έναν constructor:

```
public Bicycle(int startCadence, int startSpeed, int startGear) {  
    gear = startGear;  
    cadence = startCadence;  
    speed = startSpeed;  
}
```

Για να δημιουργήσουμε ένα νέο αντικείμενο Bicycle που ονομάζεται myBike, ένας constructor καλείται ως εξής:

```
Bicycle myBike = new Bicycle(30, 0, 8);  
new Bicycle(30, 0, 8)
```

Ετσι, ο constructor δεσμεύει χώρο στη μνήμη για το αντικείμενο προετοιμάζοντας τα πεδία του.

**194. Να αναφέρετε από ποια μέρη αποτελείται η Java Virtual Machine και ποιες μεθόδους τρέχει.**

Μια εικονική μηχανή Java (JVM) είναι μια αφηρημένη υπολογιστική μηχανή που επιτρέπει σε έναν υπολογιστή να εκτελεί ένα πρόγραμμα Java. Υπάρχουν τρία συστατικά του JVM: το specification, το implementation, και το instance. Η προδιαγραφή (specification) είναι ένα έγγραφο που περιγράφει επίσημα τι απαιτείται για μια υλοποίηση της JVM. Έχοντας μια ενιαία προδιαγραφή διασφαλίζει ότι όλες οι υλοποιήσεις (implementations) είναι διαλειτουργικές (interoperable). Μια εφαρμογή JVM (JVM implementation) είναι ένα πρόγραμμα υπολογιστή που πληροί τις απαιτήσεις της προδιαγραφής (specification) JVM. Τέλος το στιγμιότυπο (instance) ενός JVM είναι μια υλοποίηση (implementation) που εκτελείται σε μια διαδικασία που εκτελεί ένα πρόγραμμα υπολογιστή που έχει συνταχθεί σε Java bytecode.

**195. Ποιοι είναι οι τύποι μεταβλητών στην Java;**

Οι τύποι μεταβλητών (variables) που εντοπίζονται στη Java είναι οι εξής:

- a) Local variables
- b) Instance variables
- c) Class/Static variables

**196. Τι κάνει το παρακάτω JAVA πρόγραμμα: `int j=5; int k=10; long max=0; max=k>j?k:j;`**

Η σύνταξη βασίζεται στη δομή ελέγχου: `condition ? first_expression : second_expression;`

Έτσι, ελέγχει αν το `k` είναι μεγαλύτερο του `j` και θέτει στο `max` τη τιμή του `k` αλλιώς (αν το `j` είναι μεγαλύτερο) θέτει στο `max` τη τιμή του `j`.

**197. Ποιος ο ρόλος των εξαιρέσεων στα προγράμματα εισόδου / εξόδου; Δώστε δύο παραδείγματα σε JAVA.**

Ο χειρισμός εξαιρέσεων (Exception Handling) είναι η διαδικασία αντίδρασης στην εμφάνιση, κατά τον υπολογισμό, εξαιρέσεων - ανώμαλων ή εξαιρετικών συνθηκών που απαιτούν ειδική επεξεργασία - αλλάζοντας συχνά την κανονική ροή εκτέλεσης του προγράμματος. Συγκεκριμένα, οι εξαιρέσεις εισόδου/εξόδου (IOExceptions) είναι η γενική κατηγορία εξαιρέσεων που παράγονται από αποτυχημένες ή διακοπείσες λειτουργίες εισόδου / εξόδου.

Δυο παραδείγματα χρήσης αυτών παρατίθενται στη συνέχεια όπου εμφανίζονται τα `try-catch` κατά την εγγραφή σε αρχεία και την ανάγνωση αυτών.

```
import java.io.*;
import static java.nio.file.StandardOpenOption.*;
import java.nio.file.*;
import java.io.*;
import java.nio.charset.Charset;
public class Test {

    public static void main(String[] args) {
        String s = "Hello World! ";
        byte data[] = s.getBytes();
        Path p = Paths.get("C:\\Users\\User\\Desktop\\test.txt");
        write_file(p,data);
        read_file(p);
    }

    // Paradeigma xeirismou eggrafhs se arxeio me xrhsh IOException
    public static void write_file(Path p,byte data[]){
        try (OutputStream out = new BufferedOutputStream(
            Files.newOutputStream(p, CREATE, APPEND))) {
            out.write(data, 0, data.length);
        }
        catch (IOException x) {
            System.err.println(x);
        }
    }

    // Paradeigma xeirismou anagnwshs apo arxeio me xrhsh IOException
    public static void read_file(Path p){
        Charset charset = Charset.forName("US-ASCII");
```

```

        try    (BufferedReader    reader    =    Files.newBufferedReader(p,
charset)) {
    String line = null;
        while ((line = reader.readLine()) != null) {
            System.out.println(line);
        }
    }
    catch (IOException x) {
        System.err.format("IOException: %s\n", x);
    }
}
}

```

**198. Δημιουργήστε σε JAVA το παιχνίδι κρεμάλα (hangman). Ο χρήστης εισάγει γράμματα τα οποία και συγκρίνονται με λέξεις που επιλέγονται τυχαία από έναν πίνακα με λέξεις string. Μηνύματα επιτυχίας ή αποτυχίας εμφανίζονται στο χρήστη και τον καθοδηγούν για το αποτέλεσμα. Για διευκόλυνσή σας τοποθετήστε στον πίνακα 5 διαφορετικές λέξεις και το πρόγραμμά σας μην επεκταθείτε στη δημιουργία GUI.**

```

package hangman;
import java.util.Arrays;
import java.util.Scanner;

public class Hangman{
    public static void main(String[] args) {
        String[] words = {"hello", "world", "hungman", "wrote", "that", "program"};
        int randomWordNumber = (int) (Math.random() * words.length);
        char[] enteredLetters = new char[words[randomWordNumber].length()];
        int triesCount = 0;
        boolean wordIsGuessed = false;
        do {
            switch (enterLetter(words[randomWordNumber], enteredLetters)) {
                case 0:
                    triesCount++;
                    break;
                case 1:
                    triesCount++;
                    break;
                case 2:
                    break;
                case 3:
                    wordIsGuessed = true;
                    break;
            }
        } while (! wordIsGuessed);
        System.out.println("\nThe word is " + words[randomWordNumber] +
            " You missed " + (triesCount - findEmptyPosition(enteredLetters)) +
            " time(s)");
    }

    public static int enterLetter(String word, char[] enteredLetters) {
        System.out.print("(Guess) Enter a letter in word ");
        if (! printWord(word, enteredLetters)){return 3;}
    }
}

```

```

System.out.print(" > ");
Scanner input = new Scanner(System.in);
int emptyPosition = findEmptyPosition(enteredLetters);
char userInput = input.nextLine().charAt(0);
if (inEnteredLetters(userInput, enteredLetters)) {
    System.out.println(userInput + " is already in the word");
    return 2;
}
else if (word.contains(String.valueOf(userInput))) {
    enteredLetters[emptyPosition] = userInput;
    return 1;
}
else {
    System.out.println(userInput + " is not in the word");
    return 0;
}
}

public static boolean printWord(String word, char[] enteredLetters) {
    boolean asteriskPrinted = false;
    for (int i = 0; i < word.length(); i++) {
        char letter = word.charAt(i);
        if (inEnteredLetters(letter, enteredLetters))
            System.out.print(letter);
        else {
            System.out.print('*');
            asteriskPrinted = true;
        }
    }
    return asteriskPrinted;
}

public static boolean inEnteredLetters(char letter, char[] enteredLetters) {
    return new String(enteredLetters).contains(String.valueOf(letter));
}

public static int findEmptyPosition(char[] enteredLetters) {
    int i = 0;
    while (enteredLetters[i] != '\u0000') i++;
    return i;
}
}

```

- 199. Να ορίσετε σε JAVA την κλάση Tires (Λάστιχα) με τις ακόλουθές ιδιότητες^^ Πλάτος της επιφάνειας του ελαστικού (inches)radius Ακτίνα του ελαστικού (inches)rim Ζαντα (inches)type Τύπος ελαστικού0 - Παντός Καιρού 1 - Ξηρό2 - Βροχή3 - Χιόνι brand Κατασκευαστής model Μοντέλο year έτος κατασκευής Να δημιουργήσετε constructor για την κλάση αυτή, ο ποιος να παίρνει παραμέτρους για όλα τα στοιχεία της κλάσης.**

```

public class Tyre {

    private static int width;
    private static int radius;
    private static int rim;

```

```
private static String type;
private static String brand;
private static String model;
private static int year;

public Tyre(int wd, int rd, int rm, String tp, String br, String md, int yr) {
    this.width = wd;
    this.radius = rd;
    this.rim = rm;
    this.type = tp;
    this.brand = br;
    this.model = md;
    this.year = yr;
    System.out.println("Tyre created with the following specs:");
    System.out.println("Tyre width= " + this.width);
    System.out.println("Tyre radius= " + this.radius);
    System.out.println("Tyre rim= " + this.rim);
    System.out.println("Tyre type= " + this.type);
    System.out.println("Tyre brand= " + this.brand );
    System.out.println("Tyre model= " + this.model );
    System.out.println("Tyre year= " + this.year );
}

public static void main(String[] args){

    int the_width=185;
    int the_radius=60;
    int the_rim=18;
    String the_type="snow";
    String the_brand="Michelin";
    String the_model="all_terrain_pro";
    int the_year=2017;
    Tyre my_car_tyre= newTyre(the_width,the_radius,the_rim,the_type,the_brand,the_model,the_year);

}
}
```

## 200. Ποια είναι η διαφορά μεταξύ μιας abstract class και ενός interface (σε Java);

- 1) Η abstract κλάση μπορεί να έχει αφηρημένες και μη αφηρημένες μεθόδους. Η interface μπορεί να έχει μόνο αφηρημένες μεθόδους.
- 2) Η abstract κλάση δεν υποστηρίζει πολλαπλή κληρονομικότητα. Η interface υποστηρίζει πολλαπλή κληρονομικότητα.
- 3) Η abstract κλάση μπορεί να έχει final, non-final, static και non-static μεταβλητές. Η interface έχει μόνο static και final μεταβλητές.
- 4) Η abstract κλάση μπορεί να προσφέρει την υλοποίηση της interface. Η interface δεν μπορεί να προσφέρει την υλοποίηση abstract κλάσης.
- 5) Το keyword “abstract” χρησιμοποιείται για να δηλώσει abstract κλάση. Στο interface, το αντίστοιχο keyword χρησιμοποιείται για να δηλώσει interface.



6) Μια abstract κλάση μπορεί να επεκτείνει μόνο μία κλάση ή μία abstract κλάση τη φορά. Μια interface μπορεί να επεκτείνει οποιοδήποτε αριθμό interface τη φορά

7) Μια abstract κλάση μπορεί να επεκτείνει μια άλλη συγκεκριμένη (κανονική) κλάση ή abstract κλάση. Μια interface μπορεί να επεκτείνει μόνο μια άλλη interface

8) Σε μια abstract κλάση το keyword “abstract” είναι υποχρεωτικό για να δηλωθεί μια μέθοδος ως abstract. Σε ένα interface το keyword “abstract” είναι προαιρετικό για να δηλωθεί μια μέθοδος ως abstract.

9) Μια abstract κλάση μπορεί να έχει protected και public abstract μεθόδους Μια interface μπορεί να έχει μόνο public abstract μεθόδους

## **201. Τι είναι τα interfaces στην JAVA; Πώς δημιουργούμε ένα interface; Δώστε ένα παράδειγμα.**

Η διεπαφή (interface) μοιάζει με κλάση, αλλά δεν είναι κλάση. Μια διεπαφή μπορεί να έχει μεθόδους και μεταβλητές όπως ακριβώς και η κλάση, αλλά οι μέθοδοι που δηλώνονται στη διεπαφή είναι από προεπιλογή αφηρημένες (abstract) (μόνο υπογραφή μεθόδου, χωρίς σώμα). Επίσης, οι μεταβλητές που δηλώνονται σε μια διεπαφή είναι public static και final από προεπιλογή.

Δεδομένου ότι οι μέθοδοι στα interfaces δεν έχουν σώμα, πρέπει να υλοποιηθούν από τη κλάση πριν αποκτηθεί πρόσβαση. Η κλάση που υλοποιεί interface πρέπει να υλοποιήσει όλες τις μεθόδους αυτού του interface.

```
interface MyInterface{
    public void method1();
    public void method2();
}

class Demo implements MyInterface{
    public void method1(){
        System.out.println("implementation of method1");
    }
    public void method2(){
        System.out.println("implementation of method2");
    }
    public static void main(String arg[]){
        MyInterface obj = new Demo();
        obj.method1();
    }
}
```

**202. Γράψτε σε Java ένα πρόγραμμα που θα μετατρέπει την θερμοκρασία από βαθμούς Fahrenheit σε Celcius σημείωση:  $C = 5 (F - 32) / 9$**

```
import java.util.Scanner;

public class Fahrenheit {
    public static void main(String[] args) {
        float temperature;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter temperatue in Fahrenheit");
        temperature = in.nextInt();
        temperature = ((temperature - 32)*5)/9;
        System.out.println("Temperatue in Celsius = " + temperature);
    }
}
```

**203. Γράψτε πρόγραμμα σε Java, το οποίο θα ζητάει από το χρήστη να εισάγει μία λίστα με 6 ονόματα στην αρχική του σειρά, θα ταξινομεί τα ονόματα με αύξουσα αλφαβητική σειρά και κατόπιν θα εμφανίζει ξανά τη λίστα.**

```
import java.util.ArrayList;
import java.util.Scanner;

public class Test {

    public static void main(String[] args) {
        ArrayList<String> Names = new ArrayList<String>();
        Scanner in = new Scanner(System.in);
        String name;
        String temp;
        for(int i=0;i<5;i++){
            System.out.print("give name:");
            name= in.next();
            Names.add(name);
        }
        System.out.println("List of names:"+Names);
        for(int i=0;i<Names.size();i++){
            for(int j=i+1;j<Names.size();j++){
                if(Names.get(i).compareTo(Names.get(j))>0){
                    temp=Names.get(i);
                    Names.set(i, Names.get(j));
                    Names.set(j,temp );
                }
            }
        }
        System.out.println("List of names in alphabetical order:"+Names);
    }
}
```

**204. Τι είναι το Method Overloading; Δώστε ένα παράδειγμα με χρήση κώδικα Java.**

Η μέθοδος "Overloading" είναι μια δυνατότητα που επιτρέπει σε μια κλάση να έχει περισσότερες από μία μεθόδους με το ίδιο όνομα, εάν τα argument lists τους είναι διαφορετικά. Είναι παρόμοιο με την υπερφόρτωση του constructor που επιτρέπει σε μια κλάση να έχει περισσότερους από έναν constructors που έχουν διαφορετικά argument lists.

```
class DisplayOverloading {
    public void disp(char c) {
        System.out.println(c);
    }
    public void disp(char c, int num) {
        System.out.println(c + " "+num);
    }
}

class Sample{
    public static void main(String args[]) {
        DisplayOverloading obj = new DisplayOverloading();
        obj.disp('a');
        obj.disp('a',10);
    }
}
```

**205. Τι είναι το Method Overriding; Δώστε ένα παράδειγμα με χρήση κώδικα Java.**

Η δήλωση μιας μεθόδου στην υποκλάση που υπάρχει ήδη στην γονική κλάση είναι γνωστή ως μέθοδος overriding. Η παράκαμψη γίνεται έτσι ώστε μια κλάση παιδιού να μπορεί να δώσει τη δική της implementation σε μια μέθοδο που παρέχεται ήδη από τη μητρική κλάση. Σε αυτήν την περίπτωση, η μέθοδος στην γονική κλάση ονομάζεται overridden μέθοδος και η μέθοδος στην κατηγορία παιδιού ονομάζεται overriding μέθοδος. Στο παρακάτω παράδειγμα η eat() στη κλάση Boy κάνει override την eat() στη κλάση Human.

```
class Human{
    public void eat(){
        System.out.println("Human is eating");
    }
}

class Boy extends Human{
    public void eat(){
        System.out.println("Boy is eating");
    }
    public static void main( String args[]) {
        Boy obj = new Boy();
        obj.eat();
    }
}
```

**206. Εξηγήστε αναλυτικά τον παρακάτω κώδικα:**

```
interface MyInterface{ public void method1();public void method2(); }
class XYZ implements MyInterface {
    public void method1(){System.out.println("implementation of method1");}
    public void method2(){System.out.println("implementation of method2");}
    public static void main(String arg[]){MyInterface obj=new XYZ(); obj.method1();}}
```

Αυτός είναι ο τρόπος με τον οποίο μια κλάση υλοποιεί μια διεπαφή. Πρέπει να παρέχει το σώμα όλων των μεθόδων που δηλώνονται στη διεπαφή ή με άλλα λόγια ότι η κλάση πρέπει να εφαρμόσει όλες τις μεθόδους της διεπαφής.

Το interface MyInterface περιέχει τις μεθόδους void method1() και void method2(), των οποίων την υλοποίηση παρέχει η κλάση XYZ (στην ουσία κάνουν από μια εκτύπωση η κάθε μια).

Τέλος η main δημιουργεί ένα αντικείμενο της κλάσης XYZ (το obj) και πάω σε αυτό καλείται η μέθοδος method1() η οποία και τυπώνει το μήνυμα implementation of method1.

**207. Δημιουργήστε (σε Java) το class inventory χρησιμοποιώντας Vector Arrays μέσα στο inventory όπου εμπεριέχονται αντικείμενα τύπου string[ ] με χαρακτηριστικά όπλων π.χ mithril sword, Plate Armour κτλ.**

```
package inventory;

import java.util.Enumeration;
import java.util.Scanner;
import java.util.Vector;

public class Inventory {
    private String inventory_item;

    public Inventory(String item){
        this.inventory_item=item;
        System.out.println(    this.inventory_item+"    added    to    the
inventory");
    }

    public static void main(String[] args) {
        Scanner SC = new Scanner(System.in);
        System.out.print("Define invenotry capacity: ");
        int invenotry_size = SC.nextInt();
        String item;
        Vector<Inventory> invenotry_list = new
Vector<Inventory>(invenotry_size);
        for(int i=0; i<invenotry_size;i++){
            System.out.print("\nset item to be adde to the inventory: ");
            item = SC.next();
            Inventory new_item=new Inventory(item);
```

```
        invenotry_list.addElement(new_item);
    }

    System.out.print("\n\nInventory contains: ");
    for(int i=0; i<invenotry_size;i++){
        System.out.print(invenotry_list.get(i).inventory_item+ ", ");
    }

}

}
```

**208. Δημιουργήστε (σε Java) το πρόγραμμα Print(), που εκτυπώνει μέσω κληρονομικότητας στην οθόνη το αποτέλεσμα των τάξεων:**  
**sum, που προσθέτει αριθμούς integer,**  
**sub, που αφαιρεί αριθμούς integer και**  
**multiply που πολλαπλασιάζει αριθμούς integer**

```
package calculator;

import java.lang.Math;
import java.util.Scanner;

class Summary{
    public static int add(int a, int b){
        return Math.addExact(a, b);
    }
}

class Subtract extends Summary{
    public static int sub(int a, int b){
        int c=b*(-1);
        return add(a, c);
    }
}

class Multiplication extends Summary{
    public static int multi(int a, int b){
        int result=0;
        for(int i=0;i<b;i++){result=add(a, result);}
        return result;
    }
}

public class Calculator {

    private static int x;
    private static int y;
```

```

    public Calculator(int a, int b){
        this.x=a;
        this.y=b;
    }

    public static void main(String[] args) {
        int factor_1,factor_2;
        Scanner SC = new Scanner(System.in);
        System.out.print("\nset factor 1: ");
        factor_1= SC.nextInt();
        System.out.print("\nset factor 2: ");
        factor_2= SC.nextInt();
        Calculator result= new Calculator(factor_1,factor_2);
        print(result);
    }

    public static void print(Calculator input){
        System.out.println("sum="+Summary.add(input.x, input.y));
        System.out.println("sub="+Subtract.sub(input.x, input.y));
        System.out.println("mlt="+Multiplication.multi(input.x, input.y));
    }

}

```

**209. Δημιουργήστε (σε Java) την τάξη Rectangle, την τάξη Square και την τάξη Circle η κάθε μία περιέχει τουλάχιστον 3 μεταβλητές και μια συνάρτηση που υπολογίζει το εμβαδόν τους. Οι τάξεις Square και Circle κληρονομούν από την τάξη Rectangle τις μεταβλητές της, υπολογίζουν και εκτυπώνουν το εμβαδόν τους (εμβαδό κύκλου =  $2 \cdot 3.14 \cdot R \cdot R$ , εμβαδό τετραγώνου =  $Width \cdot Width$ ).**

```

package shapes;

public class Rectangle{
    public int width = 0;
    public int height = 0;
    public static String color[];
    public Rectangle(int w, int h, String cl[]){
        width = w;
        height = h;
        System.arraycopy(color, 0, cl, 0, cl.length);
    }
    public int getArea(){
        return width * height;
    }
}

-----
public class Square extends Rectangle{

```

```

public Square(int w,int h, String cl[]){
    super(w,h,cl);
}
public int getArea(){
    return width * height;
}
}

```

```

-----
public class Circle extends Rectangle{
public Circle(int w,int h,String cl[]){
    super(w,h,cl);
}
public double getCircleArea(){
    return width*2*3.14*3.14;
}
}

```

**210. Να γραφεί συνάρτηση σε OpenGL που να σχεδιάζει ένα επίπεδο (plane).**

```

void drawAPlane(){
    glBegin(GL_QUADS);
        glVertex3f(-1, -1, 0);
        glVertex3f(-1, 1, 0);
        glVertex3f(1, 1, 0);
        glVertex3f(1, -1, 0);
    glEnd();
}

```

**211. Σε τι οφείλουμε το γεγονός ότι η OpenGL μπορεί να δείξει από 256 έως δισεκατομμύρια χρώματα χωρίς να χρειαστεί να ξαναγράψουμε ή να ξανακάνουμε compile τον κώδικα;**

Η OpenGL υποστηρίζει το χρωματικό μοντέλο RGB, σε μια ελαφρώς εκτεταμένη μορφή. Προσθέτει μια τέταρτη συνιστώσα στο χρώμα, η οποία ονομάζεται alpha και παίζει το ρόλο “συντελεστής διαφάνειας”. Ο “συντελεστής διαφάνειας” χρησιμοποιείται όταν συνδυάζονται τα χρώματα μεταξύ τους. Το αναθεωρημένο μοντέλο ονομάζεται μοντέλο RGBA. Τα χρώματα στην OpenGL προσδιορίζονται καθορίζοντας τιμές για το χρωματικό μοντέλο RGBA μεταξύ 0.0 και 1.0. Για παράδειγμα η εντολή `glColor3f (0.0f , 0.0f , 1.0f , 1.0f);` // (R, G, B, A) καθορίζει το μπλέ χρώμα με πλήρη αδιαφάνεια.

**212. Η OpenGL χρησιμοποιεί μια απλή, βασική, μορφή ονοματολογίας για τις εντολές της. Αναφέρατε ποια είναι αυτή και παραθέστε παράδειγμα.**

Οι εντολές OpenGL χρησιμοποιούν το πρόθεμα `gl` και αρχικά κεφαλαία γράμματα για κάθε λέξη που συνθέτει το όνομα της εντολής (π.χ. `glClearColor()`). Πολλές από τις συναρτήσεις της δέχονται προκαθορισμένα ορίσματα (συμβολικές σταθερές) που αρχίζουν με `GL_`, γράφονται με κεφαλαία γράμματα και οι λέξεις διαχωρίζονται με κάτω παύλα (π.χ. `GL_COLOR_BUFFER_BIT`). Πολλές εντολές



χρησιμοποιούν αριθμούς και γράμματα ως επίθυμα (suffix) για να μπορούμε να χρησιμοποιούμε διαφορετικό αριθμό και τύπο ορισμάτων. Π.χ. η εντολή `glVertex3f()`; δηλώνει ότι ως όρισματα δέχεται 3 float αριθμούς.

**213. Αναφέρατε τι ονομάζουμε GLUT στην OpenGL. Για ποιο λόγο δημιουργήθηκε και ποιες βασικές λειτουργίες προσφέρει;**

Το πρότυπο της OpenGL είναι ανεξάρτητο πλατφόρμας, περιορίζεται όμως στη δυνατότητα εμφάνισης του προγράμματος. Η OpenGL Utility Toolkit (GLUT) είναι μια βιβλιοθήκη η οποία προσφέρει εντολές εντολές εισόδου- εξόδου προσφέροντας έτσι δυνατότητα αλληλεπίδρασης του χρήστη του προγράμματος με αυτό. Η βιβλιοθήκη GLUT περιλαμβάνει εντολές απεικόνισης παραθύρων στην οθόνη, δημιουργίας menus, διαχείρισης γεγονότων κλπ. Όλες οι εντολές της ξεκινούν με το πρόθεμα `gl`.

**214. Με ποια εντολή δημιουργούμε ένα τρισδιάστατο ορθογραφικό παράθυρο με διαστάσεις: αριστερά -3, δεξιά 3, πάνω 3, κάτω -3 και clipping planes στα 1 και 100 για το near και far αντίστοιχα (στην OpenGL);**

```
glOrtho (-3.0f, 3.0f, -3.0f, 3.0f, 1.0f, 100.0f);
```

**215. Εξετάστε την ισχύ των παρακάτω προτάσεων δηλώνοντας ΣΩΣΤΟ ή ΛΑΘΟΣ και εξηγώντας την επιλογή σας (Όπου κρίνετε απαραίτητο, παραθέστε παράδειγμα).**

**α) Τα επικρατέστερα Λειτουργικά Συστήματα (Windows, Unix, Linux, Mac OS) υποστηρίζουν την OpenGL.**

Σωστό. Η OpenGL είναι ανεξάρτητη λειτουργικού συστήματος και συνήθως αλληλεπιδρά με την κάρτα γραφικών ώστε να πετύχει ταχύτερη απόδοση γραφικών.

**β) Η OpenGL μπορεί αποκλειστικά να κληθεί (is callable) από τις γλώσσες προγραμματισμού C / C++ (δηλαδή υπάρχει μοναδικό language binding).**

Σωστό. Μια βιβλιοθήκη που υλοποιεί το πρότυπο της OpenGL μπορεί να συνταχθεί σε οποιαδήποτε γλώσσα προγραμματισμού (η OpenGL είναι πρότυπο ανεξάρτητο πλατφόρμας).

**γ) Το μοναδικό περιβάλλον ανάπτυξης προγραμμάτων OpenGL είναι το DEV C++.**

Λάθος. Η OpenGL είναι ένα σύνολο εντολών (Application Programming Interface – API) που μας επιτρέπει την δημιουργία τριδιάστατων γραφικών. Δεν είναι γλώσσα προγραμματισμού αλλά μπορεί να χρησιμοποιηθεί με μια πληθώρα γλωσσών προγραμματισμού (C, C++, Java και άλλες). Συνεπώς DEV C++ δεν είναι το μοναδικό περιβάλλον ανάπτυξης προγραμμάτων.

**δ) Η OpenGL περιέχει εντολές επιλογής (τύπου If ... else).**

Σωστό. Η OpenGL χρησιμοποιεί μια μηχανή καταστάσεων (state machine) για να επικοινωνεί με την εφαρμογή. Σε αυτή την μηχανή καταστάσεων η OpenGL παραμένει διαρκώς σε μια κατάσταση μέχρι να αλλάξει η εφαρμογή την κατάσταση.

**ε) Οι εντολές της OpenGL ξεκινούν με το πρόθεμα gl.**

Σωστό. Οι εντολές OpenGL χρησιμοποιούν το πρόθεμα `gl` και αρχικά κεφαλαία γράμματα για κάθε λέξη που συνθέτει το όνομα της εντολής (π.χ. `glClearColor()`). Πολλές από τις συναρτήσεις της δέχονται

προκαθορισμένα ορίσματα (συμβολικές σταθερές) που αρχίζουν με `GL_`, γράφονται με κεφαλαία γράμματα και οι λέξεις διαχωρίζονται με κάτω παύλα (π.χ. `GL_COLOR_BUFFER_BIT`).

**216. Ποια η λειτουργία των παρακάτω εντολών στην OpenGL και τι παραμέτρους δέχονται; Να δοθεί παράδειγμα.α) `glutInitWindowPosition` β) `glutInitWindowSize` γ) `glutCreateWindow`**

Η `glutInitWindowPosition(int x, int y)`; καθορίζει τη θέση στην οθόνη, στην οποία θα εμφανιστεί το παράθυρο της εφαρμογής (συντεταγμένη της άνω αριστερής κορυφής) όπου `x` παράμετρος η απόσταση του παραθύρου σε pixels από την πάνω αριστερή γωνία της οθόνης στον άξονα `X` και `y` παράμετρος η απόσταση του παραθύρου σε pixels από την πάνω αριστερή γωνία της οθόνης στον άξονα `Y`. π.χ. `glutInitWindowPosition(150,150)`; Το παράθυρο θα τοποθετηθεί 150,150 pixels από την πάνω αριστερή γωνία της οθόνης.

Η `glutInitWindowSize(int width, int height)`; καθορίζει το πλάτος και ύψος του παραθύρου της εφαρμογής σε pixels όπου `width` το πλάτος του παραθύρου σε pixels και `height` το μήκος του παραθύρου σε pixels. Π.χ. `glutInitWindowSize(400, 400)`; // Το παράθυρο θα έχει μέγεθος 400x400 pixels.

Η `glutCreateWindow(char *name)`; εμφανίζει το παράθυρο της εφαρμογής στην οθόνη και του αποδίδει έναν τίτλο.

Π.χ. `glutCreateWindow("IEK ΔΕΛΤΑ OPENGL")`; // Το παράθυρο θα έχει τον τίτλο IEK ΔΕΛΤΑ OPENGL.

**217. Περιγράψτε τρεις βασικές βιβλιοθήκες εντολών της OpenGL που περιέχουν εντολές σχεδίασης, γραφικών και απόδοσης. Σε ποιο σημείο του προγράμματος δηλώνονται;**

OpenGL core library

**Βασική βιβλιοθήκη (OpenGL core library ή GL):**

Η βασική βιβλιοθήκη της OpenGL έχει σχεδιαστεί ως μια βελτιωμένη διεπαφή ανεξάρτητη από το hardware, περιέχει τις κύριες εντολές σχεδίασης όπως η σχεδίαση βασικών γεωμετρικών σχημάτων, ο ορισμός χρωμάτων κλπ. Όλες οι εντολές της βιβλιοθήκης αυτής διακρίνονται από το πρόθεμα `gl`.

Πολλές από τις συναρτήσεις της δέχονται προκαθορισμένα ορίσματα (συμβολικές σταθερές) τα οποία έχουν οριστεί στη βιβλιοθήκη και αντιστοιχούν σε διάφορες παραμέτρους ή καταστάσεις λειτουργίας.

Οι σταθερές αυτές ξεκινούν με το πρόθεμα `GL_`.

**OpenGL Utility Library (GLU):**

Η βιβλιοθήκη GLU είναι χτισμένη πάνω στην κορυφή της OpenGL και περιλαμβάνει συναρτήσεις που εκτελούν σύνθετους αλγορίθμους όπως π.χ. τον καθορισμό μητρώων προβολής και το σχηματισμό σύνθετων καμπυλών και επιφανειών. Κάθε υλοποίηση της OpenGL εμπεριέχει τη βιβλιοθήκη GLU. Όλες οι εντολές της βιβλιοθήκης GLU ξεκινούν με το πρόθεμα `glu_` (π.χ., `gluLookAt`, `gluPerspective`).

**OpenGL Utility Toolkit (GLUT):**

Η OpenGL είναι ένας γρήγορος και ευέλικτος τρόπος για την επικοινωνία με το hardware των γραφικών του υπολογιστή χωρίς να ενδιαφέρουν το χρήστη οι λεπτομέρειες υλοποίησης του. Από την άλλη, δεν προσφέρει καθόλου λειτουργίες GUI (Graphical User Interface), δηλαδή δεν έχει τη δυνατότητα να ανοίξει και να κλείσει παράθυρα στο λειτουργικό σύστημα, να ζωγραφίσει σε αυτά, ούτε να καταλάβει το πάτημα ενός πλήκτρου ή την κίνηση του ποντικιού, ούτε μπορεί να διαβάσει ένα αρχείο από το δίσκο. Μια από τις πιο διαδεδομένες βιβλιοθήκες για αυτό το σκοπό είναι το GLUT (OpenGL Utility Toolkit) που είναι και αυτή σχεδιασμένη (η εργαλειοθήκη) να τρέχει σε πολλά λειτουργικά συστήματα. Η GLUT προσφέρει ένα σύνολο εντολών που αναλαμβάνουν να ανοίξουν και να κλείσουν εύκολα παράθυρα, να καταγράψουν το πάτημα ενός πλήκτρου ή την κίνηση του ποντικιού). Η βιβλιοθήκη αυτή περιλαμβάνει ακόμη εντολές απεικόνισης παραθύρων στην οθόνη, δημιουργία μενού, κλπ. Όλες οι εντολές της ξεκινούν με το πρόθεμα `glut_`.

Οι βιβλιοθήκες που χρησιμοποιούνται δηλώνονται στο πάνω μέρος του προγράμματος από τις εντολές

```
#include <gl\gl.h>
#include <gl\glu.h>
#include <gl\glut.h>
```

**218. Αναφέρετε την λειτουργία της `glutMainLoop()` στην OpenGL. Σε ποιο σημείο πρέπει να καλείται;**

Η `glutMainLoop()` ενεργοποιεί τον κύκλο διαχείρισης γεγονότων (event processing loop). Στον κύκλο αυτό, η εφαρμογή αναμένει επ' άπειρον και ανταποκρίνεται σε γεγονότα, όπως λ.χ. στο πάτημα ενός κουμπιού, στην αλλαγή του σκηνικού ή στην κίνηση του ποντικιού. Από το πρόθεμά της, παρατηρούμε ότι η συγκεκριμένη εντολή εμπεριέχεται στη βιβλιοθήκη GLUT, αφού το πρότυπο της OpenGL, ως πρότυπο ανεξάρτητο πλατφόρμας, δεν ορίζει διαδικασίες εισόδου - εξόδου.

**219. Ποια είναι τα πλεονεκτήματα χρήσης μιας πλατφόρμας CMS (Content Management System), σε σχέση με την δημιουργία εκ νέου ενός web site;**

- Ορθολογική και εύκολη διαδικασία συγγραφής άρθρων.
- Ταχύτερος χρόνος ολοκλήρωσης για νέες σελίδες και αλλαγές.
- Μεγαλύτερη συνοχή στην πλοήγηση του δικτυακού τόπου.
- Δυνατότητα επεκτασιμότητας της ιστοσελίδας.
- Υποστήριξη χρηστών με συγκεκριμένα δικαιώματα.
- Αυξημένη ασφάλεια.
- Μειωμένο κόστος συντήρησης site.
- Πολλές επιλογές για καλό SEO.
- Όμορφο design, γραφικά και εφέ.

**220. Το σύστημα διαχείρισης περιεχομένου (CMS) υποστηρίζει την εγκατάσταση διαφόρων modules. Τι είναι τα modules και γιατί είναι χρήσιμα για ένα CMS;**

Το Module στα CMS, είναι μια συλλογή αρχείων κώδικα, που προσθέτει επιπλέον λειτουργίες στην ιστοσελίδα μας. Απαραίτητη προϋπόθεση για να δουλέψει το module, είναι να εγκαταστήσουμε πρώτα το core code του CMS που θα χρησιμοποιήσουμε. Στη συνέχεια, αν θέλουμε να επεκτείνουμε την ιστοσελίδα μας με επιπλέον λειτουργίες (π.χ υποστήριξη forum, social media buttons κλπ), εγκαθιστούμε το module που μας ενδιαφέρει. Για κάθε CMS η έννοια του module, έχει διάφορες σημασίες. Πιο συγκεκριμένα:

- **Wordpress:** Δεν υπάρχει καθόλου η λέξη modules. Αντί αυτής, χρησιμοποιείται η λέξη plugin και widget.
- **Joomla:** Εδώ τα modules, είναι στην ουσία «κουτιά», τα οποία ενσωματώνονται γύρω από ένα component (π.χ article), προσθέτοντας νέες λειτουργίες.
- **Drupal:** Εδώ τα modules είναι ένας γενικός όρος για κώδικα, που προσθέτει επιπλέον λειτουργίες. Τα modules του Drupal, ουσιαστικά αντιστοιχούν στα plugins που έχει το wordpress.

**221. Στην διάρκεια κατασκευής μιας ιστοσελίδας πολλές φορές καλούμαστε να προσθέσουμε alt και title tags. Τι είναι το καθένα απ' αυτά και για ποιο σκοπό χρησιμοποιούνται; Είναι απαραίτητη ή προαιρετική η συμπλήρωσή τους;**

Τα tags **alt** και **title**, χρησιμοποιούνται για λόγους SEO κατά τη διάρκεια κατασκευής μιας ιστοσελίδας. Όταν λέμε SEO (Search Engine Optimization) εννοούμε όλες τις τεχνικές που χρησιμοποιούνται, για να καταφέρει η ιστοσελίδα μας να βγει ψηλά στις μηχανές αναζήτησης.

Ένας βασικός παράγοντας για πετυχημένο SEO είναι το **title tag**, στο οποίο θα πρέπει να βάζουμε το όνομα της σελίδας. Για να καταλάβουμε πόσο σημαντικό είναι το title tag, αρκεί να δούμε πως η Google είναι το πρώτο που εμφανίζει στα αποτελέσματά της για μια ιστοσελίδα και ακριβώς από κάτω εμφανίζει το description της ιστοσελίδας. Το title λοιπόν θα πρέπει να υποδηλώνει το περιεχόμενο μιας συγκεκριμένης σελίδας.

Ένας ακόμα βασικό παράγοντας SEO, είναι το **alt tag**. Το alt tag, είναι ένας ακόμα βασικός παράγοντας SEO και ουσιαστικά είναι μια επιπρόσθετη πληροφορία για το περιεχόμενο μιας φωτογραφίας. Με το alt tag, περιγράφουμε τι βρίσκεται μέσα στη φωτογραφία και χρησιμοποιείται από Screen Readers, browsers για ανθρώπους με προβλήματα όρασης κ.α.

Τέλος, υπάρχει και το **title tag** που προστίθεται σε μια **φωτογραφία** (επιπλέον από το alt tag) και αυτό περιγράφει ένα πιο γενικό περιεχόμενο της φωτογραφίας. Επίσης το title tag της φωτογραφίας χρησιμοποιείται και σαν tooltip από ορισμένους browsers (Firefox, Opera) εμφανίζοντάς το, όταν αφήνουμε τον κέρσορα του ποντικιού πάνω στη φωτογραφία. Επιτυγχάνουμε δηλαδή, καλύτερη «εμπειρία χρήστη» (user experience), κάτι που οι μηχανές αναζήτησης επιθυμούν ώστε να συμπεριλάβουν μια ιστοσελίδα στα αποτελέσματά τους.

**222. Η επιστήμη του SEO (Serch Engine Optimization), δίνει μεγάλη βαρύτητα στην σωστή συμπλήρωση τριών tags (title tag, keyword tag, description tag). Τι ακριβώς είναι το καθένα από αυτά τα tags και με ποιον τρόπο γίνεται η σωστή επιλογή του περιεχομένου τους, έτσι ώστε η ιστοσελίδα να ανταποκρίνεται στα κριτήρια σωστής διάρθρωσης της επιστήμης SEO;**

**Title tag:** Το title tag είναι ένας από τους πιο βασικούς παράγοντες για SEO. Υποδηλώνει το περιεχόμενο της ιστοσελίδας και δε θα πρέπει να ξεπερνάει τους 66 χαρακτήρες. Αυτοί οι 60 χαρακτήρες θα πρέπει να αποτελούνται από τις λέξεις κλειδιά που θέλουμε να χρησιμοποιήσουμε, με τις πιο σημαντικές λέξεις να βρίσκονται στην αρχή του title. Για να καταλάβουμε πόσο σημαντικό είναι το title tag, αρκεί να δούμε πως η Google είναι το πρώτο που εμφανίζει στα αποτελέσματά της για μια ιστοσελίδα και ακριβώς από κάτω εμφανίζει το description της ιστοσελίδας.

**Keyword tag:** Το keyword tag, είναι ένα metatag στο οποίο βάζουμε keywords τα οποία είναι οι βασικές λέξεις-κλειδιά της ιστοσελίδας μας και είναι ορατά μόνο από τις μηχανές αναζήτησης (όχι από τους επισκέπτες). Τα keywords που μπαίνουν στο keyword-tag χρησιμοποιούνται σαν ranking factor από τις μηχανές αναζήτησης. Παλιότερα, το keyword-tag, ήταν αρκετό για να πετύχουμε υψηλά αποτελέσματα στις μηχανές αναζήτησης. Το 2009, η Google, το Yahoo και το Bink ανακοίνωσαν ότι δε θα χρησιμοποιούν το keyword-tag σαν κριτήριο κατάταξης, οπότε και πλέον έχει χάσει την αξία του.

**Description tag:** Το description tag, περιγράφει με περισσότερη λεπτόμερεια το περιεχόμενο μιας ιστοσελίδας και εμφανίζεται ακριβώς κάτω από το title στα αποτελέσματα της Google. Το description δε θα πρέπει να ξεπερνάει τους 160 χαρακτήρες. Και εδώ θα πρέπει να δοθεί ιδιαίτερη προσοχή στις λέξεις που θα γράψουμε, με τις πιο σημαντικές λέξεις-κλειδιά να πρέπει να γράφονται στην αρχή του description.

**223. Μετά την υλοποίηση της ιστοσελίδας με χρήση CMS τοπικά, ποια βήματα πρέπει να γίνουν ώστε να "ανέβει" (upload) στο Internet;**

Τα βήματα που πρέπει να γίνουν ώστε να ανέβει η CMS ιστοσελίδα μας στο διαδίκτυο είναι:

- Αγορά domain name (όνομα ιστοσελίδας στο διαδίκτυο) και web hosting(υπολογιστής που θα φιλοξενεί την ιστοσελίδα μας).
- Εξαγωγή τοπικής βάσης δεδομένων (συνηθίζεται για εύκολη εξαγωγή το phpmyadmin).
- Ανέβασμα wordpress files στον κατάλληλο φάκελο που βρίσκεται στον public\_html, αν μιλάμε για linux server (για μεταφορά αρχείων μπορεί να χρησιμοποιηθεί ένας FTP client π.χ. Filezilla).
- Δημιουργία MySQL Βάσης στον web hosting μας.
- Εισαγωγή της παλιάς βάσης στη νέα βάση που δημιουργήσαμε.

**224. Τι είναι τα σχεδιαστικά χνάρια και ποια η χρησιμότητά τους;**

Η χρήση σχεδιαστικών χναριών είναι μια αρκετά πρόσφατα υιοθετημένη πρακτική από την επιστήμη της ηλεκτρονικής μάθησης, η οποία αποδεικνύεται ιδιαίτερα αποτελεσματική, ιδιαίτερα σε περιπτώσεις στις οποίες ο σχεδιαστής ενός συστήματος στερείται μεγάλης εμπειρίας, καθώς βοηθά στον αποτελεσματικό σχεδιασμό χωρίς να απαιτείται «εκ νέου ανακάλυψη του τροχού». Η έννοια του σχεδιαστικού χναριού είχε περιγραφεί αρχικά από τον Alexander ως επαναχρησιμοποιούμενες λύσεις για την αντιμετώπιση προβλημάτων αρχιτεκτονικής σχεδίασης τα οποία προκύπτουν με μεγάλη συχνότητα. Συγκεκριμένα, «ένα χνάρι περιγράφει ένα πρόβλημα που προκύπτει κατ' επανάληψη και το οποίο περιγράφει τον πυρήνα της λύσης του προβλήματος αυτού, με τέτοιο τρόπο που η λύση αυτή να εφαρμοστεί πολλαπλές φορές, χωρίς να χρειαστεί να ακολουθηθεί ο ίδιος τρόπος δεύτερη φορά». Η χρήση ενός CMS (Wordpress, Joomla, Drupal) είναι ένα παράδειγμα σχεδιαστικού χναριού, αφού ελαχιστοποιεί τον παράγοντα ανθρώπινου κόπου χάρη στην ευκολία χρήσης του και την προσαρμοστικότητά του.



## ΟΜΑΔΑ Β. ΕΙΔΙΚΕΣ ΕΡΩΤΗΣΕΙΣ

### 1. Περιγράψτε τον τρόπο λειτουργίας των οπτικών δίσκων (CD-DVD-ROM).

Οι **οπτικοί δίσκοι (Compact Disk)** είναι τα τελευταία χρόνια το πιο διαδεδομένο μέσο αποθήκευσης μεγάλου όγκου ψηφιακών δεδομένων, ειδικά για εφαρμογές πολυμέσων. Οι συσκευές ανάγνωσης οπτικών δίσκων βασίζονται στην ανάκλαση μιας φωτεινής δέσμης laser πάνω στην επιφάνεια του δίσκου. Το υλικό κατασκευής τους ποικίλλει ανάλογα με την τεχνολογία ανάγνωσης και εγγραφής. Το κοινό χαρακτηριστικό τους είναι η μεγάλη διάρκεια ζωής τους λόγω του προστατευτικού στρώματος πλαστικού που τους περιβάλλει.

### 2. Τι ονομάζουμε κύρια και τι δευτερεύουσα μνήμη; Ποια είναι η βασική διαφορά τους;

Στην **κύρια ή κεντρική μνήμη (main memory)** του υπολογιστή αποθηκεύονται τα προγράμματα που εκτελεί η Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) και τα δεδομένα για τα προγράμματα αυτά. Η αποθήκευση αυτή είναι προσωρινή, και διαρκεί όσο και η λειτουργία του υπολογιστή. Όταν κλείσει ο υπολογιστής τα δεδομένα αυτά χάνονται. Επειδή η κύρια μνήμη των υπολογιστών είναι έτσι οργανωμένη ώστε να μπορεί να προσπελαστεί άμεσα οποιαδήποτε θέση της, αναφέρεται ως μνήμη τυχαίας προσπέλασης (Random Access Memory) και αποκαλείται RAM. Συμπλήρωμα της κύριας μνήμης είναι η μνήμη ROM (Read Only Memory), στην οποία είναι καταγεγραμμένα μόνιμα από τον κατασκευαστή του υπολογιστή ορισμένα βασικά προγράμματα ή μικρά τμήματα του ΛΣ.

Στη **δευτερεύουσα ή περιφερειακή μνήμη (secondary memory)** τα δεδομένα αποθηκεύονται μόνιμα. Τα δεδομένα που θα καταγραφούν στη δευτερεύουσα μνήμη δε χάνονται όταν κλείσει ο υπολογιστής, και είναι διαθέσιμα την επόμενη φορά που θα λειτουργήσει. Μονάδες δευτερεύουσας μνήμης είναι οι σκληροί δίσκοι, οι εύκαμπτοι δίσκοι, τα CD ROM κλπ.

### 3. Πόσους αγωγούς χρησιμοποιεί ο δίαυλος USB και ποια η χρήση του καθενός απ' αυτούς;

- Κόκκινο = 5 V
- Καφέ = γείωση
- Κίτρινο = Data
- Μπλε = Data

### 4. Τι είναι το υλικό (Hardware) και τι το λογισμικό (Software) στοιχείο ενός υπολογιστικού συστήματος; Να αναφέρετε τέσσερα παραδείγματα από το καθένα.

Στην πληροφορική ως **υλικό (Hardware)** ορίζεται το σύνολο των φυσικών εξαρτημάτων ενός υπολογιστή, όπως π.χ. ηλεκτρικά και ηλεκτρονικά στοιχεία, μικροτσίπ κλπ. Παραδείγματα: Πληκτρολόγιο, Ποντίκι, Μητρική κάρτα, Κεντρική Μονάδα Επεξεργασίας.

Με τον όρο λογισμικό υπολογιστών ή **λογισμικό (Software)** ορίζεται η συλλογή από προγράμματα υπολογιστών, διαδικασίες και οδηγίες χρήσης που εκτελούν ορισμένες εργασίες σε ένα υπολογιστικό



σύστημα Παραδείγματα: επεξεργαστές κειμένου, λειτουργικό σύστημα, πρόγραμμα αναπαραγωγής ήχου, videogames.

##### **5. Ποιες είναι οι βασικές κατηγορίες των λειτουργικών συστημάτων; Ποια είναι τα ιδιαίτερα χαρακτηριστικά καθεμιάς;**

**Λειτουργικό Σύστημα Ομαδικής Επεξεργασίας(Batch Systems Processing):** Το σημαντικότερο χαρακτηριστικό των συστημάτων Ομαδικής Επεξεργασίας είναι ότι δεν υπήρχε αλληλεπίδραση μεταξύ του χρήστη και της εργασίας, ενώ η εργασία βρισκόταν σε εκτέλεση. Η εργασία προετοιμάζεται και υποβάλλεται, ενώ το αποτέλεσμα εμφανίζεται μετά τη μεσολάβηση κάποιου χρονικού διαστήματος (ώρες ή μέρες). Το χρονικό διάστημα μεταξύ της υποβολής της εργασίας της ολοκλήρωσής της ονομάζεται χρόνος περάτωσης ή ανακύκλωσης.

**Λειτουργικό Σύστημα Πολυπρογραμματισμού(Multiprogramming):** Το κύριο χαρακτηριστικό είναι ότι επιτρέπει το διαχωρισμό της κύριας μνήμης σε τμήματα, σε καθένα από τα οποία μπορεί να τοποθετηθεί ένα διαφορετικό πρόγραμμα. Τα προγράμματα αυτά μπορούν να εκτελούνται ταυτόχρονα, αρκεί το καθένα από αυτά να χρησιμοποιεί μια διαφορετική συσκευή του υπολογιστή.

**Λειτουργικό Σύστημα Καταμερισμού Χρόνου(Time Sharing):** Σε αυτό το λειτουργικό σύστημα, στην κύρια μνήμη του υπολογιστή υπάρχει ένα σύνολο προγραμμάτων τα οποία περιμένουν να εξυπηρετηθούν από την Κεντρική Μονάδα Επεξεργασίας. Επιπλέον όμως το Λειτουργικό Σύστημα διαθέτει ένα χρονοδρομολογητή ο οποίος με τη βοήθεια ενός ειδικού κυκλώματος που ονομάζεται χρονιστής (timer), αναλαμβάνει να κατανείμει το χρόνο της ΚΜΕ σε όσα προγράμματα είναι έτοιμα για εκτέλεση. Δηλαδή η ΚΜΕ εξυπηρετεί διαδοχικά όλα τα προγράμματα διαθέτοντας σε καθένα από αυτά ένα ορισμένο ποσό χρόνου. Αυτό το χρονικό διάστημα ονομάζεται κβάντο χρόνου και συνήθως είναι μερικά δέκατα του δευτερολέπτου.

##### **6. Ποιους τύπους οθονών υπολογιστών γνωρίζετε; Περιγράψτε σύντομα τις αρχές λειτουργίας τους.**

Οι οθόνες κατηγοριοποιούνται, σύμφωνα με την τεχνολογία που χρησιμοποιούν σε: **Οθόνες καθοδικού σωλήνα (CRT monitors)** και **Επίπεδες οθόνες (flat-panel displays)**.

**Οθόνες καθοδικού σωλήνα:** Αποτελούνται από ένα (για μονόχρωμες οθόνες) ή τρία (για έγχρωμες οθόνες) πυροβόλα ηλεκτρονίων μέσα σε ένα σωλήνα. Τα ηλεκτρόνια, αφού διανύσουν το σωλήνα, προσκρούουν στο πίσω μέρος της οθόνης που είναι επιστρωμένο με φωσφορίζουσα ουσία. Στο σημείο πρόσκρουσης με την οθόνη, η φωσφορίζουσα ουσία διεγείρεται και λάμπει με αποτέλεσμα την εμφάνιση μιας φωτεινής κουκκίδας. Η κουκκίδα αυτή αποτελεί το ελάχιστο ίχνος απεικόνισης και ονομάζεται εικονοστοιχείο (pixel - picture element). Η διέγερση της φωσφορίζουσας ουσίας διαρκεί μερικά χιλιοστά του δευτερολέπτου, που αποτελούν το χρόνο αναλαμπής.

**Επίπεδη οθόνη τεχνολογία υγρών κρυστάλλων (LCD):** Η λειτουργία τους στηρίζεται στην ύπαρξη ενός πλέγματος από κύτταρα υγρών κρυστάλλων που είναι οργανωμένα σε γραμμές και στήλες. Τα κύτταρα κρυστάλλων ελέγχονται από τρανζίστορ. Όταν διοχετεύεται ρεύμα στα τρανζίστορ, αυτά περιστρέφουν τους κρυστάλλους κατά γωνία ανάλογη της έντασης του ρεύματος. Στο πίσω μέρος της οθόνης υπάρχει μια φθοριούχος πλάκα που ακτινοβολεί φως, το οποίο διέρχεται μέσα από δύο φίλτρα οριζόντιας και κατακόρυφης πόλωσης, στη μέση των οποίων υπάρχει το πλέγμα των κρυστάλλων και το φίλτρο των χρωμάτων. Τα φίλτρα οριζόντιας και κατακόρυφης πόλωσης αφήνουν να περάσει ακτινοβολία συγκεκριμένου μήκους κύματος με οριζόντια ή κατακόρυφη ταλάντωση. Στη συνέχεια το πλέγμα των κρυστάλλων στρέφει την ακτίνα του φωτός ώστε να περάσει ολόκληρη, λίγη ή καθόλου μέσα από το επόμενο φίλτρο, με αποτέλεσμα στην οθόνη να εμφανίζεται το τελικό ίχνος.

**Επίπεδη οθόνη τεχνολογία LED:** Οι LED οθόνες δεν είναι τίποτα περισσότερο από μια παραλλαγή των οθονών LCD. Μάλιστα το πλήρες όνομα τους είναι LED-Backlit LCD. Η μόνη διαφορά μεταξύ των δύο τύπων οθονών είναι πως οι LED χρησιμοποιούν ως πηγή φωτισμού λυχνίες (λάμπες) LED, αντί για λυχνίες φθορισμού.

Οι λάμπες LED είναι πιο φιλικές προς το περιβάλλον, παράγουν περισσότερο φως, ενώ ταυτόχρονα έχουν χαμηλότερες ενεργειακές ανάγκες.

## **7. Με ποιο τρόπο το Λειτουργικό Σύστημα συνεργάζεται με την Κεντρική Μονάδα Επεξεργασίας;**

Ένα πρόγραμμα που εκτελείται στον υπολογιστή συνιστά μία ή περισσότερες διεργασίες. Πρόκειται για το βασικό μέσο εκτέλεσης προγραμμάτων σε ένα συνηθισμένο ΛΣ. Ο ίδιος κώδικας / πρόγραμμα μπορεί να εκτελείται ταυτόχρονα μέσα από πολλές διαφορετικές διεργασίες οι οποίες μπορεί να ανήκουν σε διαφορετικούς χρήστες. Στα πλαίσια της αρχιτεκτονικής φον Νόιμαν και των υπολογιστών που έχουν οικοδομηθεί με βάση αυτήν, μόνο μια διεργασία μπορεί να εκτελείται στην ΚΜΕ (Κεντρική Μονάδα Επεξεργασίας) οποιαδήποτε χρονική στιγμή, επομένως οι ποικίλες διεργασίες οι οποίες είναι ταυτόχρονα ενεργές εκτελούνται ψευδοπαράλληλα, με διαδοχική εναλλαγή του επεξεργαστή μεταξύ τους κάθε λίγη ώρα. Σε ένα παράλληλο σύστημα, όπου υπάρχουν πολλαπλοί επεξεργαστές, ο ταυτοχρονισμός / πολυδιεργασία μπορούν να υλοποιηθούν πραγματικά παράλληλα. Σε κάθε περίπτωση, το τμήμα του πυρήνα το οποίο λαμβάνει διάφορες αποφάσεις σχετικά με την ανάθεση των ΚΜΕ στις διάφορες διεργασίες ονομάζεται χρονοπρογραμματιστής.

## **8. Τι πρέπει να κάνουμε, έτσι ώστε να είναι δυνατόν να γράψουμε ένα μικρό κείμενο μπροστά ακριβώς από μια εικόνα που έχουμε εισάγει στο κείμενο;**

Πατάμε δεξί κλικ πάνω στην εικόνα και επιλέγουμε “αναδίπλωση κειμένου→πίσω από το κείμενο”. Κάνουμε κλικ στα αριστερά της εικόνας έξω από το περίγραμμά της και μόλις εμφανιστεί ο δείκτης θέσεως γράφουμε το κείμενο που θέλουμε. Αυτό εμφανίζεται πάνω στην εικόνα.

9. Να περιγράψετε τον κύκλο ανάπτυξης ενός Πληροφοριακού Συστήματος. Δώστε ένα παράδειγμα.

**Κύκλος ανάπτυξης ενός Πληροφοριακού Συστήματος:**

- Διερευνητική μελέτη
- Μελέτη σκοπιμότητας
- Ανάλυση απαιτήσεων
- Έλεγχος
- Υλοποίηση-κωδικοποίηση
- Σχεδιασμός
- Παράδοση
- Εγκατάσταση-Λειτουργία-Συντήρηση, Απόσυρση.

π.χ. Ένα σύγχρονο πληροφοριακό σύστημα Data Warehouse (Αποθήκευσης δεδομένων).

10. Με τη χρήση κατάλληλου προγράμματος να παρουσιάσετε την διαφορά μεταξύ των εντολών break και continue.

Η εντολή break διακόπτει τον βρόχο αμέσως όταν συναντάται, ενώ η εντολή continue παραλείπει ορισμένες δηλώσεις μέσα στο βρόχο.

Για παράδειγμα εάν θέλω να παρουσιάσω τον υπολογισμό αθροίσματος 10 θετικών ακεραίων θα χρησιμοποιήσουμε break ως εξής:

```
# include <stdio.h>
int main(){
    int i;
    double number, sum = 0.0;
    for(i=1; i <= 10; ++i) {
        printf("Enter a n%d: ",i);
        scanf("%lf",&number);
        if(number < 0.0){
            break;
        }
        sum += number;
    }
    printf("Sum = %.2lf",sum);
    return 0;
}
```

Από την άλλη αν δεν είμαστε αυστηροί ως προς την είσοδο και μπορούμε να παραλήψουμε απλά τους αρνητικούς αριθμούς, θα χρησιμοποιήσουμε το continue ως εξής:

```
# include <stdio.h>
int main(){
    int i;
    double number, sum = 0.0;
    for(i=1; i <= 10; ++i) {
        printf("Enter a n%d: ",i);
        scanf("%lf",&number);
        if(number < 0.0){
            continue;
        }
        sum += number;
    }
    printf("Sum = %.2lf",sum);
    return 0;
```

- 11. Να γραφεί πρόγραμμα σε C, που να υπολογίζει τη μέση τιμή ακέραιων αριθμών, όταν εισάγονται από το πληκτρολόγιο. Η εισαγωγή τερματίζεται, όταν δοθεί σαν τιμή ακεραίου το 0 (μηδέν).**

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

int main(int argc, char *argv[]) {

    int number=0;
    int value=0;
    double mean=0;
    int counter;
    printf("\nGive number: ");
    scanf("%d",&number);
    value=value+number;
    counter=1;
    mean=value/counter;;
    printf("\nThe mean is: %lf", mean);
    while(number!=0){
        printf("\nGive number: ");
        scanf("%d",&number);
        value=value+number;
        counter++;
        mean=value/counter;
        printf("The mean is: %lf", mean);
    }
    printf("\ncomputation terminated ...");
    return 0;
}
```

12. Να γραφεί πρόγραμμα σε C, που θα διαβάσει 2 ακεραίους αριθμούς από το πληκτρολόγιο και θα εκτυπώνει το άθροισμά τους στην οθόνη σε δεκαδική μορφή. Το αποτέλεσμα να φαίνεται επίσης σε οκταδική και δεκαεξαδική μορφή.

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

int main(int argc, char *argv[]) {
    int x,y,sum;

    printf("give x: ");
    scanf("%d",&x);
    printf("give y: ");
    scanf("%d",&y);
    sum=x+y;
    printf("%X\n", sum);
    printf("%o\n", sum);
    return 0;
}
```

13. Να γραφεί πρόγραμμα σε C, που θα υπολογίζει το μέσο όρο δύο πραγματικών αριθμών. Ο υπολογισμός της μέσης τιμής να γίνει με τη δημιουργία ξεχωριστής συνάρτησης.

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

double mean(int, int);

int main(int argc, char *argv[]) {
    int x,y;
    double result;
    printf("give x: ");
    scanf("%d",&x);
    printf("give y: ");
    scanf("%d",&y);
    result=mean(x,y);
    printf("the mean value is: %lf",result);
    return 0;
}

double mean(int a, int b){
    int sum;
    double mean_value;
    sum=a+b;
    mean_value=(double) (sum/2);
}
```

```

        return mean_value;
    }

```

- 14. Να γραφεί πρόγραμμα σε C, που θα διαβάζει το όνομα του χρήστη και θα εκτυπώνει φιλικό χαιρετισμό.**

```

#include <stdio.h>
#include <stdlib.h>

int main(){
    char array[100];
    printf("Enter your name: ");
    scanf("%s", array);

    printf("\nHello %s, nice to meet you !!! \n",array);
    return 0;
}

```

- 15. Να γραφεί πρόγραμμα σε C, που θα εκτυπώνει πίνακα αντιστοίχισης μονάδων θερμοκρασίας από Φαρενάιτ σε Κελσίου ανά 20 βαθμούς. [Ο τύπος μετατροπής είναι  $C=5(F-32)/9$  ]**

```

#include<stdio.h>
#include <stdlib.h>

int main() {
    double celsius;
    int fahrenheit;
    int i;

    for(i=0;i<1000;i=i+20){
        fahrenheit=i;
        celsius=5*(fahrenheit-32)/9 ;
        printf("\n temp in Fahrenheit : %d ---> ",fahrenheit);
        printf("temp in celsius : %.2f ", celsius);
    }

    return (0);
}

```

- 16. Να σχεδιαστεί ο αλγόριθμος εξαγωγής στοιχείου από λίστα.**

```

Algorithm Deletion
1.If (DATA(list)='VAL') then{
    Ptr=LIST;
    LIST=NEXT(list);
    Delete ptr;
    Stop;
}

```

```
Back=list;
Ptr=list;
2.while(ptr<>NULL) repeat step 3 to 5
3.    If (DATA(ptr)='VAL') then{
        NEXT(back)=NEXT(ptr);
        Delete ptr;
        Exit;
    }
4.    back=ptr;
5.    ptr=next(ptr);
6.[end of while loop]
7.END
```

**17. Να σχεδιαστεί ο αλγόριθμος εισαγωγής στοιχείου σε λίστα.**

Algorithm Insertion

```
1.X=new node;
2.Read(DATA(X));
3.If (FIRST=NULL) then{
    First=X;
    NEXT(X)=NULL;
}
Else{
    NEXT(X)=First;
    First=X;
}
4.END
```

**18. Να σχεδιαστεί ο αλγόριθμος εισαγωγής στοιχείου σε ουρά.**

Algorithm Enqueue

```
1. If ( REAR = size ) then //Queue is full
2.    print "Queue is full"
3.    Exit
4. Else
5.    If ( FRONT = 0 ) and ( REAR = 0 ) then //Queue is empty
6.        FRONT = 1
7.    End if
8.    REAR = REAR + 1 // increment REAR
9.    Que[ REAR ] = ITEM
10. End if
11. Stop
```



**19. Να σχεδιαστεί ο αλγόριθμος εξαγωγής στοιχείου από ουρά.**

```
Algorithm Dequeue
1.  If ( FRONT = 0 ) then
2.      print "Queue is empty"
3.      Exit
4.  Else
5.      ITEM = Que [ FRONT ]
6.      If ( FRONT = REAR )
7.          REAR = 0
8.          FRONT = 0
9.      Else
10.         FRONT = FRONT + 1
11.     End if
12. End if
13. Stop
```

**20. Να σχεδιαστεί ο αλγόριθμος εισαγωγής στοιχείου σε στοίβα.**

```
Algorithm Push
1. If TOP >= SIZE - 1 then
    Write "Stack is Overflow"
2. TOP = TOP + 1
3. STACK [TOP] = X
```

**21. Να σχεδιαστεί ο αλγόριθμος εξαγωγής στοιχείου από στοίβα.**

```
Algorithm Pop
1. If TOP = -1 then
    Write "Stack is Underflow"
2. Return STACK [TOP]
3. TOP = TOP - 1
```

**22. Να σχεδιαστεί ο αλγόριθμος της σειριακής αναζήτησης.**

```
Algorithm Linear Search
Input ( Array A, Value x)

1. Set i to 1
2. if i > n then go to step 7
3. if A[i] = x then go to step 6
4. Set i to i + 1
5. Go to Step 2
6. Print Element x Found at index i and go to step 8
7. Print element not found
8. Exit
```

**23. Να σχεδιαστεί ο αλγόριθμος της δυαδικής αναζήτησης.**

```
Algorithm binary_search
1. A ← sorted array
2. n ← size of array
3. x ← value to be searched
4.   Set lowerBound = 1
6.   Set upperBound = n
7.   while x not found
8.     if upperBound < lowerBound
9.       EXIT: x does not exists.
10.    set midPoint = lowerBound + ( upperBound - lowerBound ) / 2
11.    if A[midPoint] < x
12.      set lowerBound = midPoint + 1
13.    if A[midPoint] > x
14.      set upperBound = midPoint - 1
15.    if A[midPoint] = x
16.      EXIT: x found at location midPoint
17.    end while
```

**24. Να σχεδιαστεί ο αλγόριθμος της ταξινόμησης φυσαλίδας.**

```
Algorithm BubbleSort
Input (list)
1.for all elements of list
2.  if list[i] > list[i+1]
3.    swap(list[i], list[i+1])
    end if
  end for
4.  return list

end BubbleSort
```

**25. Τι είναι ένα σύστημα Πολυδιεργασίας (Multitasking System);**

Το multitasking σε ένα λειτουργικό σύστημα επιτρέπει την εκτέλεση περισσότερων από μία εργασιών κάθε φορά (ταυτόχρονα). Το λειτουργικό σύστημα είναι σε θέση να παρακολουθεί τη κατάσταση αυτών των εργασιών (tasks) και να μεταφέρει τον έλεγχο από το ένα στο άλλο χωρίς να χάνουμε πληροφορίες. Τα Microsoft Windows 2000, το OS / 390 και το Linux της IBM είναι παραδείγματα λειτουργικών συστημάτων που μπορούν να κάνουν multitasking (σχεδόν όλα τα σημερινά λειτουργικά συστήματα μπορούν). Όταν για παράδειγμα, ανοίγουμε το πρόγραμμα περιήγησης και, στη συνέχεια, μια άλλη εφαρμογή ταυτόχρονα, προκαλείτε το λειτουργικό σύστημα να κάνει πολλαπλές εργασίες. Κάθε διεργασία καταναλώνει αποθηκευτικό σύστημα και άλλους πόρους, και ως εκ τούτου ο αριθμός των παράλληλων διεργασιών δεν είναι απεριόριστος. Καθώς ξεκινούν περισσότερες εργασίες, το σύστημα μπορεί να επιβραδυνθεί ή να ξεκινήσει να εξαντλείται από κοινόχρηστο αποθηκευτικό χώρο.

**26. Τι εννοούμε με τον όρο διεργασία; Σε τι διαφέρει από το πρόγραμμα;**

**Διεργασία (process)** είναι ένας όρος της πληροφορικής ο οποίος περιγράφει το στιγμιότυπο ενός προγράμματος που εκτελείται σε έναν υπολογιστή. Σε αντιδιαστολή με την έννοια του προγράμματος, το οποίο είναι ένα στατικό σύνολο εντολών, μια διεργασία συνιστά την εκτέλεση αυτών των εντολών. Επομένως ένα πρόγραμμα γενικώς συσχετίζεται με περισσότερες από μία διεργασίες, μία για κάθε φορά που εκτελείται. Μια διεργασία αποτελείται από το ίδιο το πρόγραμμα και από κάποιες τιμές που περιέχονται στη μνήμη και στους καταχωρητές του επεξεργαστή, δηλαδή την κατάσταση του συστήματος, κάθε στιγμή που το πρόγραμμα εκτελείται.

**27. Πως λειτουργεί η σελιδοποίηση; Τι συμβαίνει όταν μια σελίδα της εικονικής μνήμης δεν αντιστοιχεί σε ενότητα της φυσικής μνήμης;**

**Σελιδοποίηση:** Ο χώρος των ιδεατών διευθύνσεων διαιρείται σε σελίδες ίσου μεγέθους και παρόμοια η κύρια μνήμη διαιρείται σε πλαίσια σελίδων ίδιου μεγέθους. Τα πλαίσια σελίδων μοιράζονται στις διεργασίες που βρίσκονται εκείνη τη στιγμή στο σύστημα, έτσι ώστε σε οποιαδήποτε χρονική στιγμή μια δεδομένη διεργασία να έχει λίγες σελίδες εγκατεστημένες στην κύρια μνήμη, ενώ οι υπόλοιπες να βρίσκονται στη βοηθητική μνήμη. Όταν μια σελίδα της εικονικής μνήμης δεν αντιστοιχεί σε ενότητα της φυσικής μνήμης, σημαίνει ότι η σελίδα αυτή βρίσκεται στη δευτερεύουσα μνήμη.

**28. Τι λέγεται εικονική μηχανή (virtual machine);**

Η **εικονική μηχανή (virtual machine)** είναι μια εξομοίωση ενός ξεχωριστού υπολογιστή μέσα στο PC μας. Έχει ακριβώς τις λειτουργίες ενός φυσικού υπολογιστή: κάνει boot, επανεκκίνηση, και τερματισμό λειτουργίας. Μπορούμε να συνδέσουμε περιφερειακές συσκευές, από φλασάκια USB μέχρι εκτυπωτές και scanner. Η virtual machine διαθέτει μέχρι και δικό της BIOS ή UEFI, ενώ χρειάζεται drivers για την εξομοίωση της κάρτας γραφικών και ήχου.

**29. Πόσα είδη διεργασιών υπάρχουν; Σε ποιες καταστάσεις μπορεί να βρίσκεται μια διεργασία;**

Με τον όρο **διεργασία (process)** νοείται ένα πρόγραμμα το οποίο έχει φορτωθεί στην κύρια μνήμη και βρίσκεται σε κατάσταση εκτέλεσης με αποτέλεσμα να καταναλώνει χρόνο της ΚΜΕ και πόρους του συστήματος(κύρια μνήμη,χώρο σε αποθηκευτικά μέσα,κανάλια επικοινωνίας). Η διεργασία δηλαδή εκφράζει κάτι δυναμικό και σε εξέλιξη, σε αντίθεση με το πρόγραμμα που είναι κάτι στατικό. Κάθε ΛΣ αποτελείται από πολλά επί μέρους προγράμματα,τα οποία εκτελούνται παράλληλα για να εξυπηρετούν τους χρήστες, είναι δηλαδή και αυτό χωρισμένο σε διεργασίες. Το ίδιο μπορεί να γίνει και με τα προγράμματα των χρηστών, να διαιρεθούν δηλαδή σε τμήματα τα οποία εκτελούνται παράλληλα. Όταν όμως τα τμήματα αυτά πρέπει να μοιράζονται διάφορα στοιχεία του προγράμματος, όπως π.χ. μεταβλητές, μπορεί να χρησιμοποιηθεί και ένα εναλλακτικό είδος διεργασιών, οι ελαφρές διεργασίες (lightweight processes) ή νήματα εκτέλεσης (threads of execution) ή απλούστερα νήματα(threads). Όταν ένα πρόγραμμα διαιρεθεί σε νήματα που εκτελούνται «παράλληλα» (όπως οι

διεργασίες) αντί σε ανεξάρτητες διεργασίες, προκαλείται μικρότερη επιβάρυνση στο υπολογιστικό σύστημα, γιατί η μεταγωγή περιβάλλοντος μεταξύ νημάτων είναι πιο γρήγορη από ό,τι μεταξύ διεργασιών.

**Οι καταστάσεις** που μπορεί μια διεργασία να βρεθεί:

- Νέα (New): Η διεργασία δημιουργείται.
- Εκτελούμενη (Running): Εκτελούνται εντολές.
- Εν αναμονή (Waiting): Η διεργασία αναμένει να συμβεί κάποιο γεγονός (όπως η λήψη ενός σήματος).
- Έτοιμη (Ready): Η διεργασία περιμένει να ανατεθεί σε έναν επεξεργαστή.
- Τερματισμένη (Terminated): Η εκτέλεση της διεργασίας έχει ολοκληρωθεί.

### 30. Τι ονομάζεται εσωτερικός και εξωτερικός κατακερματισμός;

Λόγω των κανόνων που διέπουν την κατανομή μνήμης, η μνήμη μπορεί να παραχθεί μόνο σε «τεμάχια» 4,8,16 bytes. Έτσι αν κάποιο πρόγραμμα ζητήσει 23 bytes θα πάρει 32 bytes. Άρα ουσιαστικά μένουν 9 bytes ανεκμετάλλευτα και αυτό είναι μία περίπτωση εσωτερικού κατακερματισμού.

**Εσωτερικός κατακερματισμός:** είναι η μνήμη που δεν χρησιμοποιείται (δαπανάται) και είναι ορατή μόνον από τη διεργασία που ζητά μνήμη. Συμβαίνει επειδή η ποσότητα μνήμης που θα εκχωρηθεί στη διεργασία πρέπει να είναι μεγαλύτερη ή ίση από την αιτούμενη ποσότητα.

Ο **εξωτερικός κατακερματισμός** δημιουργείται όταν η ελεύθερη μνήμη χωρίζεται σε μικρά μπλοκ και διασκορπίζεται από την κατανομημένη μνήμη. Είναι μια αδυναμία ορισμένων αλγορίθμων κατανομής αποθήκευσης, όταν αποτυγχάνουν να δεσμεύσουν αποτελεσματικά τη μνήμη που χρησιμοποιούν τα προγράμματα. Το αποτέλεσμα είναι ότι, παρόλο που είναι διαθέσιμος ο ελεύθερος χώρος αποθήκευσης, είναι πραγματικά αχρησιμοποίητος επειδή χωρίζεται σε κομμάτια που είναι πολύ μικρά ξεχωριστά για να ικανοποιήσουν τις απαιτήσεις της εφαρμογής.

### 31. Τι συμβαίνει κατά την ανταλλαγή μνήμης;

Όταν ένας υπολογιστής εξυπηρετεί ένα μεγάλο αριθμό διεργασιών, τα προγράμματα και τα δεδομένα όλων δε χωρούν στην κύρια μνήμη. Τότε επιστρατεύεται η δευτερεύουσα μνήμη για να βοηθήσει: τα προγράμματα και τα δεδομένα ορισμένων διεργασιών κρατούνται στη δευτερεύουσα μνήμη, και κάθε φορά που είναι η σειρά μιας τέτοιας διεργασίας να εκτελεστεί, τότε μόνο φορτώνονται στην κύρια μνήμη. Για να απελευθερωθεί όμως χώρος στην κύρια μνήμη για αυτά, πρέπει κάποια άλλη διεργασία να μεταφερθεί με τη σειρά της στη δευτερεύουσα μνήμη.

**32. Να ορισθούν οι έννοιες "Μερισμός χρόνου" (Time sharing) και "επεξεργασία Πραγματικού χρόνου" (Real Time processing).**

Ο “μερισμός χρόνου” (time sharing) είναι η ανταλλαγή ενός υπολογιστικού πόρου μεταξύ πολλών χρηστών μέσω του πολλαπλού προγραμματισμού και της πολλαπλής εκτέλεσης εργασιών ταυτόχρονα. Με τον όρο “επεξεργασία Πραγματικού χρόνου” (Real Time processing), εννοούμε την επεξεργασία δεδομένων που εμφανίζεται ή λαμβάνει χώρα αμέσως, κατά την εισαγωγή δεδομένων ή την παραλαβή μιας εντολής.

**33. Τι είναι τα μονότροπα και τι τα πολύτροπα καλώδια οπτικών ινών;**

Στα μονότροπα καλώδια, η διάμετρος της κεντρικής ίνας είναι πολύ μικρή και πλησιάζει το επίπεδο του μήκους κύματος του εκπεμπόμενου σήματος. Έχουμε ένα τρόπο μετάδοσης του σήματος, τον αξονικό.

Στα πολύτροπα, οι διάφορες ακτίνες του οπτικού σήματος ταξιδεύουν ανακλώμενες υπό διαφορετικές γωνίες (ανάλογα με την είσοδο τους) και έτσι έχουμε πολλούς δρόμους μετάδοσης.(Οι οπτικές ίνες χωρίζονται σε διακριτού βήματος και βαθμιαίου βήματος.)

**34. Περιγράψτε την λειτουργία του δρομολογητή (router) και του μεταγωγέα (ethernet Switch).**

**ΜΕΤΑΓΩΓΕΙΣ (SWITCHES)**

Ο μεταγωγέας αποτελεί ένα συνδυασμό της γέφυρας και του επαναλήπτη. Χρησιμοποιείται για την μείωση της κίνησης του δικτύου. Δουλεύουν με την έννοια , εξέταση ενός μέρος της πληροφορίας και μετά αποστολή (cut-through). Λειτουργεί σε 2 επίπεδα : στο φυσικό επίπεδο (MAC), και στο λογικό επίπεδο (LLC)ανάλογα με τον προορισμό της πληροφορίας.

**ΔΡΟΜΟΛΟΓΗΤΕΣ (ROUTERS)**

Η κύρια λειτουργία τους είναι η διασύνδεση των δικτύων στο 3ο επίπεδο του OSI. Συνδέουν τα δίκτυα μεταξύ τους βάσει το πρωτόκολλο που χρησιμοποιούν για την επικοινωνία τους. Οι δρομολογητές κάνουν 2 βασικές διαδικασίες :

A) τον καθορισμό της καλύτερης διαδρομής

B) τη μεταφορά των πακέτων

Η δρομολόγηση των πακέτων γίνεται βάσει της λογικής διεύθυνσης (LLC).

**35. Τι είναι η οπτική ίνα; Να περιγράψετε την λειτουργία της.**

Η οπτική ίνα χρησιμοποιείται όταν οι απαιτήσεις σε ρυθμούς μετάδοσης είναι αρκετά αυξημένες και όταν οι αποστάσεις είναι πολύ μεγάλες. Στο κέντρο του καλωδίου υπάρχει η οπτική ίνα που κατασκευάζεται από γυαλί, που θα μεταφέρει τη δέσμη φωτός (που συνήθως εκπέμπεται από LED , LASER) με πολύ λίγες απώλειες. Την οπτική ίνα την περιβάλλει ειδικό υλικό με δείκτη διάθλασης μικρότερο από αυτόν της οπτικής ίνας (cladding). Το υλικό αυτό βοηθά να αντανakλάται συνέχει η δέσμη φωτός που θα πέσει στην οπτική ίνα. Εξωτερικά υπάρχουν δέσμες συνθετικών ινών που προστατεύουν την ίνα και όλα αυτά περικλείονται από εξωτερικό πλαστικό περίβλημα. (Το φως εκπέμπεται από LED και διαθλάται κατά μήκος της οπτικής ίνας με την βοήθεια του εξωτερικού περιβλήματος).

**36. Τι είναι ημίδιπλη (ταυτόχρονα αμφίδρομη - Full Duplex) επικοινωνία και τι η ημίδιπλη (αμφίδρομη μη ταυτόχρονη - Half Duplex) επικοινωνία. Να αναφέρετε τις διαφορές μεταξύ τους.**

HALF DUPLEX (Αμφίδρομη μη Ταυτόχρονη Επικοινωνία). Η αποστολή δεδομένων γίνεται είτε προς τη μία ,είτε προς την άλλη μεριά όχι όμως ταυτόχρονα.

FULL DUPLEX (Ταυτόχρονα Αμφίδρομη Επικοινωνία). Τα δεδομένα μεταδίδονται ταυτόχρονα και προς τις δύο κατευθύνσεις.

Είναι προφανές, πως στην Ταυτόχρονα Αμφίδρομη Επικοινωνία, **δεν υπάρχει χρονική καθυστέρηση** για την αλλαγή της κατεύθυνσης ροής των πληροφοριών, όπως συμβαίνει στην Αμφίδρομη μη Ταυτόχρονη Επικοινωνία.

**37. Τι είναι το X.25; Να περιγράψετε τα επίπεδα του.**

Το X.25 είναι ακριβώς ένα πρωτόκολλο επικοινωνίας το οποίο ορίζει ένα σύνολο καθορισμών / προδιαγραφών για το DTE / DCE interface. Τα βασικά του καθήκοντα είναι τα εξής :

- 1) Ανταλλαγή κωδίκων για αποκατάσταση μιας σύνδεσης (Link Set-Up).
  - 2) Αλληλουχία λειτουργιών που αφορούν στη μετάδοση / λήψη δεδομένων, στη διαχείριση διαδικασιών ελέγχου σφαλμάτων, στην εξασφάλιση έγκυρης και σωστής λήψης πληροφοριών.
- Τυπικό Δίκτυο X.25 : Ένα X.25 δίκτυο, αποτελείται από επικοινωνιακούς κόμβους μεταγωγής πακέτων, συνδεδεμένων ανά δύο point – to – point (Connected Mesh Topology). Υπάρχει επομένως ένας τουλάχιστον φυσικός δρόμος επικοινωνίας μεταξύ οποιονδήποτε δύο κόμβων του δικτύου.

- Φυσικό επίπεδο:

Το Φυσικό Επίπεδο (Επίπεδο-1) λέγεται και Επίπεδο bits. Εδώ μας ενδιαφέρει η μεταφορά bits πληροφορίας, με όσο αξιόπιστο τρόπο γίνεται. Το επίπεδο αυτό ορίζει συστάσεις για τα ηλεκτρικά χαρακτηριστικά (επιτρεπόμενες ηλεκτρικές τάσεις, ρυθμοί μετάδοσης, κ.λ.π.), τα μηχανικά χαρακτηριστικά (χρησιμοποιούμενα καλώδια, connectors, κ.λ.π.), τα λειτουργικά και διαδικαστικά χαρακτηριστικά που απαιτούνται για μια φυσική DTE/DCE. Για μια τέτοια σύνδεση δεν καθορίζεται το μέσο, αλλά συνήθως είναι πάνω από μια αφιερωμένη γραμμή ενός επιλεγόμενου τηλεφωνικού δικτύου (PSTN) με modem στις δύο άκρες του, προκειμένου να συνδεθούμε στο Δημόσιο Δίκτυο Δεδομένων (PDN - Public Data Network). Οι ρυθμοί μετάδοσης δεδομένων μπορούν να φθάσουν έως και 48.000 bps.

- Data Link Layer (Επίπεδο 2):

Σκοπός του επιπέδου αυτού είναι η αξιόπιστη, χωρίς σφάλματα μεταφορά δεδομένων μεταξύ των δύο άκρων ενός λογικού κυκλώματος. Ενώ στο επίπεδο-1 αποκαθίσταται η φυσική σύνδεση DTE/DCE και αποστέλλονται απλά bits, στο επίπεδο-2 γίνεται μια πρώτη οργάνωση και απόδοση έννοιας στα bits του επιπέδου-1. Τώρα στέλνονται πλαίσια (frames) και μάλιστα με αξιόπιστο τρόπο. Δύο είναι τα βασικά χαρακτηριστικά του επιπέδου αυτού: Υπάρχει επιβεβαίωση σωστής λήψης από το δίκτυο. Η επικοινωνία DTE/DCE είναι σύγχρονη, point-to-point και Full Duplex (ταυτόχρονης διπλής κατεύθυνσης).

- Network Layer (Επίπεδο 3)

(packet – layer protocol) Το πρωτόκολλο αυτό περιγράφει την ανταλλαγή πακέτων (packets) μεταξύ DTE και DCE. Οι βασικές λειτουργίες που ορίζονται είναι οι εξής:

- Δημιουργία πακέτων ελέγχου και μεταφοράς δεδομένων.
- Διαδικασίες ανταλλαγής των πακέτων αυτών μεταξύ DTE/DCE.



- Δημιουργία και εποπτεία Νοητών Κυκλωμάτων. Ένα DTE μπορεί να δημιουργήσει ταυτόχρονα

περισσότερα του ενός νοητά κυκλώματα PVC ή SVC προς διάφορους συνδρομητές (DTEs).

### 38. Τι γνωρίζετε για την τεχνολογία ADSL;

Asymmetric Digital Subscriber Line (Ασύμμετρη Ψηφιακή Συνδρομητική Γραμμή) ή ADSL είναι μια μορφή DSL, δηλαδή μια τεχνολογία μετάδοσης δεδομένων που λειτουργεί πάνω σε παραδοσιακή τηλεφωνική γραμμή αλλά πετυχαίνει υψηλότερους ρυθμούς μεταφοράς από τα παραδοσιακά modem. Το απλό χάλκινο καλώδιο (γνωστό και ως τοπικός βρόχος, local loop ή last mile) που συνδέει σχεδόν κάθε σπίτι με το τοπικό τηλεφωνικό κέντρο, έχει πολύ περισσότερες δυνατότητες από την υποστήριξη της απλής τηλεφωνίας. Έτσι με χρήση ανώτερου τμήματος του εύρους ζώνης του βρόχου, εκείνου το οποίο μένει αναξιοποίητο από την κλασική τηλεφωνία (PSTN ή ISDN), επιτυγχάνονται υψηλές ταχύτητες μετάδοσης δεδομένων. Το γεγονός αυτό προσφέρει κι ένα ακόμη πλεονέκτημα: η παραδοσιακή τηλεφωνία και η μετάδοση δεδομένων μπορούν να λειτουργούν ταυτόχρονα και ανεξάρτητα η μία από την άλλη, εφόσον χρησιμοποιούν διαφορετικό φάσμα συχνοτήτων στην τηλεφωνική γραμμή. Ωστόσο οι συχνότητες που χρησιμοποιεί το ADSL εξασθενούν συντομότερα από αυτές της τηλεφωνίας, με αποτέλεσμα να μπορεί να λειτουργήσει σε αποστάσεις έως 5 Χλμ. από το τηλεφωνικό κέντρο. Επιπλέον, όσο μεγαλώνει η απόσταση από το τηλεφωνικό κέντρο τόσο μειώνεται η ταχύτητα μετάδοσης δεδομένων που μπορεί να επιτευχθεί από το ADSL.

Χαρακτηριστικό του ADSL είναι το ότι οι ταχύτητες λήψης και αποστολής δεδομένων διαφέρουν - σε αυτό οφείλει και τη λέξη «ασύμμετρη» στο όνομά του. Η μέγιστη ταχύτητα που μπορεί να επιτύχει είναι τα 24/1 Mbps. Ένα επιπλέον χαρακτηριστικό είναι ότι η σύνδεση ADSL είναι μόνιμη και διαθέσιμη ανά πάσα στιγμή (always-on). Δηλαδή δεν απαιτείται σύνδεση και αποσύνδεση από το δίκτυο όπως συμβαίνει με τις τηλεφωνικές κλήσεις.

### 39. Περιγράψτε την δομή ενός δικτύου ATM καθώς και τις κατηγορίες υπηρεσιών που προσφέρει.

Ο ασύγχρονος τρόπος μεταφοράς (Asynchronous Transfer Mode, ATM) είναι σύγχρονη και πολλά υποσχόμενη εφαρμογή της τεχνικής της μεταγωγής. Συνδυάζει την αποδοτικότητα της μεταγωγής πακέτων με την αξιοπιστία της μεταγωγής κυκλώματος. Για τη μετάδοση των δεδομένων, χρησιμοποιεί σταθερού μεγέθους πακέτα των 53 bytes, τις κυψέλες (cells). Από αυτά, τα 5 πρώτα bytes αποτελούν της ATM επικεφαλίδα (header) και τα υπόλοιπα 48 bytes την ωφέλιμη πληροφορία του χρήστη (payload). Το γεγονός ότι χρησιμοποιούνται κυψέλες σταθερού μεγέθους, επιβαρύνει πολύ λιγότερο τις διεργασίες μεταγωγής και δρομολόγησης που εκτελούνται σε κάθε κόμβο του δικτύου ATM. Έτσι, μπορούν να επιτευχθούν πολύ υψηλές ταχύτητες μεταγωγής των δεδομένων, που μπορούν να φθάσουν και στα 622 Mbps.

Το μοντέλο ATM αποτελείται από 3 επίπεδα:

**Φυσικό επίπεδο:** ασχολείται με το φυσικό μέσο (τάση, ρυθμός μετάδοσης, κτλ.) Το ATM έχει σχεδιαστεί για να είναι ανεξάρτητο από το μέσο μετάδοσης, απλά αναφέρεται ότι η μετάδοση πρέπει να γίνεται με ομοαξονικά καλώδια, οπτικές ίνες και γενικά μέσα μετάδοσης ευρείας ζώνης.

**Το επίπεδο ATM:** ασχολείται με τη μετάδοση των πακέτων (ονομάζονται κελιά) και καθορίζει τα χαρακτηριστικά και τα περιεχόμενά τους. Ασχολείται με την εγκατάσταση και αναγνώριση κυκλωμάτων καθώς και με τον έλεγχο της συμφόρησης.

**Επίπεδο προσαρμογής ATM (ATM Adaptation layer):** επειδή οι περισσότερες εφαρμογές δεν εργάζονται με κελιά το επίπεδο προσαρμογής ATM επιτρέπει την αποστολή πακέτων με μέγεθος μεγαλύτερο του ενός κελιού. Τα πακέτα αυτά χωρίζονται σε μικρότερα κομμάτια ενός κελιού και μεταδίδονται ενώ στο δέκτη επανασυναρμολογούνται δίνοντας το αρχικό πακέτο.



#### 40. Τι είναι το PSTN και το ISDN; Να αναφέρετε τις λειτουργίες τους.

Το PSTN είναι ένα δίκτυο για μετάδοση φωνής και δεδομένων. Τα βασικά στοιχεία του είναι:

- Τηλεφωνικές συσκευές των συνδρομητών.
- Τα τηλεφωνικά κέντρα.
- Το ζευκτικό δίκτυο.

Η λειτουργία του PSTN είναι αυτή του δικτύου μεταγωγής κυκλώματος και έχει 3 φάσεις:

α' φάση -

Αποκατάσταση κυκλώματος

β' φάση - Μεταφορά πληροφορίας

γ' φάση - Τερματισμός κυκλώματος

Το ISDN επιτρέπει στους χρήστες να μεταδίδουν φωνή, εικόνα κα δεδομένα σε ψηφιακή μορφή επάνω σε ένα δισύρματο καλώδιο. Έχει τα εξής βασικά στοιχεία :

1. Ψηφιακή Μετάδοση
2. Υπάρχει Σηματοδότηση
3. Πολλαπλός Σκοπός Διασύνδεσης.

#### 41. Ποια είναι τα βασικά στάδια κατά την επεξεργασία εικόνων bitmap;

α) **Ανάλυση (resolution).** Εκφράζει την πυκνότητα των εικονοστοιχείων της εικόνας σε κάθε διάσταση και μετρείται σε σημεία ανά ίντσα (dots per inch, ενός αρχείου εικόνας είναι ανάλογο του τετραγώνου της ανάλυσης της. Θα πρέπει να σημειωθεί, ότι η ανάλυση της εικόνας είναι διαφορετική από την ανάλυση της συσκευής που χρησιμοποιείται για την παρουσίασή της. Ενδεικτικά αναφέρουμε ότι μια εικόνα στην οθόνη του υπολογιστή έχει συνήθως ανάλυση 72 dpi, σε έναν κοινό εκτυπωτή 600 dpi, ενώ για επαγγελματικές εκτυπώσεις απαιτείται ανάλυση της τάξης των 3000 dpi.

β) **Χρωματικό βάθος.** Το χρωματικό βάθος σχετίζεται με τον αριθμό των χρωμάτων που είναι διαθέσιμα για τη δημιουργία μιας εικόνας. Καθορίζεται από τον αριθμό των bit που χρησιμοποιούνται για τη χρωματική περιγραφή ενός pixel. Έτσι μια εικόνα με χρωματικό βάθος 8 bits αποδίδει  $2^8=256$  χρώματα, ενώ με 16 bit αποδίδονται  $2^{16}=65536$  χρώματα. Για τέλεια χρωματική απόδοση πραγματικού χρώματος (true colour) απαιτούνται 24 bit, οπότε η εικόνα διαθέτει περίπου 16,8 εκατομμύρια (224) χρώματα.

γ) **Μέγεθος.** Η εικόνα, ανεξάρτητα από το σχήμα της, καταλαμβάνει το χώρο ενός ορθογώνιου παραλληλογράμμου που την περιβάλλει. Η απαιτούμενη χωρητικότητα για την αποθήκευση μιας εικόνας τύπου bitmap δίνεται από τη σχέση

$$\text{μέγεθος} = [ (\text{πλάτος} \times \text{ύψος}) \times (\text{οριζ.} \times \text{κατακόρ. ανάλ.}) \times (\text{χρωμ. βάθ.}_1 + \dots + \text{χρωμ. βάθ.}_n) ] / 8$$

#### 42. Τι είναι τα ψηφιακά φίλτρα σε ένα οποιοδήποτε πρόγραμμα επεξεργασίας εικόνας; Αναφέρετε τέσσερις κατηγορίες φίλτρων που εφαρμόζουμε επί εικόνων bitmap και ποια είναι τα αποτελέσματά τους.

Τα ψηφιακά φίλτρα είναι έτοιμα εφέ, τα οποία μπορούν να εφαρμοστούν είτε σε ολόκληρη την εικόνα μας ή σε μια επιλεγμένη περιοχή. Το Photoshop CS6 περιέχει από μόνο του μια μεγάλη ποικιλία φίλτρων, αλλά υπάρχουν και αρκετές χιλιάδες ξεχωριστά φίλτρα με την μορφή plug-ins. Σε γενικές γραμμές, τα φίλτρα χρησιμοποιούνται για την προσθήκη ειδικών εφέ σε μια εικόνα. Μπορούν να χρησιμοποιηθούν στο ρετουσάρισμα, την δημιουργία ρεαλιστικών υφών, εντυπωσιακών εφέ κειμένου και σε πολλά άλλα πράγματα. Μπορούν να αλλάξουν εντελώς το περιβάλλον μιας εικόνας. Τα φίλτρα βρίσκονται στο μενού επιλογών Filters.

Μερικές από τις κατηγορίες αυτών είναι το Blur (δημιουργεί θόλωμα), sharpen (κάνει τις φωτογραφίες

να είναι αιχμηρές δηλαδή να έχουν σχέδιο), render(δημιουργεί οπτικά φαινόμενα φωτιάς, σύννεφα ανέμου κλπ)

#### 43. Περιγράψτε ποιο είναι το περιεχόμενο σ' ένα αρχείο διανυσματικών γραφικών.

Τα διανυσματικά γραφικά συντίθεται από γεωμετρικά σχήματα (σημεία, γραμμές, καμπύλες, ορθογώνια, ελλείψεις, πολύγωνα κ.α.) που περιγράφονται με μαθηματικό τρόπο, με τη βοήθεια συντεταγμένων, γωνιών και αποστάσεων. Το μέγεθος τους εξαρτάται από την πολυπλοκότητα των σχημάτων που περιγράφουν. Είναι σημαντικά μικρότερα σε σχέση με τα αντίστοιχα χαρτογραφικά αρχεία, επειδή αποθηκεύονται μόνο οι πληροφορίες που είναι απαραίτητες για τον σχεδιασμό των σχημάτων.

#### 44. Αναφέρετε πλεονεκτήματα και μειονεκτήματα των διανυσματικών γραφικών.

Πλεονεκτήματα:

- Τα διανυσματικά αρχεία παρέχουν δυνατότητες αλλαγής μεγέθους και περιστροφής των σχημάτων, χωρίς να προκαλούνται αλλοιώσεις.
- Τα νέα αρχεία διατηρούν αναλλοίωτα τα βασικά χαρακτηριστικά τους (σχετικές διαστάσεις και σχετική θέση).
- Τα διανυσματικά γραφικά είναι ανεξάρτητα ανάλυσης (resolution free) και προσαρμόζονται αυτόματα στο μέγεθος και την ανάλυση του μέσου που προβάλλονται ή εκτυπώνονται.

Μειονέκτημα:

- Για τη δημιουργία ενός σχήματος επιτρέπεται ο προσδιορισμός ενός μόνο χρώματος ή γέμισμα με χρωματική διαβάθμιση (color gradient), αλλά δεν είναι δυνατή η ενσωμάτωση φωτορεαλιστικής ποιότητας (σε αντίθεση με τις χαρτογραφικές εικόνες όπου μπορείς να αλλάξεις το χρώμα σε κάθε pixel).

#### 45. Ποιες είναι οι βασικές διαφορές μεταξύ bitmap graphics και vector graphics;

- Τα vector graphics αρχεία με την αλλαγή μεγέθους και περιστροφής των σχημάτων δεν προκαλούνται αλλοιώσεις ενώ τα bitmap graphics μεταβάλλονται και ειδικά η μεγάλες μεγεθύνεις προκαλούν απώλειες στη ποιότητα
- Τα vector graphics είναι ανεξάρτητα ανάλυσης (resolution free) και προσαρμόζονται αυτόματα στο μέγεθος και την ανάλυση του μέσου που προβάλλονται ή εκτυπώνονται ενώ τα bitmap graphics η εκτύπωσή τους εξαρτάται άμεσα με την αρχική τους ανάλυση
- Τα vector graphics. Για τη δημιουργία ενός σχήματος επιτρέπεται ο προσδιορισμός ενός μόνο χρώματος ή γέμισμα με χρωματική διαβάθμιση (color gradient), αλλά δεν είναι δυνατή η ενσωμάτωση φωτορεαλιστικής ποιότητας (σε αντίθεση με τις χαρτογραφικές εικόνες όπου μπορείς να αλλάξεις το χρώμα σε κάθε pixel).

#### 46. Περιγράψτε τις διαφορές μεταξύ των προτύπων TIFF, JPEG, GIF. Που χρησιμοποιείται το καθένα;

Από άποψη μεγέθους, οι εικόνες τύπου JPEG είναι πολύ μεγαλύτερες από τις άλλες δύο. Το jpg χρησιμοποιείται ευρέως για την διανομή εικόνων μέσω διαδικτύου ή ηλεκτρονικού ταχυδρομείου (mail) και αυτό γιατί μας δίνει πολύ περισσότερες λεπτομέρειες, συν του ότι είναι πολύ αποτελεσματικός στην διαχείρισή του. Από την άλλη, οι δύο άλλοι τύποι αρχείων -οι GIF και PNG- καταναλώνουν πολύ λιγότερο χώρο στο σύστημά μας και, συγχρόνως, είναι

αποτελεσματικοί στην χρήση και τις λειτουργίες τους. Σε ό,τι αφορά την στήριξη, ο τύπος αρχείων GIF υποστηρίζεται από όλους τους φυλλομετρητές (web browsers) -εν αντιθέσει με τον τύπο αρχείων PNG ο οποίος δεν υποστηρίζεται από όλους τους browsers καθώς αναπτύχθηκε ως μια δωρεάν εναλλακτική λύση του GIF. Όσον αφορά την χρήση τους τώρα· ο τύπος εικόνων JPEG χρησιμοποιείται για την αναπαράσταση των εικόνων που δεν εμπεριέχουν κίνηση, όπως και για φωτογραφίες και διάφορα είδη (φωτο)σκιάσεων αναλόγως με τον φωτισμό (στο σκοτάδι ή στο φως αντίστοιχα). Από την άλλη, η μορφή αρχείου εικόνας PNG πολύ συχνά χρησιμοποιείται για την δημιουργία των λογοτύπων που ενέχει διαφάνεια και ξεθώριασμα. Η μορφή αρχείου εικόνων PNG είναι η ιδανική και για τα screenshots, δηλαδή, για τα λογισμικά που μας παρέχουν την λήψη στιγμιότυπου οθόνης. Ο δε τύπος αρχείων GIF, χρησιμοποιείται σε κινούμενα σχέδια.

Και βέβαια, υπάρχουν πολλές περισσότερες διαφορές που μπορούν να εντοπιστούν μόνο από τους ειδικούς. Για παράδειγμα, ο τύπος αρχείων εικόνων GIF δεν έχει απώλειες -όπως ο τύπος JPEG που μπορεί να αλλάξει ριζικά την εικόνα, προς το χειρότερο όμως. Από την άλλη βέβαια, σε ορισμένες περιπτώσεις μπορεί (ο τύπος JPEG) να δώσει ένα ενδιαφέρον αποτέλεσμα στην εικόνα. Λόγω του ότι ο τύπος αρχείων εικόνων GIF δεν έχει απώλειες, μπορεί να χρησιμοποιηθεί στις περιπτώσεις που χρειάζεται να διατηρηθούν οι αυστηρές γραμμές στην τυπογραφία και στα γεωμετρικά σχήματα. Ο τύπος GIF δεν είναι ιδανικός για την σύγχρονη φωτογραφία, λόγω του ότι έχει περιορισμένους πίνακες χρωμάτων και μικρά μεγέθη. Από την άλλη, βέβαια, αυτά τα μικρά μεγέθη αποβαίνουν ιδιαίτερος χρήσιμα κάποιες φορές -ειδικά όταν δεν κρίνεται κατάλληλος ο τύπος JPEG εξαιτίας του τεραστίου μεγέθους του.

- 47. Να βρεθούν τα χρώματα που είναι στην παρακάτω λίστα. Υπάρχουν διαδικτυακά χρώματα συμπληρωματικά μεταξύ τους; Αν ναι, ποια είναι αυτά και ως προς τι είναι συμπληρωματικά;α) #FF0000 b) #FFFFFF c) #00FFFF d) #000000 e) #A1A1A1 Σημείωση: δοκιμάστε πρώτα να μετατρέψετε τις παραπάνω τιμές στην κλίμακα 0 έως 255 για κάθε κανάλι χρώματος.**

Από τα 6 σύμβολα μετά το # τα δύο πρώτα FF αφορούν το κόκκινο, το 3-4 FF το πράσινο και το 5-6 FF το μπλε. Έχουν τιμές 255 για το κάθε κύριο χρώμα. Έτσι έχουμε

- α) #FF0000 είναι το Κόκκινο (R:255, G:0, B:0)
- β) #FFFFFF είναι το Άσπρο (R:255, G:255, B:255)
- γ) #00FFFF είναι το γαλάζιο (R:0, G:255, B:255)
- δ) #000000 είναι το μαύρο(R:0, G:0, B:0)
- ε) #A1A1A1 είναι το γκρι (R:161, G:161, B:161)

Συμπληρωματικά είναι:

Για το α) #FF0000 είναι το γαλάζιο (R:0, G:255, B:255)

Για το β) #FFFFFF είναι το Μαύρο (R:0, G:0, B:0)

Για το γ) #00FFFF είναι το κόκκινο (R:255, G:0, B:0)

Για το δ) #000000 είναι το άσπρο (R:255, G:255, B:255)

Για το ε) #A1A1A1 είναι το γκρι (R:161, G:161, B:161)

#### 48. Εξηγήστε την τεχνική Antialiasing και Alpha channel που χρησιμοποιείται σε εφαρμογές επεξεργασίας εικόνας και φωτορεαλιστικής απεικόνισης αντικειμένων.

Ένα από τα σημαντικότερα προβλήματα των γραφικών είναι ότι κάθε γραμμή που είναι κυρτή ή παρουσιάζει κάποια κλίση εμφανίζει οδοντώματα, μοιάζοντας περισσότερο με μια μεγάλη σκάλα παρά με μια ευθεία γραμμή. Αυτές οι ακανόνιστες γραμμές είναι ιδιαίτερα εμφανείς στις κινούμενες εικόνες, πράγμα που τις καθιστά ιδιαίτερα ενοχλητικές στα βιντεοπαιχνίδια. Αυτό το φαινόμενο στα 3D γραφικά είναι γνωστό ως aliasing. Το φαινόμενο εμφανίζεται διότι η εικόνα στην οθόνη είναι μόνο ένα δείγμα από pixels της αρχικής τρισδιάστατης εικόνας που η κάρτα γραφικών σας έχει υπολογίσει. Στις μεγαλύτερες αναλύσεις, καθώς χρησιμοποιούνται περισσότερα pixels, το δείγμα έρχεται πιο κοντά στην αρχική του μορφή και ως εκ τούτου η εικόνα είναι ευκρινέστερη και εμφανίζει πολύ λιγότερο aliasing.

Για να αντιμετωπιστεί το πρόβλημα χρησιμοποιείται μια τεχνική που ονομάζεται Full Scene Anti-Aliasing (FSAA) ή απλώς Anti-Aliasing (AA). Το Anti-Aliasing βοηθά στο να εμφανίζονται οι ακανόνιστες αυτές γραμμές πιο “ομαλές”, χωρίς να χρειάζεται να αυξήσετε την ανάλυση της οθόνης σας. Ουσιαστικά το AA συνδυάζει τις ακανόνιστες γραμμές με το περιβάλλον τους και μπορεί να εφαρμοστεί, είτε σε 2D ή 3D γραφικά.

#### Τύποι anti-aliasing

Υπάρχουν διάφορες μέθοδοι για την επίτευξη του anti-aliasing. Οι περισσότερες από αυτές βασίζονται σε δύο μεθόδους:

1. Στη μέθοδο του **sampling**: το sampling είναι μια διαδικασία όπου επιλέγεται μία μόνο τιμή από μια συνεχώς μεταβαλλόμενη σειρά τιμών. Το αποτέλεσμα του sampling είναι μία τιμή που ονομάζεται sample. Στα 3D γραφικά, ένα sample περιγράφει μια περιοχή γεωμετρίας.
2. Στη μέθοδο του **post-processing**: σε αυτή τη μέθοδο ένα αλγόριθμος τροποποιεί την εικόνα λίγο πριν αυτή προβληθεί στη οθόνη μας, αφαιρώντας, όσο aliasing μπορεί να εντοπίσει.

Η πιο κοινή μέθοδος πριν από λίγα χρόνια ήταν το **Super Sampling (SSAA)** η οποία απλά αυξάνει την ανάλυση ολόκληρης της εικόνας εντός της κάρτας γραφικών, την οποία μετά συρρικνώνει στην επιλεγμένη ανάλυση της οθόνης σας. Αυτή η μέθοδος μειώνει τις επιδόσεις σε πολύ μεγάλο βαθμό, αλλά λειτουργεί με σχεδόν οποιοδήποτε παιχνίδι, προσφέροντας τα καλύτερα αποτελέσματα.

Το **Multi Sampling (MSAA)** είναι μια νεότερη μορφή AA που βασίζεται στο super sampling anti-aliasing, αλλά αντί να αυξάνει την ανάλυση ολόκληρης της εικόνας εντοπίζει τις πλευρές που χρειάζονται anti-aliasing και επεξεργάζεται μόνο τις περιοχές αυτές. Έτσι ο αντίκτυπος στις επιδόσεις είναι πολύ μικρότερος.

#### 49. Τι είναι το encapsulation; Δώστε ένα παράδειγμα σε γλώσσα προγραμματισμού Java.

Ο συνδυασμός δεδομένων και συναρτήσεων μέσα στα αντικείμενα, και η «προστασία» των private δεδομένων μέσω των public συναρτήσεων λέγεται ενθυλάκωση (encapsulation).

#### Παράδειγμα Java:

```
public class EncapTest {  
    private String name;  
    private String idNum;  
    private int age;  
}
```

```
public int getAge() {
    return age;
}

public String getName() {
    return name;
}

public String getIdNum() {
    return idNum;
}

public void setAge( int newAge) {
    age = newAge;
}

public void setName(String newName) {
    name = newName;
}

public void setIdNum( String newId) {
    idNum = newId;
}
}

public class RunEncap {

    public static void main(String args[]) {
        EncapTest encap = new EncapTest();
        encap.setName("James");
        encap.setAge(20);
        encap.setIdNum("12343ms");

        System.out.print("Name : " + encap.getName() + " Age : " + encap.getAge());
    }
}
```

- 50. Δημιουργήστε ένα πρόγραμμα σε Java το οποίο θα ζητάει από το χρήστη να εισάγει/πληκτρολογήσει μία φράση και στη συνέχεια θα εμφανίζει πόσες φορές εμφανίζεται το κάθε φωνήεν στη φράση.**

```
import java.util.Scanner;
public class Test {
    public static void main(String[] args) {
        Scanner SC = new Scanner(System.in);
        System.out.print("Enter a phrase: ");
        String s = SC.next().toLowerCase();
        int vowelCount = 0;
        int count[]=new int [5];

        for (int i = 0; i<5;i++){
            count[i]=0;
        }
    }
}
```

```

        for (int i = 0; i < s.length(); ++i) {
            switch(s.charAt(i)) {
                case 'a':
                    count[0]++;break;
                case 'e':
                    count[1]++;break;
                case 'i':
                    count[2]++;break;
                case 'o':
                    count[3]++;break;
                case 'u':
                    count[4]++;break;
                case ' ':
                    break;
                default:
            }
        }
        for (int i = 0; i<5;i++){
            switch(i) {
                case 0:
                    System.out.println("a->" +count[i]);break;
                case 1:
                    System.out.println("e->" +count[i]);break;
                case 2:
                    System.out.println("i->" +count[i]);break;
                case 3:
                    System.out.println("o->" +count[i]);break;
                case 4:
                    System.out.println("u->" +count[i]);break;
                default:
            }
        }
    }
}

```

**51. Δημιουργήστε ένα πρόγραμμα σε Java το οποίο θα σχεδιάζει ένα κύκλο. Το χρώμα και το μέγεθος της ακτίνας του κύκλου θα καθορίζεται από τον χρήστη.**

```

package circle;
import java.lang.*;
import java.util.*;
import java.util.List;
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;
import java.lang.reflect.Field;
import java.util.logging.Level;
import java.util.logging.Logger;
public class Circle extends Frame {

    public static void main(String args[])
    {
        Circle frame = new Circle();
        frame.addWindowListener(
            new WindowAdapter()
            {

```

```

        public void windowClosing(WindowEvent we)
        {
            System.exit(0);
        }
    };

    frame.setSize(400, 400);
    frame.setVisible(true);
}

public void paint(Graphics g) {
    Scanner SC = new Scanner(System.in);
    System.out.print("SET R: ");
    int R = SC.nextInt();
    System.out.print("\nSET COLOR: ");
    String current_color = SC.next();
    Graphics2D ga = (Graphics2D)g;

    Field field=null;
    try {
        field
Class.forName("java.awt.Color").getField(current_color.toLowerCase());
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Circle.class.getName()).log(Level.SEVERE, null, ex);
    } catch (NoSuchFieldException ex) {
        Logger.getLogger(Circle.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SecurityException ex) {
        Logger.getLogger(Circle.class.getName()).log(Level.SEVERE, null, ex);
    }
    Color color = null;
    try {
        color = (Color)field.get(null);
    } catch (IllegalArgumentException ex) {
        Logger.getLogger(Circle.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        Logger.getLogger(Circle.class.getName()).log(Level.SEVERE, null, ex);
    }
    ga.setPaint(color);
    ga.drawOval(150,150,R,R);
}
}

```

## 52. Τι είναι τα Demon Threads και τί τα Non-Demon Threads; Από που εκτελούνται και πως;

**Demon** είναι τα **Threads** που συνεχίζουν να ζουν ακόμα και αν ο πατέρας τους πέθανε. Αντίθετα τα **Non-Demon** πεθαίνουν όταν πεθάνει αυτός που τα δημιούργησε.

Τα Demon Threads στην Java είναι τα threads που τρέχουν στο παρασκήνιο και δημιουργούνται κυρίως από την JVM(java virtual machine) για την εκτέλεση εργασιών στο παρασκήνιο, όπως τη συλλογή σκουπιδιών(garbage collection). Τα Demon Threads τρέχουν ακόμα και μετά το τέλος του προγράμματος.Η **βασική διαφορά** μεταξύ Demon Threads και Non-Demon Threads είναι ότι η JVM δεν περιμένει κάποιο Demon Thread πριν τερματίσει η εφαρμογή, ενώ τα Non-Demon Threads τα περιμένει και δεν τερματίζει αν τα Non-Demon Threads δεν τελειώσουν την εκτέλεσή τους.



- 53. Δίνεται το μη ASCII αρχείο DATA.dbs όπου κάθε εγγραφή είναι αντικείμενο που έχει την ακόλουθη δομή: IBAN λογαριασμού (20 χαρακτήρες) Ανάλυση/Κατάθεση (Ο/Υ αντίστοιχα) Ποσό συναλλαγής (δεκαδικός αριθμός) Όνομα πελάτη (40 χαρακτήρες) ΑΦΜ πελάτη (ακέραιος 8-ψήφιος).**

**Να γραφεί πρόγραμμα Java προσπέλασης του αρχείου DATA.dbs που να τυπώνει στην οθόνη το άθροισμα των καταθέσεων, το άθροισμα των αναλήψεων καθώς και τα στοιχεία του πελάτη με τη μεγαλύτερη ανάληψη και κατάθεση.**

```
import java.io.*;
import com.linuense.javadb.*;
import org.xBase.*;
import org.xBase.fields.CharField;
import org.xBase.fields.LogicalField;
import org.xBase.fields.NumField;
import org.xBase.fields.DoubleField;

public class JavaDBFReaderTest {

    public static void main( String args[]) {

        String IBAN[];
        String TRANSACTION[];
        Double AMOUNT[];
        String CREDENTIALS[];
        String TIN[];
        CharField iban;
        CharField transaction;
        DoubleField amount;
        CharField credentials;
        CharField tax_number;
        int numberOfFields;
        double withdrawals=0;
        double deposits=0;
        double max_withdrawal;
        double max_deposit;
        int max_withdrawal_customer;
        int max_deposit_customer;

        try{
            InputStream inputStream=null;
            try {
                inputStream = new FileInputStream("C:/DATA.dbs");
            } catch (IOException e) {
                e.printStackTrace();
            } finally {
                try {
                    if (inputStream != null)
                        inputStream.close();
                } catch (IOException ex) {
                    ex.printStackTrace();
                }
            }

            DBFReader dataDB = new DBFReader(inputStream);
            numberOfFields = dataDB.getFieldCount();
            IBAN=new String[numberOfFields];
            TRANSACTION=new String[numberOfFields];
            AMOUNT=new Double[numberOfFields];
            CREDENTIALS=new String[numberOfFields];
            TIN=new String[numberOfFields];
```

```
for(int i=0; i<numberOfFields; i++) {
    dataDB.read();
    iban = (CharField) dataDB.getField("IBAN");
    IBAN[i]=iban.getValue();
    transaction = (CharField) dataDB.getField("TRANSACTION");
    TRANSACTION[i]=transaction.getValue();
    amount = (CharField) dataDB.getField("AMOUNT");
    AMOUNT[i]=amount.getValue();
    credentials = (CharField) dataDB.getField("CREDENTIALS");
    CREDENTIALS[i]=credentials.getValue();
    tax_number = (CharField) dataDB.getField("TAX_IDENTIFICATION_NUMBER");
    TIN[i]=tax_number.getValue();
}

max_withdrawal=AMOUNT[0];
max_deposit=AMOUNT[0];
for(int i=0; i<numberOfFields; i++) {
    if(TRANSACTION[i].equals('O')){
        withdrawals=withdrawals+AMOUNT[i];
        if(AMOUNT[i]>max_withdrawal){
            max_withdrawal=AMOUNT[i];
            max_withdrawal_customer=i;
        }
    }
    if(TRANSACTION[i].equals('Y')){
        deposits=deposits+AMOUNT[i];
        if(AMOUNT[i]>max_deposit){
            max_deposit=AMOUNT[i];
            max_deposit_customer=i;
        }
    }
}

System.out.println("The total ammount of withdrawals is: "+withdrawals);
System.out.println("The total ammount of deposits is: "+deposits);
System.out.println("The customer with max withdrawal amount is
"+CREDENTIALS[max_withdrawal_customer]+ " with TAX_IDENTIFICATION_NUMBER " +
TIN[max_withdrawal_customer]+" and IBAN " + IBAN[max_withdrawal_customer]);
System.out.println("The customer with max deposit amount is
"+CREDENTIALS[max_deposit_customer]+ " with TAX_IDENTIFICATION_NUMBER " +
TIN[max_deposit_customer]+" and IBAN " + IBAN[max_deposit_customer]);
    inputStream.close();
}
catch( DBFException e) {

    System.out.println( e.getMessage());
}
catch( IOException e) {

    System.out.println( e.getMessage());
}
}
}
```

**54. Περιγράψτε τη βασική δομή μιας εφαρμογής φτιαγμένης σε OpenGL.**

Η βασική δομή μιας εφαρμογής σε OpenGL αποτελείται από δύο κυρίως κομμάτια. Το πρώτο κομμάτι είναι η αρχικοποίηση όπου καθορίζονται οι παράμετροι και οι βασικές μεταβλητές απαραίτητες για την δημιουργία της σκηνής. Οι παράμετροι αυτοί σχετίζονται με το μέγεθος της οθόνης, τα χρώματα, το είδος της σκηνής(τρεις ή δύο διαστάσεις) κτλ. Σε δεύτερη φάση μια OpenGL εφαρμογή δημιουργεί την σκηνή συνεχώς και αλληλεπιδρά με τον χρήστη. Εδώ δηλαδή, έχουμε συναρτήσεις όπου δημιουργούν κάθε ένα Pixel στην οθόνη, αλληλεπιδρούν με το πληκτρολόγιο και το ποντίκι και είναι υπεύθυνες για το resize του παραθύρου

**55. Να αναφέρετε και να περιγράψετε σύντομα τα callback functions της OpenGL.**

- 1 glutDisplayFunc
- 2 glutOverlayDisplayFunc
- 3 glutReshapeFunc
- 4 glutKeyboardFunc
- 5 glutMouseFunc
- 6 glutMotionFunc, glutPassiveMotionFunc
- 7 glutVisibilityFunc
- 8 glutEntryFunc
- 9 glutSpecialFunc
- 10 glutSpaceballMotionFunc
- 11 glutSpaceballRotateFunc
- 12 glutSpaceballButtonFunc
- 13 glutButtonBoxFunc
- 14 glutDialsFunc
- 15 glutTabletMotionFunc
- 16 glutTabletButtonFunc
- 17 glutMenuStatusFunc
- 18 glutIdleFunc
- 19 glutTimerFunc

**56. Ποιες εντολές χρησιμοποιούμε για τις εξής λειτουργίες της OpenGL: α. translate β. rotate γ. scale ή stretch;**

Η συνάρτηση `glTranslate` πολλαπλασιάζει τον τρέχων πίνακα με έναν translation πίνακα. Η συνάρτηση αυτή παράγει ένα translation πίνακα βάσει των  $x$   $y$   $z$ . Ο τρέχων πίνακας πολλαπλασιάζεται με αυτόν το translation πίνακα και το γινόμενο αντικαθιστά τον τρέχοντα πίνακα. Έχει τις συναρτήσεις C, `void glTranslated(GLdouble x, GLdouble y, GLdouble z)` και `void glTranslatef(GLfloat x, GLfloat y, GLfloat z)`, όπου οι παράμετροι  $x$ ,  $y$ ,  $z$  καθορίζουν τις  $x$ ,  $y$ , και  $z$  συντεταγμένες ενός translation vector.

Η συνάρτηση `glRotate` πολλαπλασιάζει τον τρέχων πίνακα με έναν rotation πίνακα. Η συνάρτηση αυτή παράγει μια επίστρωση της γωνίας των αξόνων περί το διάνυσμα  $x, y, z$ . Ο τρέχων πίνακας πολλαπλασιάζεται με αυτόν το rotation πίνακα και το γινόμενο αντικαθιστά

τον τρέχοντα πίνακα. Έχει τις συναρτήσεις C, void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z) και void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z); όπου οι παράμετροι angle, x, y, z καθορίζουν τη γωνία περιστροφής και τις x, y, z συντεταγμένες αντίστοιχα.

Η συνάρτηση glScale πολλαπλασιάζει τον τρέχον πίνακα με έναν γενικό scaling πίνακα. Η συνάρτηση αυτή παράγει μια ανομοιόμορφη κλιμάκωση (scaling) στους x,y,z άξονες. Οι τρεις παράμετροι x,y,z καθορίζουν την επιθυμητή κλιμάκωση κατά μήκος των x,y,z αξόνων. Έχει τις συναρτήσεις C, void glScaled(GLdouble x, GLdouble y, GLdouble z) και void glScalef(GLfloat x, GLfloat y, GLfloat z) , όπου οι παράμετροι x, y, z των x,y,z αξόνων αντίστοιχα. Επιπρόσθετα, η οι συναρτήσεις glViewport glOrtho αξιοποιούνται στο να πολλαπλασιαστού με έναν πίνακα που ορίζει πόσο stretch πρέπει να γίνει. Το ortho ορίζει το νέο ορθοκανονικό σύστημα συντεταγμένων και το viewport αλλάζει το view που θα έχει το object πάνω στο νέο σύστημα συντεταγμένων.

### 57. Για ποιο λόγο είναι καλό να χρησιμοποιούμε τους τύπους δεδομένων της OpenGL;

Με τους τύπους δεδομένων της OpenGL επιτυγχάνεται κυρίς καλύτερο compatibility γιατί η OpenGL είναι cross-platform δηλαδή ανεξάρτητη λειτουργικού, και ως εκ τούτου, δεν απαιτείται περεταίρω κώδικας ή compile.

Εντοπίζονται κυρίως έξι βασικοί τύποι δεδομένων, μερικοί από αυτούς είναι οι ίδιοι με τους C, άλλοι προστίθενται ειδικά για προγραμματισμό GPU, αυτοί οι τύποι είναι:

float - ένας αριθμός κινητής υποδιαστολής 32 bit

half - έναν αριθμό κινητής υποδιαστολής 16 bit

int - ένας ακέραιος 32-bit

fixed - αριθμός σταθερού σημείου 12-bit

bool - μια μεταβλητή boolean

sampler \* - αντιπροσωπεύει ένα αντικείμενο υφής

Το Cg διαθέτει επίσης τύπους δεδομένων vector και matrix που βασίζονται στους βασικούς τύπους δεδομένων, όπως float3, float4\*4, αυτοί οι τύποι δεδομένων είναι αρκετά συνηθισμένοι όταν αξιοποιούνται στον προγραμματισμό τρισδιάστατων γραφικών, το Cg έχει επίσης τύπους δεδομένων struct και array, οι οποίοι λειτουργούν σε παρόμοια τρόπο ισοδύναμα C.

### 58. Περιγράψτε τη σύνταξη και τις παραμέτρους της glLookAt() (στην OpenGL).

Η gluLookAt δίνει την δυνατότητα να τοποθετήσουμε μια κάμερα στην σκηνή και να θέσουμε την κατεύθυνση της. Συντάσσεται ως εξής:

```
gluLookAt (GLdouble eyex, GLdouble eyey, GLdouble eyez, GLdouble centerx, GLdouble centery, GLdouble centerz, GLdouble upx, GLdouble upy, GLdouble upz);
```

Η εντολή παίρνει ως παραμέτρους 3 διανύσματα: το [eyex, eyey, eyez] ορίζει τις συντεταγμένες του σημείου θέασης (μάτι), το [centerx, centery, centerz] καθορίζει την κατεύθυνση (τι δείχνει δηλαδή) και το [upx, upy, upz] είναι το μοναδιαίο διάνυσμα που καθορίζει ποια είναι η προς τα πάνω κατεύθυνση της κάμερα (ποιος ο άξονας y).

**59. Ποια είναι η λειτουργία της glClear() και ποια της glClearColor3f() (στην OpenGL);**

Η glClear() είναι μια συνάρτηση της OpenGL που καθαρίζει τα buffers(ενταμιευτές) και επαναφέρει τις τιμές που υπήρχαν πριν οποιαδήποτε αλλαγή. Ανάλογα με τα buffers που θέλουμε να καθαρίσουμε βάζουμε και τα ανάλογα ορίσματα. Η glClear μπορεί να καθαρίσει ένα και περισσότερα buffers ταυτόχρονα, με την χρήση με το σύμβολο του bitwise OR.

π.χ. `glClear(GL_COLOR_BUFFER_BIT);` // καθαρίζεται ο buffer του χρώματος.

Η εντολή glClearColor3f() δεν υπάρχει από όσο γνωρίζω, θα απαντήσω για την εντολή glClearColor(). Η glClearColor() είναι εντολή καθαρισμού της οθόνης. Καθορίζει το χρώμα που χρησιμοποιείται κάθε φορά που εκτελείται. Στην OpenGL το χρώμα του φόντου είναι μία μεταβλητή κατάστασης, η οποία διατηρεί την τιμή που της ανατέθηκε την τελευταία φορά. Το χρώμα καθορίζεται από τα βάρη του στο χρωματικό μοντέλο RGBA. Ως παραμέτρους καταχωρούμε τις τιμές σε εύρος [0,1] του κόκκινου, του πράσινου, του μπλε και της διαφάνειας.

π.χ. `glClearColor(1,1,1,1);` // Για να χρησιμοποιήσουμε το λευκό χρώμα

**60. Στην OpenGL όπως και στα περισσότερα API γραφικών χρησιμοποιούμε "normalized values", τι σημαίνει αυτό;**

Στην OpenGL όπως και στα περισσότερα API γραφικών χρησιμοποιούμε κανονικοποιημένες τιμές (normalized values) για τις διαστάσεις και τη θέση των αντικειμένων, που σημαίνει ότι μια τιμή 1 υποδηλώνει ολόκληρο το πλάτος ή το ύψος της οθόνης (ανάλογα με την παράμετρο) ανεξάρτητα από την ανάλυση οθόνης. Αυτό επιτρέπει την εύκολη διατήρηση της αναλογίας διαστάσεων γραφικών αντικειμένων, ανεξάρτητα από τη συσκευή προβολής.

**61. Αναφέρατε τις εντολές (μαζί με τις παραμέτρους τους) για προοπτική (perspective) και για ορθογραφική (orthographic) απεικόνιση (στην OpenGL).**

```
glViewport(xmin, ymin, width, height); //perspective  
glOrtho (Left, Right, Bottom, Top, Near, Far); //orthographic
```

**62. Με ποιους τρόπους μπορούμε να μετατοπίσουμε ένα αντικείμενο -σχήμα στην OpenGL;**

Μετακίνηση (translation) ενός αντικειμένου με την glTranslatef(GLfloat X, GLfloat Y, GLfloat Z). Η εντολή αυτή παίρνει ως παραμέτρους την μετακίνηση που επιθυμούμε κατά άξονα, κατασκευάζει ένα μετασχηματισμό μετακίνησης T και τον εφαρμόζει (πολλαπλασιάζει) με τον μετασχηματισμό Μοντέλου-Κάμερας. Ένας άλλος τρόπος είναι να μετατοπίσουμε την κάμερα.

**63. Περιγράψτε με εντολές και παραμέτρους της OpenGL, τα βασικά γεωμετρικά σχήματα α) Γραμμές, β) Τρίγωνα, γ) Πολύγωνα. Παραθέστε σχήματα.**

GL\_POINTS Σχεδιάζει σημεία. Κάθε κορυφή είναι ένα σημείο.

```
glBegin(GL_POINTS); //starts drawing of points
    glVertex3f(1.0f,1.0f,0.0f); //upper-right corner
    glVertex3f(-1.0f,-1.0f,0.0f); //lower-left corner
glEnd(); //end drawing of points
```

GL\_LINES Σχεδιάζει γραμμές. Κάθε ζεύγος κορυφών σχεδιάζει μια γραμμή.

```
glBegin(GL_LINES);
glVertex2i(20,20);
glVertex2i(40,40);
glEnd();
```

GL\_LINE\_STRIP Σχεδιάζει ενωμένες γραμμές. Κάθε κορυφή μετά τις δύο πρώτες ενώνεται.

```
glBegin(GL_LINE_STRIP);
glVertex3f(0.2, 0.2,0.0);
glVertex3f(0.8, 0.2,0.0);
glVertex3f(0.2, 0.5,0.0);
glVertex3f(0.8, 0.5,0.0);
glVertex3f(0.2, 0.8,0.0);
glVertex3f(0.8, 0.8,0.0);
glEnd();
```

GL\_LINE\_LOOP Σχεδιάζει ενωμένες γραμμές. Η τελευταία κορυφή ενώνεται με την πρώτη.

```
glBegin(GL_LINE_LOOP); //start drawing a line loop
glVertex3f(-1.0f,0.0f,0.0f); //left of window
glVertex3f(0.0f,-1.0f,0.0f); //bottom of window
glVertex3f(1.0f,0.0f,0.0f); //right of window
glVertex3f(0.0f,1.0f,0.0f); //top of window
glEnd(); //end drawing of line loop
```

GL\_TRIANGLES Κάθε τρεις κορυφές σχεδιάζουν ένα τρίγωνο.

```
glBegin(GL_TRIANGLES); //start drawing triangles
glVertex3f(-1.0f,-0.25f,0.0f); //triangle one first vertex
glVertex3f(-0.5f,-0.25f,0.0f); //triangle one second vertex
glVertex3f(-0.75f,0.25f,0.0f); //triangle one third vertex
glEnd(); //end drawing of triangles
```

GL\_TRIANGLE\_STRIP Σχεδιάζει συνδεδεμένα τρίγωνα. Κάθε κορυφή μετά τις τρεις πρώτες σχεδιάζει τρίγωνο.

```
glBegin(GL_TRIANGLE_STRIP);
glVertex3f(0.2, 0.2,0.0);
glVertex3f(0.8, 0.2,0.0);
glVertex3f(0.2, 0.5,0.0);
glVertex3f(0.8, 0.5,0.0);
```

```
glVertex3f(0.2, 0.8,0.0);  
glVertex3f(0.8, 0.8,0.0);  
glEnd();
```

GL\_TRIANGLE\_FAN Όπως και η GL\_TRIANGLE\_STRIP, σχεδιάζει σε σχήμα βεντάλιας.

```
glBegin(GL_TRIANGLE_FAN);  
glVertex3f(0.1, 0.1,0.0);  
glVertex3f(0.6, 0.1,0.0);  
glVertex3f(0.8,0.3,0.0);  
glVertex3f(0.6,0.6,0.0);  
glVertex3f(0.1,0.6,0.0);  
glVertex3f(0.0,0.3,0.0);  
glEnd();
```

GL\_QUADS Σχεδιάζει τετράπλευρα. Κάθε τέσσερις κορυφές σχεδιάζουν ένα τετράπλευρο.

```
glBegin(GL_QUADS);  
glVertex3f(0.2, 0.2,0.0);  
glVertex3f(0.8, 0.2,0.0);  
glVertex3f(0.6,0.4,0.0);  
glVertex3f(0.4,0.4,0.0);  
glVertex3f(0.4,0.6,0.0);  
glVertex3f(0.6,0.6,0.0);  
glVertex3f(0.8,0.8,0.0);  
glVertex3f(0.2,0.8,0.0);  
glEnd();
```

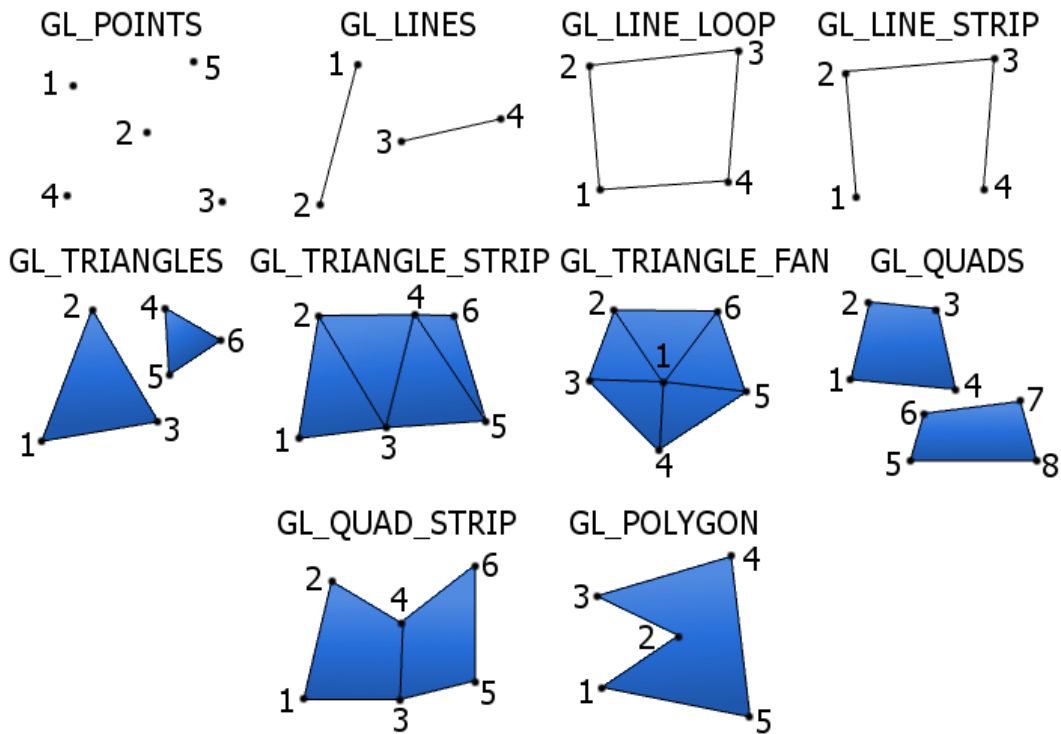
GL\_QUAD\_STRIP Κάθε δύο κορυφές μετά τις 4 πρώτες σχεδιάζουν ένα συνδεδεμένο τετράπλευρο.

```
glBegin(GL_QUAD_STRIP);  
glVertex3f(0.2, 0.2,0.0);  
glVertex3f(0.8, 0.2,0.0);  
glVertex3f(0.2,0.4,0.0);  
glVertex3f(0.8,0.4,0.0);  
glVertex3f(0.2,0.8,0.0);  
glVertex3f(0.8,0.8,0.0);  
glEnd();
```

GL\_POLYGON Πολύγωνο. Το πολύγωνο μπορεί να έχει όσες πλευρές θέλετε.

```
glBegin(GL_POLYGON);//begin drawing of polygon  
glVertex3f(-0.5f,0.5f,0.0f);//first vertex  
glVertex3f(0.5f,0.5f,0.0f);//second vertex  
glVertex3f(1.0f,0.0f,0.0f);//third vertex  
glVertex3f(0.5f,-0.5f,0.0f);//fourth vertex  
glVertex3f(-0.5f,-0.5f,0.0f);//fifth vertex  
glVertex3f(-1.0f,0.0f,0.0f);//sixth vertex  
glEnd();//end drawing of polygon
```





**64. Να γραφεί function (στην OpenGL) με όνομα keyboard που να λειτουργεί έτσι ώστε, όταν ο χρήστης πατήσει το 'Q' (ή 'q') να κλείνει την εφαρμογή και όταν πατήσει το 'C' (ή 'c') να καθαρίζει την οθόνη στο χρώμα του GL\_COLOR\_BUFFER\_BIT. Επίσης να γράψετε την εντολή του mainloop που θα το καλέσετε.**

```
void keyboard (unsigned char key, int x, int y) {
    switch (key) {
        case 'q':
        case 'Q':
            exit(0);
        break;
        case 'c':
        case 'C':
            glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        break;
    }
}

int main(int argc, char** argv) {
    ...
    glutKeyboardFunc(keyboard);
    ....
    glutMainLoop();
    return 0;
}
```

## 65. Περιγράψτε τα βήματα για τη δημιουργία animation στην OpenGL.

Η OpenGL χρησιμοποιεί μια μηχανή καταστάσεων (state machine) για να επικοινωνεί με την εφαρμογή. Σε αυτή την μηχανή καταστάσεων η OpenGL παραμένει διαρκώς σε μια κατάσταση μέχρι να αλλάξει η εφαρμογή την κατάσταση. Παράδειγμα αν θέσουμε το χρώμα που θα χρησιμοποιεί η OpenGL για να ζωγραφίσει ένα μοντέλο, το χρώμα θα παραμείνει στη μνήμη και θα χρησιμοποιείται μέχρι να το αλλάξουμε ή να κλείσουμε την εφαρμογή.

Εφόσον λοιπόν η εφαρμογή καθορίσει το περιβάλλον που θα χρησιμοποιήσει η OpenGL για να απεικονίσει το μοντέλο μας (χρώματα, υφές, πηγές φωτός, κάμερα κλπ), περνάει στο πρώτο στάδιο κατά το οποίο θα μετασχηματίζει και θα φωτίσει (transform and lighting) τα σημεία(vertices) του μοντέλου.

Στην συνέχεια η OpenGL περνά στο στάδιο της ψηφιοποίησης το οποίο λαμβάνει όλες τις πληροφορίες και την γεωμετρία από το προηγούμενο στάδιο (του μετασχηματισμού) και παράγει την τελική, ψηφιακή, εικόνα. Η εικόνα αντιγράφεται στην μνήμη του frame buffer και φτάνει έτσι στην οθόνη του υπολογιστή.

Η OpenGL είναι ένας γρήγορος και ευέλικτος τρόπος να επικοινωνούμε με το hardware γραφικών του υπολογιστή χωρίς να ενδιαφερόμαστε για τις λεπτομέρειες υλοποίησης του. Όμως δεν προσφέρει καθόλου λειτουργίες GUI (Graphical User Interface), δηλαδή δεν έχει την δυνατότητα να ανοίξει και να κλείσει παράθυρα στο λειτουργικό σύστημα, να ζωγραφίσει σε αυτά, ούτε να καταλάβει το πάτημα ενός πλήκτρου ή την κίνηση του ποντικιού, ούτε μπορεί να διαβάσει ένα αρχείο από το δίσκο.

Αυτό έγινε επί σκοπού, μιας και η OpenGL σχεδιάστηκε να τρέχει σε πολλά λειτουργικά συστήματα τα οποία έχουν το δικό τους τρόπους επικοινωνίας με την οθόνη, το ποντίκι, το δίσκο, το πληκτρολόγιο.

Για να γίνει αυτό θα πρέπει να χρησιμοποιήσουμε απευθείας τις εντολές του λειτουργικού μας συστήματος (Win32 εντολές στην περίπτωση των Windows). Εναλλακτικά μπορούμε να χρησιμοποιήσουμε μια από τις έτοιμες βιβλιοθήκες εντολών που υπάρχουν και που θα κάνουν αυτές τις λειτουργίες για εμάς εύκολα.

Μια από τις πιο διαδεδομένες βιβλιοθήκες για αυτό το σκοπό είναι το GLUT (OpenGL Utility Toolkit) το οποίο είναι και αυτό σχεδιασμένο να τρέχει σε πολλά λειτουργικά συστήματα.

Το GLUT προσφέρει ένα σύνολο εντολών που αναλαμβάνουν να ανοίξουν και να κλείσουν εύκολα παράθυρα, να καταγράψουν το πάτημα ενός πλήκτρου ή την κίνηση του ποντικιού. Δεν είναι κατάλληλο να γράψουμε μια κανονική εφαρμογή με αυτό (πχ ένα παιχνίδι) όμως είναι πολύ κατάλληλο για εκπαιδευτικές και πρότυπες εφαρμογές

## 66. Ποιες είναι οι ιδιότητες που έχουν οι επιφάνειες στην OpenGL;

Στον πραγματικό κόσμο, ένας παρατηρητής αντιλαμβάνεται την παρουσία επιφανειών από τις ανακλάσεις που αυτές προκαλούν στις ακτίνες των πηγών φωτισμού. Κάθε επιφάνεια γίνεται αντιληπτή με διαφορετικά χαρακτηριστικά ανάλογα με τα χαρακτηριστικά της ανακλαστικότητάς της (χρώμα). Επιπλέον όμως, παίζει ρόλο και η σχετική θέση της κάθε επιφάνειας ως προς τις υπάρχουσες

πηγές φωτισμού. Συνοψίζοντας μπορούμε να πούμε ότι οι βασικές ιδιότητες, οι οποίες καθορίζουν τα χαρακτηριστικά των επιφανειών είναι:

- Το χρώμα
- Προσανατολισμός επιφανείας (NORMALIZATION). Προκειμένου να περιγράψουμε τον προσανατολισμό μιας επιφάνειας σε κάθε σημείο της, χρησιμοποιούμε την έννοια του κανονικού διανύσματος.
- Σκίαση Gouraud, κάθε στοιχειώδης πολυγωνική επιφάνεια χαρακτηρίζεται από το κανονικό της διάνυσμα, το οποίο χρησιμοποιείται για υπολογισμούς από το μοντέλο φωτισμού.
- Υφή. Στις περιπτώσεις απόδοσης φυσικών σκηνών με περίπλοκη μορφολογία, καταφεύγουμε στη σχεδίαση ρεαλιστικών επιφανειών βασιζόμενοι σε προκαθορισμένα πρότυπα επιφανειών. Τα πρότυπα αυτά αποδίδουν με πιστότητα τη μορφολογία των επιφανειών και μπορούν να ληφθούν με τη μορφή ψηφιοποιημένων εικόνων. Στον κόσμο των γραφικών σε Η/Υ ένα τέτοιο πρότυπο απόδοσης αποκαλείται υφή (texture) και η διαδικασία της απόδοσης υφής σε μία επι-φάνεια ονομάζεται απόδοση υφής (texture mapping).

**67. Περιγράψτε αναλυτικά τι είναι το double buffering. Με ποια εντολή ενεργοποιούμε το double buffering σε RGB mode στην OpenGL; Με ποια εντολή κάνουμε swap τον front και τον back buffer έτσι ώστε να έχουμε animation;**

Στην επιστήμη των υπολογιστών, το double buffering είναι μια τεχνική για την σχεδίαση γραφικών, με την οποία αποκλείουμε εξολοκλήρου (ή σχεδόν εξολοκλήρου) το τραύλισμα, το τρεμοπαίξιμο, το σχίσσιμο και άλλα τεχνουργήματα που δημιουργούνται κατά τη σχεδίαση και αναπαράσταση γραφικών.

Η χρήση διπλού buffer (double buffering) κατά την απεικόνιση μιας σκηνής είναι μια βασική μέθοδος αποφυγής «τρεμοπαίγματος»(flickering). Ο τρόπος με τον οποίο δουλεύει είναι ορίζοντας 2 ενδιάμεσες περιοχές στην μνήμη (buffers) στις οποίες αποθηκεύουμε τις εικόνες που δημιουργούμε. Όσο η οθόνη παίρνει (απεικονίζει) την εικόνα από τον έναν buffer εμείς ζωγραφίζουμε την σκηνή στον άλλον. Έπειτα ανταλλάσσουμε τους ρόλους των buffer και συνεχίζουμε. Έτσι η οθόνη δείχνει πάντα μια ολοκληρωμένη εικόνα της σκηνής.

Η εντολή glutSwapBuffers(); θα αντικαταστήσει τον buffer που χρησιμοποιεί η οθόνη με αυτόν που περιέχει το αντικείμενο που μόλις ζωγραφίσαμε. Ανάλογα με αυτόν που χρησιμοποιούμε καθορίζουμε στη συνάρτηση αν θα είναι ο front ή ο back buffer.

Η εντολή glutInitDisplayMode(GLUT\_DOUBLE); θα ενεργοποιήσει τη λειτουργία του double buffer. Το RGB mode είναι από το πρόγραμμα αρχικοποιημένο.

**68. Για ποιό σκοπό χρειάζονται οι συναρτήσεις κλήσης στην OpenGL και πώς ορίζονται;**

Μια συνάρτηση κλήσης είναι μια λειτουργία την οποία καλεί η βιβλιοθήκη (GLUT) όταν χρειάζεται να ξέρει πώς να επεξεργαστεί κάτι. Για παράδειγμα όταν η GLUT λαμβάνει μια εισαγωγή κλειδιού από το πληκτρολόγιο ή το ποντίκι χρησιμοποιεί τη συνάρτηση κλήσης `glutKeyboardFunc` για να μάθει τι να κάνει με το εν λόγω πάτημα πλήκτρων.

Αυτή η τεχνική χρησιμοποιείται επίσης για ζωντανή ανατροφοδότηση/ μετάδοση, όπου καλεί το `TrackControl` (ή κάτι παρόμοιο) και καθορίζει μια ρουτίνα που θα καλέσει επανειλημμένα για να μετακινηθεί ο χρήστης στη σελίδα.

Συνεπώς χρησιμοποιούνται για τις παρακάτω διαδικασίες:

- Αναπροσαρμογή μεγέθους ή ανασύνταξη παραθύρου
- Εισαγωγή κλειδιού από χρήστη (ποντίκι, πληκτρολόγιο)
- Animation (καθιστούν πολλά πλαίσια)

Η βασική τους δομή είναι όνομα\_συνάρτησης(όνομα\_συνάρτησης\_γεγονότος που θα διαχειριστεί), π.χ. `glutKeyboardFunc( my_key_events_func );` – `glutMouseFunc ( my_mouse_events_func );`

**69. Με ποιές συναρτήσεις γίνεται μοντελοποίηση επιφανειών που ανακλούν στην OpenGL;**

Οι συναρτήσεις με τις οποίες γίνεται μοντελοποίηση επιφανειών που ανακλούν στην OpenGL είναι οι παρακάτω:

`genType reflect(genType I, genType N);`

`genDType reflect(genDType I, genDType N);`

**70. Σε πόσες και ποιες συνιστώσες διαχωρίζετε το φως πηγής κατά την μοντελοποίηση πηγών φωτισμού;**

Το φως πηγής κατά την μοντελοποίηση πηγών φωτισμού χωρίζεται σε 3 συνιστώσες, οι οποίες είναι οι εξής:

- Ένταση και χρώμα πηγής
- Θέση πηγής
- Κατεύθυνση εκπομπής - Γωνία αποκοπής

**71. Με ποια σειρά γίνονται οι διεργασίες για τον υπολογισμό της τελικής εικόνας (rendering) στην OpenGL;**

Η OpenGL χρησιμοποιεί μια μηχανή καταστάσεων (state machine) για να επικοινωνεί με την εφαρμογή. Σε αυτή την μηχανή καταστάσεων η OpenGL παραμένει διαρκώς σε μια κατάσταση μέχρι να αλλάξει η εφαρμογή την κατάσταση. Παράδειγμα αν θέσουμε το χρώμα που θα χρησιμοποιεί η OpenGL για να ζωγραφίσει ένα μοντέλο, το χρώμα θα παραμείνει στη μνήμη και θα χρησιμοποιείται μέχρι να το αλλάξουμε ή να κλείσουμε την εφαρμογή. Εφόσον λοιπόν η εφαρμογή καθορίσει το περιβάλλον που θα χρησιμοποιήσει η OpenGL για να απεικονίσει το μοντέλο μας (χρώματα, υφές,

πηγές φωτός, κάμερα κλπ), περνάει στο πρώτο στάδιο κατά το οποίο θα μετασχηματίσει και θα φωτίσει (transform and lighting) τα σημεία(vertices) του μοντέλου. Στην συνέχεια η OpenGL περνά στο στάδιο της ψηφιοποίησης το οποίο λαμβάνει όλες τις πληροφορίες και την γεωμετρία από το προηγούμενο στάδιο (του μετασχηματισμού) και παράγει την τελική, ψηφιακή, εικόνα. Η εικόνα αντιγράφεται στην μνήμη του frame buffer και φτάνει έτσι στην οθόνη του υπολογιστή. Αυτή η ακολουθία βημάτων που ακολουθεί η OpenGL για να απεικονίσει το μοντέλο λέγεται graphics pipeline (διασωλήνωση).

## **72. Με ποια εντολή και ποιες συναρτήσεις θα πραγματοποιήσετε απόδοση υφής σε γραμμές και επιφάνειες στην OpenGL;**

Η ανάθεση συντεταγμένων υφής γίνεται με τη χρήση της εντολής `glTexCoord{1,2}{i,s,f,d}`.

Για ορίσματα κινητής υποδιαστολής (GLfloat) οι συναρτήσεις είναι:

```
void glTexCoord1f ( GLfloat s );
```

για μμονοδιάστατες υφές

ή

```
void glTexCoord2f ( GLfloat s, GLfloat t );
```

για δισδιάστατες υφές.

## **73. Για ποιο λόγο χρησιμοποιούνται οι λίστες απεικόνισης (display lists) στην γλώσσα OpenGL και πώς συντάσσονται; Δώστε και από ένα παράδειγμα για κάθε περίπτωση.**

Συχνά, είναι βολικό η περιγραφή ενός σύνθετου γεωμετρικού σχήματος να περικλείεται σε μια αυτόνομη ενότητα κώδικα, η οποία θα εκτελείται κάθε φορά που θέλουμε να σχεδιάσουμε αυτό το σχήμα. Στην OpenGL τη δυνατότητα αυτή δίνουν οι λίστες απεικόνισης (display lists). Οι λίστες απεικόνισης διευκολύνουν την επαναχρησιμοποίηση κώδικα και απαλλάσσουν τον προγραμματιστή από περιττές επαναλαμβανόμενες δηλώσεις του ίδιου σύνθετου σχήματος. Μια λίστα απεικόνισης περικλείεται μεταξύ δύο εντολών: των `glNewList` και `glEndList`. Όταν σε ένα πρόγραμμα ορίζονται πολλαπλές λίστες, μια αυτόνομη λίστα απεικόνισης διακρίνεται από τις υπόλοιπες βάσει ενός ακεραίου αριθμού που παίζει το ρόλο του “αναγνωριστικού αριθμού” της (list identifier).

Προεκτιμένου να αποδώσουμε αναγνωριστικούς αριθμούς σε λίστες απεικόνισης, πρέπει να δεσμεύσουμε το απαιτούμενο εύρος τιμών. Η δέσμευση αυτή γίνεται με την εντολή `glGenLists()`. `GLuint glGenLists(GLint range);`

Όπου range το πλήθος των αναγνωριστικών που θέλουμε να χρησιμοποιήσουμε. Η συνάρτηση επιστρέφει μια ακέραιη τιμή που αντιστοιχεί στον πρώτο αναγνωριστικό αριθμό.

Π.χ. με τη σύνταξη:

```
listID=glGenLists(2);
```

παράγουμε 2 identifiers. Ο πρώτος έχει ακέραιη τιμή listID, ο δεύτερος listID+1 και ούτω καθ' εξής.

Η ανάθεση αναγνωριστικής τιμής σε μια λίστα απεικόνισης γίνεται κατά την έναρξη της δήλωσής της στην εντολή `glNewList`:

```
void glNewList(GLuint listID, GLenum listMode);
```

Π.χ. σύνταξη λίστας απεικόνισης:

```
glNewList();
```

```
// Εντολές δήλωσης σχήματος
```

```
glEndList();
```

Μεταξύ των εντολών `glNewList` και `glEndList` ορίζουμε το σύνθετο γεωμετρικό σχήμα της λίστας απεικόνισης, χρησιμοποιώντας τις εντολές σχεδίασης σχημάτων που αναφέραμε παραπάνω.

**74. Έστω κώδικας: `<?php $a = "6"; $b = 8; echo $b - $a; ?>` Ποιο θα είναι το αποτέλεσμα; Δικαιολογήστε την απάντησή σας.**

Το αποτέλεσμα θα είναι 2. Η `php` δεν κάνει διάκριση μεταξύ των τύπων των μεταβλητών, οπότε η παραπάνω πράξη εκτελείται κανονικά

**75. Ποιες είναι οι διαφορές μεταξύ των πινάκων της PHP και των πινάκων της C++;**

Στην PHP μπορούμε να δημιουργήσουμε έναν πίνακα με δύο τρόπους, είτε με τη συνάρτηση `array` ή με τους ειδικούς τελεστές `[ ]`. Επίσης στην PHP:

α) Ένας πίνακας (`array`) είναι μια κανονική μεταβλητή της PHP η οποία έχει την ιδιότητα ότι περιέχει άλλες μεταβλητές μέσα της.

β) Η κάθε μεταβλητή που περιέχεται μέσα σ' έναν πίνακα είναι γνωστή ως στοιχείο (`element`) και το κάθε στοιχείο ενός πίνακα περιέχει ένα κλειδί (`key`) και μια τιμή (`value`), που μπορεί να είναι μια άλλη μεταβλητή.

**76. Περιγράψτε τα στάδια επεξεργασίας μιας PHP σελίδας από τη στιγμή της αίτησής της μέχρι την εμφάνιση στον browser του χρήστη.**

Μια σελίδα PHP περνά από επεξεργασία από συμβατό web server (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. `*.php`, `*.php4`, `*.phtml` κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης `.html` δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση `.php`, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται κώδικα PHP.

**77. Αναφέρετε τους κανόνες που αφορούν στην ονομασία των μεταβλητών της PHP;**

Μεταβλητές στην PHP:

- Ξεκινούν με `$`
- Ακολουθεί γράμμα ή underscore (`_`)
- Μπορεί να περιέχει γράμματα, αριθμούς, underscores (`_`) ή dashes (`\`, `/`)
- Δεν μπορεί να περιέχει κενά
- Είναι case sensitive

**78. Η εντολή \$c = 1; echo ' \$c' . \$c; τι θα δώσει ως τύπωση;**

Ο κώδικας θα εμφανίσει : \$c1

**79. Έστω κώδικας:**

```
<?php $a = "Hello, $b = "world!"; echo $a; echo $b; ?>
```

**Ποιό θα είναι το αποτέλεσμα; Δικαιολογήστε την απάντησή σας.**

Ο κώδικας θα εμφανίσει : Hello, world!

**80. Έστω κώδικας php:**

```
$a = 2; echo $a; ++$a; echo $a; --$a; echo $a;
```

**Ποιό θα είναι το αποτέλεσμα; Δικαιολογήστε την απάντησή σας.**

Ο κώδικας θα εμφανίσει : 2 3 2

**81. Να γραφεί ένα πρόγραμμα PHP το οποίο θα εμφανίζει με αλλαγή γραμμής το μήνυμα "Καλωσήρθατε στην PHP" 5 φορές.**

```
<?php
for ($i=1; $i<=5; $i++)
{
    echo "Καλωσηρθατε στην PHP </br>";
}
?>
```

**82. Ποιοί είναι οι βασικοί τύποι σχέσεων μεταξύ πινάκων στην PHP(MySql);**

Υπάρχουν τρεις (3) τύποι σχέσεων μεταξύ πινάκων :

- Σχέση ένα προς ένα (one to one): μια εγγραφή (γραμμή) ενός πίνακα σχετίζεται με μία μόνο εγγραφή ενός άλλου πίνακα και το αντίστροφο.
- Σχέση ένα προς πολλά (one to many): μία εγγραφή (γραμμή) ενός πίνακα σχετίζεται με πολλές εγγραφές ενός άλλου πίνακα, αλλά όχι το αντίστροφο.
- Σχέση πολλά προς πολλά (many to many): μία εγγραφή (γραμμή) ενός πίνακα σχετίζεται με πολλές εγγραφές ενός άλλου πίνακα και το αντίστροφο.

**83. Να γραφεί μία φόρμα σε HTML με δύο πλαίσια κειμένου με τα συμβολικά ονόματα text1 και text2 και ένα πλήκτρο πίεσεως submit το οποίο θα ενεργοποιεί μία σελίδα σε PHP η οποία θα εμφανίζει τις τιμές των πλαισίων κειμένου με πλάγια γράμματα.**

**Form.html**

```
<html>
<body>
<form action="text.php" method="post">
Text1: <input type="text" name="mytext1" />
Text2: <input type="text" name="mytext2" />
```



```
Submit:<input type="submit" value ="Submit" />
</form>
</body>
</html>
```

**text.php**

```
<html>
<body>
<?php
$mytext1= $_POST["mytext1"];
$mytext2= $_POST["mytext2"];
?>
<i> <?php echo $mytext1 . $mytext2; ?></i>
</body>
</html>
```

- 84. Να γραφεί μία φόρμα η οποία θα δέχεται από τον χρήστη ένα όνομα, ένα επίθετο και έναν αριθμητικό κωδικό. Αν κάποιος από τα πεδία δεν συμπληρωθεί από τον χρήστη τότε θα πρέπει να επανεμφανίζεται η φόρμα με ένα κόκκινο μήνυμα που να γράφει ποια πεδία πρέπει να συμπληρωθούν. Η φόρμα ενεργοποιεί μία σελίδα PHP που εκτυπώνει τα πεδία της φόρμας.**

```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<?php
// define variables and set to empty values
$firstnameErr = $lastnameErr = $pswdErr = "";
$firstname = $lastname = $pswd = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["firstname"])) {
        $firstnameErr = "FirstName is required";
    } else {
        $firstname = test_input($_POST["firstname"]);
    }

    if (empty($_POST["lastname"])) {
        $lastnameErr = "LastName is required";
    } else {
```

```
$lastname = test_input($_POST["lastname"]);
}

if (empty($_POST["pswd"])) {
    $pswdErr = "Password is required";
} else {
    $pswd = test_input($_POST["pswd"]);
}

}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>

<h2>PHP Form Validation Example</h2>
<p><span class="error">* required field.</span></p>
<form                                method="post"                                action="<?php
echohtmlspecialchars($_SERVER["PHP_SELF"]);?>">
    FirstName: <input type="text" name="firstname">
    <span class="error">* <?php echo $firstnameErr;?></span>
    <br><br>
    LastName: <input type="text" name="lastname">
    <span class="error">* <?php echo $lastnameErr;?></span>
    <br><br>
    Password: <input type=" password " name="pswd">
    <span class="error"><?php echo $pswdErr;?></span>
    <br><br>
    <input type="submit" name="submit" value="Submit">
</form>

<?php
echo "<h2>Your Input:</h2>";
echo $firstname;
echo "<br>";
echo $lastname;
echo "<br>";
echo $pswd;
?>
</body>
</html>
```

85. Σε ένα αρχείο PHP υπάρχουν οι μεταβλητές var1,var2, Ex1 και Ex2 με διάφορες τιμές. Να γραφεί μία φόρμα που θα έχει ένα πλαίσιο κειμένου και θα ενεργοποιεί το παραπάνω αρχείο και το οποίο θα εμφανίζει την τιμή της μεταβλητής που έχει όνομα ίδιο με αυτό που είναι η τιμή του πλαισίου κειμένου.

#### Form.html

```
<html>
<body>
<form action="text.php" method="post">
Text1: <input type="text" name="mytext1" />
Submit:<input type="submit" value ="Submit" />
</form>
</body>
</html>
```

#### text.php

```
<html>
<body>
<?php
$var1 = value1;
$var2 = value2;
$Ex1 = value3;
$Ex2 = value4;

$x= $_POST["mytext1"];
if ($x=$var1){
echo $x;
}
else if ($x=$var2){
echo $x;
}
else if ($x=$Ex1){
echo $x;
}
else if ($x=$Ex2){
echo $x;
}
?>
</body>
</html>
```

**86. Ποιες είναι οι βασικές διαφορές μεταξύ PHP και JAVASCRIPT;**

Η javascript είναι μια Client-side γλώσσα προγραμματισμού ενώ η PHP μία Server-side γλώσσα προγραμματισμού. Έτσι οι διαφορές τους μπορούμε να πούμε ότι συνοψίζονται στον παρακάτω πίνακα

Client-side	Server-side
Εκτελείται στον τοπικό υπολογιστή	Εκτελείται στο server
Επηρεάζεται από παραμέτρους του τοπικού περιβάλλοντος πχ. Javascript engine version	Έχει σταθερή συμπεριφορά για όλους τους clients
Τα αποτελέσματα εμφανίζονται αμέσως στον χρήστη	Υπάρχει καθυστέρηση μέχρι να μεταδοθούν τα αποτελέσματα στον χρήστη
Ο φόρτος που δημιουργείται από ένα script δεν επηρεάζει άλλους clients	Όταν υπάρχει φόρτος καθυστερούν όλοι οι clients
Δεν είναι ασφαλής για καταχώρηση ευαίσθητων δεδομένων – ο υπολογιστής του χρήστη θεωρείται πάντα εκτεθειμένος	Αν στηθεί σωστά μπορεί να αποθηκεύσει ευαίσθητα δεδομένα με ασφάλεια

**87. Πώς μπορούμε να έχουμε καθολικές μεταβλητές μεταξύ πολλών συναρτήσεων; Δώστε από ένα παράδειγμα σε κάθε περίπτωση.**

Στην PHP οι καθολικές μεταβλητές πρέπει να δηλωθούν σαν global μέσα σε μια συνάρτηση αν πρόκειται να τις χρησιμοποιήσουμε μέσα σ' αυτή τη συνάρτηση. Ακολουθεί ένα παράδειγμα :

```
$a = 1;
$b = 2;
Function Sum () {
    global $a, $b;
    $b = $a + $b;
}
Sum ();
echo $b;
```

Το παραπάνω script θα εμφανίσει το 3. Δηλώνοντας τις \$a και \$b σαν global μέσα στη συνάρτηση, όλες οι αναφορές και στις δύο μεταβλητές θα αφορούν τις καθολικές τιμές. Δεν υπάρχει κάποιος περιορισμός στον αριθμό των καθολικών μεταβλητών που μπορεί να χειριστεί μια συνάρτηση.

Ένας δεύτερος τρόπος για να έχουμε πρόσβαση σε μεταβλητές σε καθολική εμβέλεια είναι να χρησιμοποιήσουμε τον ειδικό πίνακα που ορίζεται στην PHP με όνομα **\$GLOBALS**. Έτσι, το προηγούμενο παράδειγμα θα μπορεί να ξαναγραφεί ως εξής :

```

$a = 1;
$b = 2;
Function Sum () {
    $GLOBALS["b"] = $GLOBALS["a"] + $GLOBALS["b"];
}
Sum ();
echo $b;

```

Ο πίνακας `$GLOBALS` είναι ένας associative πίνακας με το όνομα της καθολικής μεταβλητής να αποτελεί το key και τα περιεχόμενα αυτής της μεταβλητής να αποτελούν την τιμή του στοιχείου του πίνακα.

**88. Γιατί δεν μπορούμε να εκτελέσουμε ένα αρχείο PHP απλά ανοίγοντάς το με έναν διαχειριστή αρχείων όπως γίνεται με τις HTML σελίδες; Δικαιολογήστε την απάντησης σας.**

Η γλώσσα προγραμματισμού PHP είναι γλώσσα Server – side (εκτελείται στον server) οπότε δε μπορεί να εκτελεστεί με έναν διαχειριστή αρχείων όπως η HTML η οποία είναι Client – side γλώσσα προγραμματισμού.

**89. Πώς θα έπρεπε να δημιουργήσετε έναν βρόχο με την εντολή while για την εκτύπωση όλων των περιττών αριθμών από το 1 μέχρι το 49;**

```

<?php
$i=1;
while($i<=49)
{
    echo $i . "<br />";
    $i+=2;
}
?>

```

**90. Πώς θα έπρεπε να χρησιμοποιήσετε μία εντολή if για να εκτυπώσετε το αλφαριθμητικό "Youth message" στο παράθυρο μιας εφαρμογής browser εάν η τιμή μιας ακέραιης μεταβλητής ( της \$age) είναι μεταξύ 18 και 35; Εάν η \$age περιέχει οποιαδήποτε άλλη τιμή θα πρέπει να εκτυπώνεται το αλφαριθμητικό "Generic message" στο παράθυρο της εφαρμογής browser.**

```

if ($age>18 && $age<35) {
    echo "Youth message";
}
else {
    echo "Generic message";
}

```

- 91. Να υλοποιήσετε μια ιστοσελίδα η οποία να περιέχει κώδικα HTML και PHP script, βάσει των οποίων θα εμφανίζει φόρμα στην οποία ο χρήστης θα δίνει δυο αριθμούς, θα επιλέγει την πράξη (πρόσθεση ή αφαίρεση) και πατώντας στο Execute θα εμφανίζεται το κατάλληλο αποτέλεσμα.**

```
<head>
<meta charset="utf-8">
<title>Calculator</title>
</head>
<body>
<form method="post" attribute="post" action="disp_form.php">
<p>First Value:<br/>
<input type="text" id="first" name="first"></p>
<p>Second Value:<br/>
<input type="text" id="second" name="second"></p>
<input type="radio" name="group1" id="add" value="add"
checked="true"><p>+</p><br/>
<input type="radio" name="group1" id="subtract" value="subtract"><p>-</p><br/>
<input type="radio" name="group1" id="times" value="times"><p>x</p><br/>
<input type="radio" name="group1" id="divide" value="divide"><p>/</p><br/>
<p></p>
<button type="submit" name="answer" id="answer" value="answer">Execute</button>
</form>
</body>
</html>
```

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Answer</title>
</head>
<body>
<p>The answer is:
<?php
$first = $_POST['first'];
$second= $_POST['second'];
if($_POST['group1'] == 'add') {
echo "$first + $second";
}
else if($_POST['group1'] == 'subtract') {
echo "$first - $second";
}
else if($_POST['group1'] == 'times') {
echo "$first * $second";
}
else($_POST['group1'] == 'divide') {
echo "$first / $second";
}
?>
</p>
</body>
</html>
```

- 92. Να υλοποιηθεί ένας απλός υπολογισμός για το εμβαδόν ενός κύκλου χρησιμοποιώντας τον τύπο  $E = \pi * r^2$ . Η ακτίνα θα δίνεται μέσω φόρμας από τον χρήστη και το PHP script θα κάνει τον υπολογισμό του εμβαδού. Ο υπολογισμός του εμβαδού θα γίνεται υποχρεωτικά από κατάλληλη συνάρτηση (function) την οποία θα ορίσετε στο PHP script. (Το  $\pi$  ισούται με 3,14).**

```
<html>
<body>
<form action="emvado.php" method="post">
Text1: <input type="text" name="aktina" />
Submit:<input type="submit" value ="Submit" />
</form>
</body>
</html>
```

### emvado.php

```
<html>
<body>
<?php
function emvado($r)
{
$E=3.14*$r^2;
return $E;
}
$r= $_POST["aktina"];
echo emvado($r);
?>
</body>
</html>
```

- 93. Διατυπώστε τον κώδικα που χρειάζεται η PHP για να συνδεθεί στον (My)SQL DataBase Server με πρόβλεψη εκτύπωσης τυχόν σφάλματος ή επιτυχούς σύνδεσης. Χρησιμοποιήστε το username "admin" και password "fail".**

```
<?php
$conn = mysql_connect("localhost","admin","fail");
if (!$conn)
{
die('Η σύνδεση απέτυχε: ' . mysql_error());
}
// Κώδικας εδώ
?>
```



**94. Πού χρησιμοποιούμε τις μεταβλητές συνόδου στην PHP (PHP session variables); Δώστε δύο παραδείγματα.**

Τα sessions είναι ένας μηχανισμός με τον οποίο μπορείτε να ελέγξετε την ταυτότητα του επισκέπτη όταν αυτός περιηγείται στις σελίδες ενός ιστότοπου. Με την php μπορείτε να δημιουργείτε sessions για τους επισκέπτες και έτσι να ελέγχετε την ταυτότητά τους και κατά συνέπεια την πρόσβαση αυτών σε ευαίσθητες σελίδες. Τα session χρησιμοποιούν την τεχνολογία των cookies. Δηλαδή κάθε επισκέπτης δέχεται ένα cookie στο μηχανήμά του το οποίο περιέχει έναν αριθμό (id) το οποίο είναι και το αναγνωριστικό του. Ελέγχοντας εσείς το αναγνωριστικό του επισκέπτη μπορείτε να δείτε αν ο επισκέπτης είναι σε session ή όχι.

Παράδειγμα

Εστω η σελίδα first.php η οποία θα πρέπει να αρχίζει με τις δύο παρακάτω γραμμές.

```
<?php
session_start();
$_SESSION['visitor']=session_id();
?>
```

Για να μπορείτε να ελέγχετε τον επισκέπτη στη σελίδα second.php θα πρέπει να υπάρχει ο παρακάτω κώδικας.

```
<?php
session_start();
if(isset($_SESSION['visitor']) && $_SESSION['visitor']==session_id())
{
echo "welcome";
}
else
{
echo "sorry not in session";
}
?>
```

**95. Δημιουργήστε στην PHP τη συνάρτηση addNums() η οποία δέχεται ως ορίσματα 2 ακέραιους και επιστρέφει το αποτέλεσμα της πρόσθεσης αυτών στην οθόνη.**

```
<?php
function addNums($x,$y)
{
$sum=$x + $y;
return $sum;
}
?>
```

96. Φτιάξτε ένα script στην PHP που θα διαβάζει την τιμή της μεταβλητής συνόδου (session) "logged" και εάν αυτή δεν είναι αληθής να κάνει ανακατεύθυνση (redirect) στην σελίδα login.html.

```
<?php
session_start();
$_SESSION['logged']=session_id();
?>

<?php
session_start();
if(isset($_SESSION['logged']) && $_SESSION['logged']==session_id())
{
echo "welcome";
}
else
{
echo "<a href=login.html>Αρχική Σελίδα!</a>";
}
?>
```

97. Δημιουργήστε στην PHP τη συνάρτηση sum() η οποία δέχεται ως ορίσματα 2 ακέραιους και επιστρέφει το αποτέλεσμα της πρόσθεσης αυτών στην οθόνη. Εκτελέστε το σε παράδειγμα χρησιμοποιώντας τους παρακάτω ακέραιους αριθμούς:α)  $5 + 10 = 15$ β)  $7 + 13 = 20$ γ)  $2 + 4 = 6$

```
<?php
function sum($x,$y)
{
$s=$x + $y;
return $s;
}
?>
```

98. Στείλτε ένα e-mail σε όλες τις εγγραφές του πίνακα subscribers χρησιμοποιώντας τη συνάρτηση mail (to, subject, body). Το email θα έχει θέμα "Login info" και περιεχόμενο "Your username is <το username χρήστη> and your password is <το password χρήστη> ". Δεν απαιτείται ο κώδικας σύνδεσης με τον (My)Sql server. Ο πίνακας subscribers έχει τα πεδία id, username, password, email.

```
<?php
$result = mysql_query("SELECT * FROM subscribers");
while($row = mysql_fetch_array($result))
{
$to = $row['email'];
$subject = "Login info";
```

```
$body = "Your username is $row['username'] and your password is  
$row['password']";  
$sent = mail($to, $subject, $body);  
}  
>
```

**99. Τι εννοούμε με τον όρο static mesh στην αρχιτεκτονική των Game Engines (Unreal Engine, Source και Unity);**

Με τον όρο Static Mesh εννοούμε τα πολυγωνικά πλέγματα που συνιστούν ένα σημαντικό μέρος της αρχιτεκτονικής των χαρτών σε πολλές μηχανές παιχνιδιών, συμπεριλαμβανομένων της Unreal Engine και της Unity. Ο όρος "static", αναφέρεται στο γεγονός ότι τα δεν μπορούν να κινηθούν ανά κορυφή του πλέγματος, δεδομένου ότι μπορούν να κινηθούν κλιμακωθούν και να ξαναγίνουν skin σε πραγματικό χρόνο.

**100. Τι είναι τα collision boxes στα Game Engines; Τι εξυπηρετούν και με ποιο σκεπτικό τα κατασκευάζουμε;**

Ίσως από τα πιο σημαντικά στοιχεία ενός video game είναι τα events που συμβαίνουν με διάδραση στα αντικείμενα, όπως με τη σύγκρουση μαζί τους ή με την είσοδο σε μια συγκεκριμένη περιοχή. Τέτοια events μπορούν να χρησιμοποιηθούν σε πληθώρα εφαρμογών, όπως, για παράδειγμα, ένα σύστημα μάχης, ή ένα σύστημα που πυροδοτεί διαφορετικά event με την είσοδο του παίκτη σε έναν χώρο. Για ενεργοποίηση των παραπάνω events χρησιμοποιούμε collision boxes.

Τόσο οι colliders όσο και τα triggers αποτελούν περιοχές στο χώρο (2D ή 3D) οι οποίες μπορεί να εξυπηρετούν διάφορους σκοπούς. Η κύρια διαφορά τους είναι πως οι colliders εμποδίζουν στοιχεία με άλλους colliders να προσπελάσουν τη περιοχή και πρακτικά αποτελούν «αόρατους τοίχους», ενώ τα triggers είναι αόρατες περιοχές στις οποίες οποιοδήποτε αντικείμενο έχει πρόσβαση. Συνεπώς, οι colliders υπάρχουν για να προσομοιώνουν τη σύγκρουση δυο φυσικών αντικειμένων (και να μεταφέρουν οποιαδήποτε πληροφορία αφορά τη σύγκρουση αυτή), ενώ τα triggers αφορούν στη μετάδοση πληροφοριών για τα αντικείμενα που εισέρχονται, μένουν ή εξέρχονται από αυτά. Σημειώνεται πως προκειμένου τα triggers να αναγνωρίσουν ένα αντικείμενο, αυτό το αντικείμενο θα πρέπει να έχει collider.

**101. Αναφέρετε ποιες τεχνικές φωτισμού χρησιμοποιούνται στα video games και ποια τα χαρακτηριστικά τους.**

Ο φωτισμός είναι ο σημαντικότερος παράγοντας στην τεχνολογία των γραφικών καθώς από αυτό καθορίζεται το πόσο ρεαλιστικά θα απεικονίζεται η γεωμετρία ενός αντικειμένου. Στην σημερινή εποχή το επίπεδο ρεαλισμού έχει φτάσει στο σημείο να μην μπορούμε να διακρίνουμε τη διαφορά μεταξύ των γραφικών και της πραγματικότητας. Κάποιοι από τους αλγόριθμους που χρησιμοποιούνται κυρίως για την παραγωγή οπτικών εφέ σε ταινίες είναι οι: ray tracing, global illumination, final gather,

το μεγαλύτερο μειονέκτημα των οποίων είναι ο χρόνος υπολογισμού (rendering). Οι κάρτες γραφικών δεν μπορούν να εκτελέσουν αυτούς τους αλγόριθμους σε πραγματικό χρόνο λόγω των απαιτήσεων τους σε υπολογιστική ισχύ, με αποτέλεσμα στα παιχνίδια να εφαρμόζεται η τεχνική του χάρτη φωτισμού (Lightmap). Ο χάρτης φωτισμού είναι μία δομή δεδομένων που περιέχει πληροφορίες για σκιές και φωτεινότητα επιφανειών και είναι στατικός, δηλαδή οι σκιές δε μεταβάλλονται κατά τη διάρκεια του παιχνιδιού. Για τον φωτισμό στα video games χρησιμοποιούνται τρία στοιχεία ανάλογα με το είδος του φωτισμού που χρειαζόμαστε.

Directional light: Χρησιμοποιείται για φωτισμό μεγάλων περιοχών π.χ.ολόκληρης πίστας.

Point Light και lightmaps: Χρησιμοποιούνται για φωτισμό μικρότερων χώρων π.χ. σε καταφύγια. Υπάρχει η δυνατότητα δημιουργίας lightmaps, δηλαδή περισσότερα από ένα στοιχεία Point Light μπορούν να ομαδοποιηθούν και να έχουν κοινά χαρακτηριστικά π.χ. ένταση φωτισμού, για να πετύχουμε καλύτερα και πιο ομοιόμορφα φωτισμένες περιοχές.

Spot light: Μοιάζουν πολύ με τα Point Light με την διαφορά όμως ότι η εστίασή τους περιορίζεται σε μια συγκεκριμένη περιοχή π.χ. σαν φωτισμός που αναπαράγεται από μία λάμπα.

## **102. Τι είναι το bill boarding και ποια τα πλεονεκτήματά του στην κατασκευή 3D games;**

Το Bill boarding που είναι μία τεχνική απόδοσης αντικειμένων που βασίζονται σε εικόνες, ώστε να φαίνονται πιο ρεαλιστικές και ζωντανές. Εφαρμόζεται συνήθως στην οπτικοποίηση μοντέλων τοπίων, λόγω της πολυπλοκότητας αυτών των τόπων αντικειμένων, όπως για παράδειγμα των δέντρων. Η τεχνική Bill boarding αντικαθιστά τα αντικείμενα αυτά με μία εικόνα που τοποθετείται σε ένα διαφανές τετράπλευρο σχήμα. Καθώς ο θεατής κινείται στη σκηνή το τετράπλευρο περιστρέφεται, έτσι ώστε να φαίνεται πάντα η μπροστινή του μεριά.

## **103. Πολλές φορές στους κατασκευαστές παιχνιδιών (πχ Unity) εφαρμόζεται το Level of Detail (LOD), περιγράψτε τι είναι και ποια είναι η χρησιμότητά του.**

Επίπεδο λεπτομέρειας (LOD-Level of Detail) συνεχόμενο ή προκαθορισμένο. Το χαρακτηριστικό αυτό αφορά την τρισδιάστατη απεικόνιση του θέματος σε πραγματικό χρόνο και σχετίζεται με την ποσότητα των πολύγωνων που περιγράφουν το θέμα (λεπτομέρεια), κάθε φορά που αυτό αποδίδεται γραφικά στην οθόνη του Η/Υ, ανάλογα με την απόσταση του αντικειμένου από τη θέση της ιδεατής κάμερας και κατά συνέπεια της επιφάνειας που αυτό καταλαμβάνει στην οθόνη του Η/Υ. Έτσι, όσο πιο μικρή είναι η επιφάνεια που καταλαμβάνει το θέμα στην οθόνη του υπολογιστή, τόσο λιγότερη είναι η οπτική πληροφορία που μπορεί να αποδοθεί σε αυτή και συνεπώς τόσο πιο απαρατήρητη είναι η χρήση λιγότερων πολύγωνων για την οπτική απόδοσή του θέματος. Η τεχνική αυτή έχει σαν αποτέλεσμα την ταχύτερη οπτική απόδοση της τρισδιάστατης σκηνής και εφαρμόζεται σε περιπτώσεις ιδεατών περιπάτων.

**104. Να αναφερθούν συνοπτικά οι βασικές κλάσεις του Unreal Editor και ποια η λογική της χρησιμοποίησής τους.**

Οι βασικότερες κλάσεις από τις οποίες όλες οι υπόλοιπες παράγονται είναι οι εξής:

**Object** – Είναι η γονική κλάση όλων των κλάσεων στην Unreal. Όλες οι συναρτήσεις στην κλάση Object είναι προσπελάσιμες από παντού γιατί όλες οι κλάσεις παράγονται από αυτή. Η Object είναι μια αφηρημένη κλάση, υπό την έννοια ότι δεν κάνει τίποτα χρήσιμο. Όλη η λειτουργικότητα προέρχεται από τις υπό-κλάσεις, όμως σε αυτή βρίσκονται δεκάδες μεταβλητές, δομές και συναρτήσεις που είναι χρήσιμες σε όλες τις υπόλοιπες κλάσεις.

**Actor** – Προέρχεται από την κλάση Object και είναι η γονική κλάση όλων των αυτόνομων αντικειμένων στην Unreal. Η κλάση Actor περιέχει όλη την λειτουργικότητα που χρειάζεται ένα αντικείμενο (actor) για να κινηθεί, να αλληλεπιδράσει με άλλα αντικείμενα, να επηρεάσει το περιβάλλον και να κάνει όλα τα χρήσιμα σχετικά με το παιχνίδι πράγματα.

**Pawn** – Προέρχεται από την κλάση Actor και είναι η γονική κλάση όλων των παιχτών και πλασμάτων, τα οποία είναι ικανά για υψηλού επιπέδου τεχνητή νοημοσύνη ή έλεγχο από τον παίχτη.

**Class** – Προέρχεται από την κλάση Object και είναι ένα ειδικού είδους αντικείμενο, το οποίο περιγράφει την κλάση ενός αντικειμένου. Αν και φαίνεται αρχικά κάπως μπερδεμένο, μια κλάση είναι ένα αντικείμενο και η κλάση περιγράφει ορισμένα αντικείμενα. Για παράδειγμα, όταν δημιουργούμε ένα νέο αντικείμενο (actor), μπορούμε να ορίσουμε την κλάση του αντικειμένου χρησιμοποιώντας τον τύπο Class.

**105. Περιγράψτε τη κλάση actor (στην UnrealScript), αναφέρατε τη χρησιμότητά της.**

Προέρχεται από την κλάση Object και είναι η γονική κλάση όλων των αυτόνομων αντικειμένων στην Unreal. Η κλάση Actor περιέχει όλη την λειτουργικότητα που χρειάζεται ένα αντικείμενο (actor) για να κινηθεί, να αλληλεπιδράσει με άλλα αντικείμενα, να επηρεάσει το περιβάλλον και να κάνει όλα τα χρήσιμα σχετικά με το παιχνίδι πράγματα.

**106. Περιγράψτε τη κλάση pawn (στην UnrealScript), αναφέρατε τη χρησιμότητά της.**

Προέρχεται από την κλάση Actor και είναι η γονική κλάση όλων των παιχτών και πλασμάτων, τα οποία είναι ικανά για υψηλού επιπέδου τεχνητή νοημοσύνη ή έλεγχο από τον παίχτη.

**107. Να αναφερθούν οι τύποι μεταβλητών που υποστηρίζονται από την Unrealscript καθώς και οι περιγραφές τους.**

Οι τύποι μεταβλητών που υποστηρίζονται από την Unrealscript είναι οι εξής:

- byte: Μια τιμή ενός byte που κυμαίνεται από 0 ως 255
- int: Ένας 32-bit ακέραιος αριθμός
- bool: Μια λογική τιμή. True/false
- float: Ένας 32-bit αριθμός κινητής υποδιαστολής
- string: Μια σειρά χαρακτήρων
- constant: Μια μεταβλητή που δεν μπορεί να τροποποιηθεί
- array<Type>: Ένας πίνακας του τύπου “Type”
- enumeration: Μια μεταβλητή που μπορεί να πάρει μια σειρά από ακέραιες τιμές προκαθορισμένου ονόματος.
- struct: Όμοια με τις δομές στην γλώσσα C, οι δομές στην Unrealscript επιτρέπουν την δημιουργία νέων τύπων μεταβλητών, που περιέχουν άλλες μεταβλητές. Για παράδειγμα, δυο πολύ χρησιμοποιούμενες δομές στην Unreal είναι οι δομές vector (περιέχει τις μεταβλητές X,Y,Z) και rotator.
- name: Χρησιμοποιείται για την αποθήκευση του ονόματος ενός αντικειμένου στην Unreal, όπως το όνομα μιας συνάρτησης, κατάστασης (state), κλάσης, κτλ. Αποθηκεύονται σε έναν καθολικό πίνακα ως δείκτες για ταχύτερη προσπέλαση.
- object and Actor references: Μεταβλητή που αναφέρεται σε ένα άλλο αντικείμενο στον τρισδιάστατο χώρο. Μέσω αυτών των μεταβλητών μπορούμε να προσπελάσουμε τις μεταβλητές ενός άλλου αντικειμένου. Μπορούν να πάρουν την ειδική τιμή “none”, που είναι αντίστοιχη με τιμή “null” στην C.
- delegate: Τα delegates είναι αναφορές σε κάποια συνάρτηση μέσα σε ένα στιγμιότυπο ενός αντικειμένου. Πρόκειται για τον συνδυασμό δυο προγραμματιστικών ιδεών, των συναρτήσεων και των μεταβλητών. Οι μεταβλητές διατηρούν κάποιες τιμές που μπορούν κατά τη διάρκεια της εκτέλεσης του προγράμματος να αλλάξουν. Υπό κάποια έννοια τα delegates είναι μεταβλητές που κρατούν κάποιες τιμές, μόνο που αυτές οι τιμές είναι άλλες συναρτήσεις μέσα στην κλάση. Επίσης, εκτελούνται σαν συναρτήσεις και για αυτό είναι ένα τόσο ισχυρό εργαλείο στον προγραμματισμό παιχνιδιών

**108. Η UnrealScript δίνει αρκετή λειτουργικότητα για τα όπλα, να αναφερθούν αναλυτικά οι βασικές καταστάσεις στις οποίες μπορεί να περιέλθει ένα όπλο με τη χρήση της.**

OnStartedFiring () - που ονομάζεται όταν το όπλο εισέρχεται στην κατάσταση πυροδότησης.

OnStoppedFiring () - που ονομάζεται όταν το όπλο βγαίνει από την κατάσταση πυροδότησης.

Το OnContinuedFiring () - που ονομάζεται όταν πυροδοτείται το όπλο, ολοκληρώνει την καθυστέρηση ανανέωσης(refire delay) και η σκανδάλη για τη λειτουργία πυρκαγιάς(fire mode) είναι ακόμα κάτω, ώστε το όπλο να πυροδοτήσει και πάλι.

**109. Τι σημαίνουν οι όροι Modeling, Rendering και Animation;**

Η μοντελοποίηση (Modeling) είναι η δημιουργία της γεωμετρίας ενός αντικειμένου. Ως γεωμετρία ενός 3D αντικειμένου ορίζεται μια συλλογή σημείων (vertices), στον 3D χώρο, συνδεδεμένα με διάφορες γεωμετρικές οντότητες όπως καμπύλες, τρίγωνα, ευθύγραμμα τμήματα, κλπ. Η σχεδιοκίνηση (Animation) είναι η μετατόπιση της γεωμετρίας σε σημαντικά χρονικά σημεία, σε συγκεκριμένες θέσεις του κόσμου, με στόχο την απόδοση κίνησης και την οπτικοποίηση της χρονικής μεταβολής. Η φωτοαπόδοση (Rendering) συμπληρώνει τη γεωμετρία που έχουμε δημιουργήσει με χρώματα, σκιές, φωτισμό και πολλά άλλα οπτικά εφέ όπως τα ατμοσφαιρικά εφέ. Στην ουσία το Rendering είναι η μέθοδος με την οποία παίρνουμε τις σκηνές που έχουμε δημιουργήσει στον υπολογιστή και τις παρουσιάζουμε στον κόσμο.

**110. Τι είναι φωτοαπόδοση και ποιοι παράγοντες την επηρεάζουν;**

Το τελευταίο στάδιο της ψηφιακής αναπαράστασης μίας σκηνής του πραγματικού κόσμου είναι η φωτοαπόδοση της ή αλλιώς Rendering. Το rendering συμπληρώνει τη γεωμετρία που έχουμε δημιουργήσει με χρώματα, σκιές, φωτισμό και πολλά άλλα οπτικά εφέ όπως τα ατμοσφαιρικά εφέ. Στην ουσία το Rendering είναι η μέθοδος με την οποία παίρνουμε τις σκηνές που έχουμε δημιουργήσει στον υπολογιστή και τις παρουσιάζουμε στον κόσμο. Μέσα από τις ρυθμίσεις του μηχανισμού φωτοαπόδοσης που θα χρησιμοποιήσουμε, ορίζουμε το είδος και το μέγεθός του αρχείου στο οποίο θα φωτοαποδοθεί η σκηνή αλλά και το είδος των εφέ (ατμοσφαιρικά, εικόνες φόντου) που θα προσθέσουμε στο τελικό αποτέλεσμα. Η φωτοαπόδοση μπορεί να δημιουργήσει είτε μία δισδιάστατη στατική εικόνα είτε κινούμενη εικόνα (animation) βασισμένη στο 3D μοντέλο μας χρησιμοποιώντας τον φωτισμό και τα υλικά που έχουμε προσθέσει στην σκηνή μας κατά τα δύο προηγούμενα στάδια. Οι παράγοντες που την επηρεάζουν είναι υπολογιστική ισχύ, τα χρώματα, οι υφές (textures), τα υλικά (materials), ο φωτισμός και οι σκιάσεις.

**111. Αναφέρετε τα στάδια δημιουργίας κινούμενης φωτορεαλιστικής απεικόνισης ενός θέματος πολυμέσων με τη χρήση κάποιου εργαλείου τρισδιάστατης σχεδίασης και κίνησης (π.χ. 3d Studio).**

Η δημιουργία κινούμενης φωτορεαλιστικής απεικόνισης χωρίζεται σε τρία βασικά στάδια: Τη μοντελοποίηση (modeling), την απόδοση σχεδιοκίνησης (Animation) και τη φωτορεαλιστική απεικόνιση (rendering). Η μοντελοποίηση είναι η δημιουργία της γεωμετρίας ενός αντικειμένου. Ως γεωμετρία ενός 3D αντικειμένου ορίζεται μια συλλογή σημείων (vertices), στον 3D χώρο, συνδεδεμένα με διάφορες γεωμετρικές οντότητες όπως καμπύλες, τρίγωνα, ευθύγραμμα τμήματα, κλπ. Η απόδοση σχεδιοκίνησης είναι η μετατόπιση της γεωμετρίας σε σημαντικά χρονικά σημεία, σε συγκεκριμένες θέσεις του κόσμου, με στόχο την απόδοση κίνησης και την οπτικοποίηση της χρονικής μεταβολής. Η φωτορεαλιστική απεικόνιση είναι η διαδικασία ρεαλιστικής απόδοσης των



χαρακτηριστικών ενός μοντέλου. Αυτό γίνεται με τη χρήση κατάλληλων χρωμάτων, υφών (textures), υλικών (materials), φωτισμού και σκιάσεων.

**112. Τι είναι οι τεχνικές φωτισμού (lighting) και σκίασης (shading); Αναφέρετε τις βασικές τεχνικές φωτισμού και σκίασης που χρησιμοποιούνται σε μια εφαρμογή τρισδιάστατης σχεδίασης και κίνησης.**

**Ο φωτισμός (lighting)** είναι πολύ σημαντικός για την απεικόνιση του φωτός, χρώματος και σκιάς ενός τρισδιάστατου μοντέλου, μιας σκηνής ή μιας εικόνας και άλλοτε χρησιμεύει στην φωτορεαλιστική απόδοση τρισδιάστατης σχεδίασης και σε άλλες περιπτώσεις η κατεύθυνση, η φωτεινότητα και το χρώμα του φωτός διαμορφώνονται κατάλληλα ώστε να επιτευχθεί κάποιο εφέ (π.χ. ηλιοβασίλεμα).

**Τεχνικές φωτισμού:**

**Point/Omni Light:** Εκπομπή φωτός από ένα συγκεκριμένο μικρό σημείο στον τρισδιάστατο χώρο προς όλες τις κατευθύνσεις.

**Κατευθυντικό φως (Directional Light):** Ένα κατευθυντικό φως αντιπροσωπεύει μια μακρινή πηγή φωτός (όπως ο ήλιος ή το φεγγάρι) και οι ακτίνες του τρέχουν παράλληλα προς μία κατεύθυνση από κάθε σημείο του ουρανού και χρησιμοποιούνται συνήθως για την προσομοίωση του άμεσου ηλιακού φωτός.

**Φώς σημείων (Spot Light):** Ένα σημειακό φως εκπέμπει ένα φωτεινό πεδίο με σχήμα κώνου από ένα μόνο σημείο στο διάστημα.

**Φως χώρου (Area Light):** Είναι ένα φως περιοχής που εκπέμπει κατευθυντικές ακτίνες μέσα από ένα καθορισμένο σχήμα (είτε ορθογώνιο είτε κυκλικό)

**Φως όγκου (Volume Light):** Είναι ένα σημείο φωτός (Spot Light) που εκπέμπει ακτίνες από ένα κεντρικό σημείο και έχει ένα συγκεκριμένο σχήμα και μέγεθος (κύβος, σφαίρα, κύλινδρος κ.λπ.).

**Φωτισμός περιβάλλοντος(Ambient Light):** Ένα φως περιβάλλοντος εκπέμπει ακτίνες φωτός προς κάθε κατεύθυνση ανεβάζοντας το συνολικό επιπέδου διάχυτου φωτισμού σε μια σκηνή.

**Η σκίαση (shading)** είναι η τεχνική κατά την οποία, με την προσθήκη πηγή φωτός, δημιουργούμε την αντίληψη βάθους σε τρισδιάστατα μοντέλα ή εικονογραφήσεις καθώς και φωτορεαλισμό ανάλογα με την ένταση της σκιάς. Επιτυγχάνεται είτε με προσθήκη πηγή φωτός είτε με προσομοίωση σκίασης επιπλέον σχεδιασμένων στοιχείων.

### 113. Ποιες τεχνικές γνωρίζετε για την απόδοση υλικών (materials & textures) πάνω στα τρισδιάστατα μοντέλα;

Είναι μια μέθοδος για τον καθορισμό, υφής επιφάνειας ή πληροφοριών χρώματος σε ένα τρισδιάστατο μοντέλο. Έτσι μία επίπεδη εικόνα τυλίγεται γύρω από το τρισδιάστατο αντικείμενο και δημιουργεί μια υφή σε μια 3D επιφάνεια. Η μέθοδος αυτή είναι μια σύνθετη διαδικασία καθώς εμπεριέχει πολλαπλές ανεξάρτητες ιδιότητες (όπως ύψος, τραχύτητα, μετατόπιση, ανάκλασης, κατοπτρική αντανάκλαση πολλές άλλες παραλλαγές)

Η δημιουργία αυτών των ανεξάρτητων υλικών(εικόνων) μπορούν να σχεδιαστούν από το εκάστοτε σχεδιαστικό πρόγραμμα εξαρχής, είτε από μια σειρά φωτογραφιών επιφάνειας από το αντικείμενο ή παρόμοιο στον πραγματικό κόσμο, είτε σχεδιάζοντας εξαρχής δισδιάστατες εικόνες (photoshop).

Σκοπός η φωτορεαλιστική απόδοση του τρισδιάστατου μοντέλου.

### 114. Τι είναι το Normal Mapping και ποια η χρησιμότητά του; Περιγράψτε αναλυτικά τον τρόπο λειτουργίας του.

**Normal Map** είναι η εικόνα που χρησιμοποιείται για να δημιουργηθεί η ψευδαίσθηση του ανάγλυφου. Το φως διαχέεται διαφορετικά στην επιφάνεια απεικόνισης σύμφωνα με το normal map το οποίο αποτελείται από χρώματα rgb που δηλώνουν την διεύθυνση της γεωμετρίας της επιφάνειας. Στο Normal Mapping, έχουμε ολική αντικατάσταση του διανύσματος με προϋπολογισμένα κανονικά διανύσματα. Οι χάρτες κανονικών διανυσμάτων, περιέχουν αναπαραστάσεις των κανονικών διανυσμάτων. Όπως τα κανονικά διανύσματα χρησιμοποιούνται για τον υπολογισμό του φωτισμού και των ανακλάσεων πάνω στην γεωμετρία του αντικειμένου, τα Normal Maps μπορούν να χρησιμοποιηθούν για τον προσδιορισμό της γωνίας με την οποία πέφτει το φως στις διάφορες επιφάνειες της γωνίας που χρησιμοποιείται για τους υπολογισμούς της ανάκλασης. Για την αναπαράσταση των κανονικών διανυσμάτων, γίνεται χρήση εικόνων 24-bit, όπου τα διαφορετικά χρώματα αντιπροσωπεύουν διαφορετικές κατευθύνσεις. Συγκεκριμένα, αναλογία του RGB μοντέλου με τα κανονικά διανύσματα είναι εξής: το κόκκινο χρώμα αναπαριστά τον άξονα x, το πράσινο αναπαριστά τον άξονα y και το μπλε αναπαριστά τον άξονα z.

### 115. Τι είναι οι τεχνικές σχεδίασης και φωτισμού; Περιγράψτε περιληπτικά τρεις τεχνικές δημιουργίας της τελικής εικόνας (rendering).

Rendering είναι η τελική υπολογιστική διαδικασία προσθήκης φωτισμού, σκίασης, χρώματος και υφής σε 2D ή 3D σχέδια, προκειμένου να πραγματοποιηθεί σύνθεση εικόνων που μοιάζουν πραγματικές.

#### **Global illumination rendering (GI):**

Στην περίπτωση αυτή προσδίδεται πιο ρεαλιστικός φωτισμός σε τρισδιάστατες σκηνές λαμβάνοντας υπόψη όχι μόνο το φως που προέρχεται απευθείας από μια πηγή φωτός αλλά και αυτές οι οποίες αντανακλώνται από άλλες επιφάνειες της σκηνής, είτε αντανακλαστικές είτε όχι.

**Ambient occlusion rendering (AO):**

Στην περίπτωση αυτή προσομοιώνονται απαλές σκιές που εμφανίζονται στις ρωγμές και τα επιμέρους στοιχεία των 3D αντικειμένων καθορίζοντας τον όγκο του και διαχωρίζοντας αυτό και τα μεταξύ τους αντικείμενα στη τρισδιάστατη σκηνή προσδίδοντας ένα άλλο επίπεδο ρεαλισμού. Δεν λαμβάνονται υπόψη πηγές φωτός και αντανάκλασεις παρά μόνον ο φωτισμός περιβάλλοντος (Ambient Light).

**Raytraced rendering:**

Με την τεχνική αυτή γίνεται ανίχνευση της διαδρομής των ακτινών του φωτός για τη δημιουργία εικόνων ως εικονοστοιχεία (pixels) σε διαφορετικά επίπεδα. Με το σύνολο αυτών, δημιουργούνται εικονικά αντικείμενα. Η τεχνική αυτή αποδίδει υψηλότερο οπτικό ρεαλισμό από τις άλλες μεθόδους rendering, αλλά με μεγαλύτερο υπολογιστικό κόστος.

**116. Γιατί είναι απαραίτητος ο φωτισμός της σκηνής; Περιγράψτε τρεις τρόπους με τους οποίους μπορεί να επιτευχθεί το παραπάνω.**

Ο ακριβής φωτισμός μιας σκηνής είναι αυτός που θα αποδώσει το ποσοστό του τελικού φωτορεαλισμού μιας εικόνας (render) ή και το επιθυμητό οπτικό αποτέλεσμα.

Σύνθετος φωτισμός είναι αυτός των **τριών σημείων** η οποία χρησιμοποιεί τρία φώτα που ονομάζονται βασικό φως (key light), φως γεμίσματος (fill light) και το οπίσθιο φως (back light).

Η χρησιμοποίηση **μιας πηγής φωτός (Directional Light )** προσομοιώνοντας το φως του ήλου.

**HDRI lighting** είναι η τεχνική κατά την οποία δημιουργούμε ένα θόλο (dome) γύρω από τη τρισδιάστατη σκηνή ή αντικείμενο προσδίδοντας υλικό μια εικόνα περιβάλλοντος (π.χ .ουρανός) με αποτέλεσμα η σκηνή να φωτιστεί , χρωματιστεί και να αντανάκλα ανάλογα με την εικόνα αυτή.

**117. Τι είναι τα function curves (στα προγράμματα τρισδιάστατης απεικόνισης) και που χρησιμοποιούν;**

Οι καμπύλες λειτουργιών (function curves) είναι επεξεργάσιμες καμπύλες (σφήνες) που αντιπροσωπεύουν τις τιμές κινούμενων εικόνων σε μορφή γραφήματος γραμμής. Παρέχουν έναν πολύτιμο τρόπο τόσο για την οπτικοποίηση όσο και για την επεξεργασία των κομματιών κινούμενων εικόνων. Με την εμφάνιση της καμπύλης λειτουργίας, μπορούμε πραγματικά να δούμε τα χαρακτηριστικά της κινούμενης εικόνας καθώς αλλάζουν με την πάροδο του χρόνου. Η απότομη καμπύλη υποδεικνύει την ταχύτητα ενός αντικειμένου στη σκηνή. Αν η καμπύλη είναι βαθιά, το αντικείμενο επιταχύνει. Εάν η καμπύλη ισιώνει, το αντικείμενο επιβραδύνει.

**118. Τι είναι το UV Mapping;**

Αφού έχουμε δημιουργήσει το βασικό πλέγμα του μοντέλου μας πρέπει ύστερα να δημιουργήσουμε μια δισδιάστατη αναπαράσταση του. Αυτό γίνεται με μια διαδικασία που λέγεται UV-χαρτογράφηση (UV-mapping). Τα αρχικά της "U" και "V" στην ουσία καθορίζουν τις μεταβλητές των δύο αξόνων που προβάλλεται η εικόνα μιας και οι μεταβλητές "X", "Y" και "Z" είναι ήδη δεσμευμένα για τις

αναπαράσταση των σημείων που αποτελούν το τρισδιάστατο μοντέλο. Αυτή η δισδιάστατη αναπαράσταση θα χρησιμοποιηθεί στην συνέχεια για να δοθεί υφή(texture) στο μοντέλο μας αλλά και για άλλες οπτικές του βελτιώσεις.

#### **119. Τι είναι το Diffuse Map και ποια η διαφορά του από το Texture Map;**

Το diffuse δίνει την αντανάκλαση ή διάχυση του φωτός και του χρώματος σε πολλές γωνίες. Συνήθως μία τιμή για όλο το αντικείμενο είναι αρκετή, αλλά μπορεί να χρησιμοποιηθεί map για να σκουρύνουν κάποια σημεία του αντικειμένου. Η ανάκλαση αυτή δεν είναι γυαλιστερή.

#### **120. Τι είναι το Bump Map;**

Είναι υφές που αποθηκεύουν μια "ένταση", το σχετικό ύψος των pixel από το σημείο θέασης της κάμερας. Τα pixel φαίνεται ότι κινούνται από την απαιτούμενη απόσταση στην κατεύθυνση της επιφάνειας. Μπορείτε να χρησιμοποιήσετε είτε εικόνες στη κλίμακα του γκρι είτε την ένταση των τιμών από την Υφή-RGB (συμπεριλαμβανομένου και εικόνων).

#### **121. Τι είναι το Opacity Map;**

Ο Opacity map/χάρτης υφής αδιαφάνειας καθορίζει την αδιαφάνεια του υλικού με βάση την ένταση του χρώματος της απεικόνισης. Επιλέγοντας μία εικόνα ως χάρτη αδιαφάνειας μπορούμε να καταστήσουμε ένα αντικείμενο μερικώς ή πλήρως διαφανές. Οι τιμές της κλίμακας του γκρι σε αυτόν τον χάρτη υφής καθορίζουν την αδιαφάνεια του υλικού, Οι τιμές του λευκού δημιουργούν απόλυτη αδιαφάνεια ενώ οι τιμές του μαύρου προσφέρουν απόλυτη διαφάνεια και διατηρούν τη μορφή του texture.

#### **122. Τι είναι η τεχνική extrude και πώς μπορεί να αξιοποιηθεί;**

Η τεχνική Extrude μπορεί να ανυψώσει ένα πολυγωνικό αντικείμενο κάθετα προς την επιφάνειά του, όπως, για παράδειγμα, η δημιουργία ενός κυλίνδρου ο οποίος προκύπτει από την ανύψωση ενός κύκλου. Το εργαλείο Extrude Modifier δίνει ύψος σε αντικείμενα δύο διαστάσεων και τα κάνει τρισδιάστατα. Το πρόγραμμα υψώνει το αντικείμενο κάθετα ως προς την βάση του, δημιουργεί ένα αντίγραφο του spline στην καθορισμένη απόσταση και μετά συνενώνει τα δύο splines.

#### **123. Στην τεχνική του Box Modeling, ποια η διαφορά μεταξύ του Convert to editable poly και τότε του convert to editable mesh; Πότε χρησιμοποιείται το δεύτερο;**

Τα αντικείμενα editable mesh αποτελούνται από συνδεδεμένα τριγωνικά faces, ενώ τα editable poly αντικείμενα αποτελούνται από πολύγωνα με περισσότερες πλευρές. Χρησιμοποιούμε Convert to editable poly όταν θέλουμε να επεξεργαστούμε κάποιο μοντέλο σε επίπεδο sub-object. Όταν έχουμε τελειώσει με το μοντέλο και θέλουμε να ελευθερώσουμε μνήμη χρησιμοποιούμε convert to editable mesh.

**124. Τι είναι το Specular Map;**

Ένας χάρτης Specular καθορίζει τη φωτεινότητα των αντανakλάσεων στο αντικείμενο. Μπορούμε να εισάγουμε ασπρόμαυρη bitmap εικόνα (alpha channel). Στα λευκά σημεία κατά την απόδοση θα φαίνονται οι αντανakλάσεις, θα υπάρχει δηλαδή πλήρη όψη γυαλάδας, ενώ στα μαύρα δεν θα συμβεί τίποτα, ενώ οι ενδιάμεσες τιμές μειώνουν το μέγεθος ενός φωτεινού τόνου.

**125. Τι είναι το modifier Lathe; Τι μπορεί να σχεδιαστεί χρησιμοποιώντας το;**

Ο τροποποιητής τόνου – Lathe δημιουργεί ένα αντικείμενο πλέγματος περιστρέφοντας μία καμπύλη spline γύρω από έναν άξονά της. Με τον τροποποιητή αυτό θα μπορούσε να δημιουργηθεί ένα μπουκάλι ή ένα βάζο.

**126. Τι είναι το Displacement Mapping;**

Οι Displacement maps/χάρτες μετατόπισης μετατοπίζουν τη γεωμετρία των επιφανειών. Το αποτέλεσμα τους είναι παρόμοιο με το αποτέλεσμα του modifier Displace . Σε αντίθεση με το Bump map ένας χάρτης μετατόπισης αλλάζει τη γεωμετρία της επιφάνειας. Κάνοντας εφαρμογή της κλίμακας του γκρι (grayscale) του χάρτη που χρησιμοποιείται παράγουν μετατόπιση. Σε μια δυσδιάστατη εικόνα τα φωτεινότερα χρώματα εξωθούν τη γεωμετρία της επιφάνειας πιο έντονα σε σχέση με τα σκοτεινότερα και ως αποτέλεσμα έχουν την τρισδιάστατη μετατόπιση της. Η ποσότητα μετατόπισης είναι το ποσοστό της διαγώνιου του κύβου που οριοθετεί το αντικείμενο και περιέχει τον χάρτη. Έτσι, επηρεάζει άμεσα όλες τις επιφάνειες του αντικειμένου. Για παράδειγμα, Εάν κάνουμε scale στο αντικείμενο, τότε αλλάζει μαζί και ο χάρτης μετατόπισης. Ένας χάρτης μετατόπισης μπορεί να εφαρμοστεί άμεσα σε: Bezier patches (επιφάνειες που έχουν υποστεί Bezier), Editable meshes (επεξεργάσιμα πλέγματα), Editable polymeshes (επεξεργάσιμα πολυγωνικά πλέγματα), NURBS surfaces (επιφάνειες NURBS). Για άλλου είδους γεωμετρικές μορφές, όπως τα primitives, τα extended primitives, τα compound objects κ.τ.λ., για να χρησιμοποιηθεί ο χάρτης μετατόπισης θα πρέπει πρώτα να εφαρμοστεί ο modifier Disp Approx (Displacement Approximation). Μόνο με αυτόν τον τρόπο θα υπάρξει μετατόπιση της επιφάνειας του αντικειμένου. Ο modifier Disp Approx εφαρμόζεται σε editable mesh αντικείμενα. Τέλος, οι χάρτες μετατόπισης δεν είναι ορατοί στα viewports εκτός εάν εφαρμοστεί κάποιος Modifier που θα τα εμφανίσει.

**127. Δημιουργείστε ένα παράθυρο 300X400 (window 300X400) σε JAVA, το οποίο θα μπορεί να περιλαμβάνει ένα υπό δημιουργία παιχνίδι.**

```
JFrame jframe = new JFrame("JFrame Size Example");
jframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
jframe.setPreferredSize(new Dimension(300, 400));
jframe.pack();
jframe.setLocationRelativeTo(null);
jframe.setVisible(true);
```

### 128. Περιγράψτε τη χρησιμότητα των μεταβλητών κατά τη δημιουργία παιχνιδιών σε κώδικα JAVA (JAVA Game Variables).

Κατά τη δημιουργία ενός παιχνιδιού σε κώδικα JAVA, χρησιμοποιούμε τις JAVA Game Variables για να «ξέρουμε» ποια είναι η τρέχουσα κατάσταση του παιχνιδιού. Αυτό μπορεί να περιλαμβάνει το σκορ του παίκτη, τον αριθμό των εχθρικών διαστημικών πλοίων που υπάρχουν ή το που βρίσκονται και τι κάνουν όλα τα αντικείμενα του παιχνιδιού στην οθόνη.

### 129. Να γραφεί απλό πρόγραμμα με κώδικα σε Java, που να τυπώνει ως αποτέλεσμα στον χρήστη τις λέξεις "I love JAVA Game Programming"

```
JFrame frame = new JFrame("I love JAVA Game Programming");
JPanel panel = new JPanel();
JLabel label = new JLabel("I love JAVA Game Programming");
panel.add(label);
frame.add(panel);
frame.setVisible(true);
```

### 130. Να αναφέρετε τη χρησιμότητα των μεθόδων (methods) στο JAVA Game Programming.

Οι χρησιμοποίηση Java μεθόδων στην ανάπτυξη JAVA Game λογισμικού κρίνεται σκόπιμη για τους εξής λόγους:

#### Επαναχρησιμοποιήσιμο κώδικα

Εάν πρέπει να κάνουμε το ίδιο πράγμα, ή σχεδόν το ίδιο πράγμα, πολλές φορές, γράφουμε μια μέθοδο για να το κάνει, και στη συνέχεια απλά καλούμε τη μέθοδο κάθε φορά που πρέπει να εκτελεστεί αυτή η εργασία.

#### Παραμετροποίηση τον κώδικα

Εκτός από τον επαναχρησιμοποιήσιμο κώδικα που είναι ο ίδιος σε όλες τις περιπτώσεις, θα θέλαμε συχνά να χρησιμοποιήσουμε παραμέτρους που αλλάζουν τον τρόπο με τον οποίο λειτουργεί η μέθοδος.

#### Top-Down programming

Ένα πολύ χρήσιμο στυλ προγραμματισμού ονομάζεται "top-down" προγραμματισμός. Αντιμετωπίζουμε ένα μεγάλο πρόβλημα (το "κορυφαίο"), διασπώντας το σε μικρά προβλήματα. Για να το κάνετε αυτό σε ένα πρόγραμμα, γράφετε μια μέθοδο για την επίλυση του μεγάλου προβλήματός σας καλώντας άλλες μεθόδους για να λύσετε τα μικρότερα τμήματα του προβλήματος (Διαίρει και Βασίλευε). Οι μέθοδοι για την επίλυση των απλούστερων προβλημάτων παρομοίως απαιτούν άλλες μεθόδους μέχρι να καταλήξετε σε απλές μεθόδους που λύουν απλά προβλήματα.

#### Δημιουργία Εννοιολογικών Μονάδων



Δημιουργούμε μεθόδους για να κάνουμε κάτι που είναι μια ενέργεια/διεργασία η οποία ενοσιολογικά επιλύει ένα τμήμα του προβλήματος, συνθέτοντας τμήμα της λύσης του.

### Απλοποίηση

Επειδή οι τοπικές μεταβλητές και δηλώσεις μιας μεθόδου δεν μπορούν να φανούν εκτός της μεθόδου, αυτές (και η πολυπλοκότητά τους) κρύβονται από άλλα μέρη του προγράμματος, γεγονός που αποτρέπει τυχαία σφάλματα ή σύγχυση.

**131. Ένα παιχνίδι που έχει δημιουργηθεί σε JAVA, συνήθως περιέχει:1) Frame2) Canvas3) GameLoop4) Rendering method5) Update method6) Mouse and Key input Εξηγείστε συνοπτικά τις έννοιες που αναφέρθηκαν, καθώς και τη χρησιμότητά τους.**

#### **a) Frame**

Ένα frame είναι ένα παράθυρο ανώτατου επιπέδου με τίτλο και περίγραμμα. Το μέγεθος του frame περιλαμβάνει οποιαδήποτε περιοχή που έχει οριστεί για τα σύνορα. Οι διαστάσεις της περιοχής των συνόρων μπορούν να ληφθούν χρησιμοποιώντας τη μέθοδο `getInsets`, ωστόσο, εφόσον αυτές οι διαστάσεις εξαρτώνται από την πλατφόρμα, δεν είναι δυνατή η επίτευξη μιας έγκυρης τιμής εισόδου έως ότου γίνει η εμφάνιση του frame είτε από το πακέτο κλήσης είτε από την εμφάνιση.

#### **b) Canvas**

Μια κλάση Canvas είναι παράγωγο ή υποκατηγορία της κλάσης Component και όταν τοποθετείται πάνω σε ένα Frame, εμφανίζεται ως κενή περιοχή. Ωστόσο, για το σχεδιασμό γραφικών, μπορεί να χρησιμοποιηθεί οποιαδήποτε άλλη κλάση που προέρχεται από την κλάση Component, για παράδειγμα JPanel ή ακόμα και JTextField ή JButton.

#### **c) GameLoop**

Ένα GameLoop τρέχει συνεχώς κατά τη διάρκεια του παιχνιδιού. Κάθε στροφή του βρόχου, επεξεργάζεται την είσοδο του χρήστη χωρίς να εμποδίζει, ενημερώνει την κατάσταση του παιχνιδιού και καθιστά το παιχνίδι. Τα GameLoop είναι το βασικό παράδειγμα ενός "παιχνιδιού προγραμματισμού". Κάθε επανάληψη μέσω του GameLoop προχωράει την κατάσταση του παιχνιδιού κατά κάποιο ποσό. Συγκεκριμένα, εάν μετρήσουμε πόσο γρήγορα ο κύκλος παιχνιδιών κυκλώνει σε πραγματικό χρόνο, παίρνουμε τα "frames ανά δευτερόλεπτο" του παιχνιδιού. Αν το GameLoop κυλάει γρήγορα, το FPS είναι υψηλό και το παιχνίδι κινείται ομαλά και γρήγορα.



#### d) Rendering

Υπάρχουν δύο διαφορετικά είδη rendering: το ενεργό (Active) και το παθητικό (Passive). Ο κώδικας σχεδίασης τοποθετείται σε μια μέθοδο χρώματος και ο κώδικας καλείται σε απάντηση στις αιτήσεις επανακαθορισμού. Αυτά μπορεί να προέρχονται από τον ίδιο τον κώδικα, αλλά μπορεί επίσης να προέρχονται από το λειτουργικό σύστημα ως απόκριση σε γεγονότα όπως αλλαγή μεγέθους ενός παραθύρου ή κλικ σε ένα στοιχείο.

Το Active Rendering αντί να αφήσει κάποιον άλλο να αποφασίσει πότε να ζωγραφίσει, το πρόγραμμα επαναλαμβάνει συνεχώς την οθόνη σε πολύ σφιχτό while loop. Ενώ αυτό το είδος rendering δεν συνιστάται για εφαρμογές γενικού σκοπού, είναι απαραίτητο για την ανάπτυξη video games.

#### e) Update

Στον εικονικό κόσμο του παιχνιδιού διατηρείται μια συλλογή αντικειμένων. Κάθε αντικείμενο εφαρμόζει μια μέθοδο update που προσομοιώνει ένα πλαίσιο της συμπεριφοράς του αντικειμένου. Σε κάθε frame, το παιχνίδι ενημερώνει κάθε αντικείμενο της συλλογής.

#### f) Mouse and Key input

Στα παιχνίδια υπολογιστή, το πληκτρολόγιο και το ποντίκι είναι οι κύριες μέθοδοι αλληλεπίδρασης με τον υπολογιστή. Το πρόβλημα είναι ότι, ενώ η Java έχει μεγάλη υποστήριξη για αυτές τις συσκευές εισόδου για εφαρμογές GUI, τα παιχνίδια υπολογιστών πρέπει να χειρίζονται την είσοδο λίγο διαφορετικά. Παρόλο που δεν υπάρχουν ενσωματωμένες κλάσεις που να μας δίνουν αυτό που χρειαζόμαστε, μπορούμε εύκολα να δημιουργήσουμε τις δικές μας. Ακόμη, call-back functions μπορούμε να λάβουμε τη διάδραση του χρήστη μέσω του ποντικιού ή του πληκτρολογίου (Mouse and Key input

) και να τα διαχειριζόμαστε τη στιγμή που προκύπτουν.

**132. Δημιουργείστε σε JAVA απλό κινούμενο αντικείμενο για το παιχνίδι σας (πχ μπαλάκι του τένις), ορίζοντας πρώτα την αρχική του θέση με x και y και κατόπιν τις επόμενες ανάλογα με την κίνηση που θέλετε να πάρει.**

```
package ball_move;

import javax.swing.JFrame;
import javax.swing.WindowConstants;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.util.Scanner;
import javax.swing.JFrame;
import javax.swing.JPanel;
```

```
@SuppressWarnings("serial")

public class Ball_Move extends JPanel{

    int x = 0;
    int y = 0;

    private void moveBall(int X, int Y) {
        x = X;
        y = Y;
    }

    @Override
    public void paint(Graphics g) {
        super.paint(g);
        Graphics2D g2d = (Graphics2D) g;
        g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
            RenderingHints.VALUE_ANTIALIAS_ON);
        g2d.fillOval(x, y, 30, 30);
    }

    public static void main(String[] args) throws InterruptedException {
        int X,Y;
        Scanner reader = new Scanner(System.in);
        JFrame frame = new JFrame("Sample Frame");
        Ball_Move game = new Ball_Move();
        frame.add(game);
        frame.setSize(300, 400);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        while(true) {
            System.out.print("X:");
            X= reader.nextInt();
            System.out.println();
            System.out.print("Y:");
            Y= reader.nextInt();
            System.out.println();
            game.moveBall(X,Y);
            game.repaint();
            Thread.sleep(10);
        }
    }
}
```

**133. Αναφέρατε τουλάχιστον τέσσερις τύπους πολυμέσων που μπορούν να εισαχθούν (import) σε μια ιστοσελίδα και εξηγήστε γιατί είναι απαραίτητη η εγκατάσταση ενός πρόσθετου προγράμματος αναπαραγωγής (add-ons) για ορισμένους τύπους πολυμέσων.**

Τα πολυμέσα στο διαδίκτυο είναι ήχος, μουσική, βίντεο, ταινίες και κινούμενα σχέδια. Τα πολυμέσα έρχονται σε πολλές διαφορετικές μορφές:

- a) Εικόνες,
- b) Μουσική,
- c) Ήχος,
- d) Βίντεο,
- e) Αρχεία,
- f) Ταινίες,
- g) Κινούμενα σχέδια και πολλά άλλα.

Οι ιστοσελίδες περιέχουν συχνά στοιχεία πολυμέσων διαφορετικών τύπων και μορφών. Στοιχεία πολυμέσων (όπως ήχου ή βίντεο) αποθηκεύονται σε αρχεία πολυμέσων. Ο πιο συνηθισμένος τρόπος για να ανακαλύψουμε τον τύπο ενός αρχείου είναι να εξετάσετε την επέκταση αρχείου. Έτσι, τα αρχεία πολυμέσων έχουν μορφές και διαφορετικές επεκτάσεις όπως: .swf, .wav, .mp3, .mp4, .mpg, .wmv και .avi.

Τα add-ons είναι απλά μικρά προγράμματα που επεκτείνουν τη λειτουργικότητα ενός προγράμματος περιήγησης. Χρησιμοποιώντας τα add-ons, μπορούμε να κάνουμε το πρόγραμμα περιήγησής να χειριστεί μια ποικιλία εργασιών που χωρίς αυτά τα εργαλεία δεν θα μπορούσε να εκτελέσει. Ένα add-on είναι στην ουσία ένα στοιχείο λογισμικού που προσθέτει μια συγκεκριμένη λειτουργία σε ένα υπάρχον πρόγραμμα υπολογιστή. Τα συνηθισμένα παραδείγματα είναι τα πρόσθετα που χρησιμοποιούνται στα προγράμματα περιήγησης ιστού για την προσθήκη νέων λειτουργιών, όπως μηχανές αναζήτησης, σαρωτές ιών ή η δυνατότητα χρήσης ενός νέου τύπου αρχείου, όπως μια νέα μορφή βίντεο. Οι δημοφιλέστερες προσθήκες για ένα πρόγραμμα περιήγησης είναι το Adobe Flash Player, το QuickTime Player και το plug-in Java, τα οποία μπορούν να εκκινήσουν μια εφαρμογή Java που έχει ενεργοποιηθεί από τον χρήστη σε μια ιστοσελίδα για την εκτέλεση σε μια τοπική εικονική μηχανή Java.

Οι περιηγητές υποστηρίζουν add-ons για πολλούς λόγους. Μερικοί από τους κύριους λόγους περιλαμβάνουν:

- για να επιτρέψει σε third-party developers να δημιουργήσουν δυνατότητες που επεκτείνουν μια εφαρμογή
- για να υποστηριχθεί εύκολα την προσθήκη νέων δυνατοτήτων
- για να μειωθεί το μέγεθος μιας εφαρμογής
- να διαχωριστεί ο πηγαίος κώδικας από μια εφαρμογή λόγω μη συμβατών αδειών λογισμικού.

**134. Τι είναι τα keywords, γιατί είναι απαραίτητη η χρησιμοποίησή τους στην δημιουργία μιας ιστοσελίδας και με ποιον τρόπο γίνεται η σωστή επιλογή τους;**

Τα **keywords** είναι οι λέξεις κλειδιά που θα εισάγει κάποιος στις μηχανές αναζήτησης για να βρει αυτό που χρειάζεται. Είναι απαραίτητη η χρησιμοποίησή τους για καλύτερα αποτελέσματα στις μηχανές αναζήτησης ώστε η ιστοσελίδα μας να έχει στοχευμένη αναζήτηση βάση των keywords και να καλύπτει όλους τους κανόνες του σωστού SEO (Search Engine Optimization). Η σωστή επιλογή των keywords γίνεται με μια σειρά από εργαλεία που υπάρχουν δωρεάν στο διαδίκτυο αλλά και με μεγάλη έρευνα στις μηχανές αναζήτησης.

**135. Τι εννοούμε με τον όρο Search Engine Optimization (SEO); Για ποιον λόγο μια ιστοσελίδα θα πρέπει να εφαρμόζει όλους τους κανόνες που εισάγει η επιστήμη του SEO;**

**SEO (search engine optimization)** είναι, η διαδικασία βελτιστοποίησης μιας ιστοσελίδας με σκοπό την κατάταξη της σε υψηλότερες θέσεις, κατά την παρουσίαση αποτελεσμάτων έρευνας μέσω μηχανών αναζήτησης. Ο κυριότερος λόγος που πρέπει μια ιστοσελίδα να εφαρμόζει όλους τους κανόνες είναι για να εμφανίζεται η ιστοσελίδα μας όσο γίνεται υψηλότερα, στα αποτελέσματα που φέρνουν οι μηχανές αναζήτησης. Όσο υψηλότερη θέση κατέχουμε στις μηχανές αναζήτησης, τόσο περισσότερη επισκεψιμότητα θα έχουμε στην ιστοσελίδα μας.

**136. Τι είναι οι εφαρμογές CMS (Content Management systems); Αναφέρατε δύο τουλάχιστον εργαλεία CMS.**

Τα **Συστήματα Διαχείρισης Περιεχομένου (Content Management Systems, CMS)** είναι διαδικτυακές εφαρμογές που επιτρέπουν την online τροποποίηση του περιεχομένου ενός δικτυακού τόπου. Οι διαχειριστές μέσω του διαδικτύου ενημερώνουν το περιεχόμενο στο Σύστημα Διαχείρισης Περιεχομένου, το οποίο είναι εγκατεστημένο σ' ένα διακομιστή. Οι αλλαγές αυτές γίνονται αυτόματα διαθέσιμες πάλι μέσω του διαδικτύου, σε όλους τους επισκέπτες και χρήστες του δικτυακού τόπου. Μερικά εργαλεία CMS είναι: Wordpress, Joomla, Drupal.

**137. Ποια η χρησιμότητα του RSS και πώς επιτυγχάνεται η ενσωμάτωσή του σε μια ιστοσελίδα τύπου CMS;**

Πρόκειται για μια τεχνολογία του Διαδικτύου, η οποία επιτρέπει σε εκατομμύρια χρήστες παγκοσμίως, που επιθυμούν να παρακολουθούν τα αγαπημένα τους websites και blogs να λαμβάνουν ενημερώσεις και φρέσκο περιεχόμενο την στιγμή που δημοσιεύεται κάνοντας απλώς μια εγγραφή στο συγκεκριμένο ιστότοπο από τον οποίο θέλουν να λαμβάνουν ειδοποιήσεις. Η ενσωμάτωση του RSS σε μια ιστοσελίδα τύπου CMS γίνεται απλά με την εγκατάσταση ενός μικρού μέρους λογισμικού(plugin) που έχει δημιουργηθεί για την συγκεκριμένη λειτουργία.

**138. Πώς ελέγχουμε την επισκεψιμότητα μιας ιστοσελίδας που δημιουργήθηκε με CMS;**

Την **επισκεψιμότητα** μιας ιστοσελίδας μπορούμε να την ελέγξουμε απλά κατεβάζοντας και εγκαθιστώντας ένα plugin που έχει δημιουργηθεί για αυτόν ακριβώς το σκοπό. Θα πρέπει πρώτα να

έχουμε κάνει έναν λογαριασμό στο Google Analytics έτσι ώστε μόλις κάνουμε εγκατάσταση το plugin να δώσουμε τα απαραίτητα στοιχεία και η google να μας στείλει ένα κομμάτι κώδικα που θα πρέπει να τον τοποθετήσουμε στο αντίστοιχο πεδίο στο plugin.

### 139. Αναφέρετε τις διαφορές μεταξύ component, module και plugin.

**Plugin:** δίνουν τη δυνατότητα να εκτελέσετε τον κώδικα ως απόκριση σε ορισμένα γεγονότα που ενεργοποιούνται/ προκαλούνται από το δικό σας κώδικα.

**Components:** είναι οι βασικές λειτουργικές μονάδες που παρουσιάζονται σε μια ιστοσελίδα π.χ άρθρα, φόρμες επικοινωνίας, web links κλπ. Παρουσιάζονται συνήθως στο κέντρο του βασικού περιεχομένου ενός δείγματος / πρότυπου (ανάλογα με το πρότυπο).

**Module:** είναι μια πιο ελαφριά και ευέλικτη επέκταση που χρησιμοποιείται για την απόδοση των σελίδων. Τα module χρησιμοποιούνται για μικρά κομμάτια της σελίδας που είναι γενικά λιγότερο περίπλοκα και μπορούν να φανούν σε διαφορετικά κομμάτια / μέρη.

### 140. Τι κάνουν τα εργαλεία τύπου filezilla?

**FTP (FileTransferProtocol – Πρωτόκολλο μεταφοράς αρχείων)** είναι το πρωτόκολλο που χρησιμοποιείται για την απομακρυσμένη μεταφορά αρχείων από έναν υπολογιστή σε έναν server και αντίστροφα μέσω ενός client όπως είναι το FileZilla.

### 141. Ποιοι τύποι χρηστών υπάρχουν σε μια ιστοσελίδα και ποια τυπικά δικαιώματα έχει ο καθένας;

Οι τύποι χρηστών που υπάρχουν σε μια ιστοσελίδα είναι:

- Ο **διαχειριστής (administrator)**: έχει τον πλήρη έλεγχο όλου του website και φυσικά όλα τα δικαιώματα όπως: την επεξεργασία του περιεχομένου της σελίδας, το ανέβασμα αρχείων, τη δημιουργία νέων σελίδων, να προσθέτει και να αφαιρεί δικαιώματα από τους άλλους χρήστες, αντιγραφή - μετονομασία - διαγραφή μιας σελίδας κλπ.
- Ο **συντάκτης (editor)**: τα δικαιώματα που του ανήκουν είναι η δημιουργία - επεξεργασία - δημοσιοποίηση - διαγραφή οποιουδήποτε άρθρου ή σελίδας, όπως επίσης και την διαχείριση των κατηγοριών, των links και των tags.
- Ο **συγγραφέας (author)**: τα δικαιώματα που του ανήκουν είναι η δημιουργία - επεξεργασία - δημοσιοποίηση - διαγραφή των άρθρων τα οποία έχει συντάξει και επίσης μπορεί να ανεβάζει αρχεία και εικόνες. Τέλος μπορεί να επεργαστεί σχόλια που έχουν γίνει στα δικά του άρθρα (posts).
- Ο **συνεργάτης (contributor)**: τα δικαιώματα που του ανήκουν είναι η δημιουργία και η επεξεργασία των άρθρων που έχει συντάξει ο ίδιος αλλά δεν έχει το δικαίωμα να τα δημοσιοποιήσει. Για να γίνει η δημοσιοποίηση πρέπει να δώσει έγκριση και να το κάνει

publish ο ίδιος ο administrator. Μόλις γίνει αυτό, ο contributor χάνει το δικαίωμα της επεξεργασίας του συγκεκριμένου post.

- **Ο συνδρομητής (subscriber):** μπορεί μόνο να διαβάσει τα άρθρα στο site, να πλοηγηθεί, να κάνει σχόλια αλλά δεν έχει κανένα δικαίωμα στην επεξεργασία του περιεχομένου του site ή στις ρυθμίσεις του.

#### **142. Τι είναι το ψηφιακό Video; Αναφέρετε ονομαστικά τις γνωστότερες μεθόδους συμπίεσης και αντίστοιχους τύπους ψηφιακών αρχείων video.**

Το ψηφιακό video μπορεί να εκληφθεί ως ένα σύνολο στατικών εικόνων που εναλλάσσονται γρήγορα πολλές φορές το δευτερόλεπτο. Συνεπώς το video «κληρονομεί» τα στοιχεία των εικόνων που το αποτελούν (χρωματική και χωρική ανάλυση - color special resolution) και επιπλέον χαρακτηρίζεται από τη συχνότητα εναλλαγής τους (συχνότητα καρέ-frame rate). Το πρότυπο για την προβολή οποιοδήποτε είδους μη-κινηματογραφικού video είναι 30 καρέ το δευτερόλεπτο, ενώ για film είναι 24-30 καρέ το δευτερόλεπτο. Συνεπώς η πληροφορία video αποτελείται από 30 ή 24 εικόνες (ή καρέ) κάθε δευτερόλεπτο.

Η χρησιμοποίηση της μεθόδου συμπίεσης RLE σε εικόνες που είναι σκίτσα ή σχεδιαγράμματα, κυρίως με αποχρώσεις του γκρι, οδηγεί σε σημαντική μείωση του όγκου τους. Η RLE μέθοδος είναι πολύ αποτελεσματική για εικόνες με συνεχόμενες περιοχές εικονοστοιχείων που έχουν το ίδιο χρώμα, καθώς τα αντικαθιστά με ένα κωδικό και το πλήθος τους. Τα αρχεία αυτά είναι τα PCX αλλά και πολλές μορφές TIFF.

Η συμπίεση LZW εφαρμόζεται σε περισσότερο σύνθετες εικόνες, όπως π.χ. έγχρωμες φωτογραφίες. Εκεί, υπάρχει μεγάλη ποικιλία χρωμάτων, που όμως επαναλαμβάνονται πολλές φορές τα ίδια, άρα, όπως είδαμε στο κεφάλαιο 2.3, η κωδικοποίηση τους με λιγότερα bit φέρνει μεγάλα ποσοστά συμπίεσης. Τέτοια συμπίεση υπάρχει στα αρχεία GIF

Η κωδικοποίηση κωδικοποίηση JPEG (Joint Photographic Expert Group) βασίζεται στην εφαρμογή μιας σειράς πολύπλοκων μαθηματικών μετασχηματισμών στην εικόνα. Για τη συμπίεση ή την αποσυμπίεση από JPEG χρειάζεται αρκετή υπολογιστική ισχύς, κάτι που παλαιότερα ήταν ιδιαίτερα χρονοβόρα. Σήμερα, που υπάρχουν ισχυροί επεξεργαστές, η κωδικοποίηση JPEG δεν είναι πια πρόβλημα, και έχει το πλεονέκτημα της δυνατότητας επιλογής του ποσοστού συμπίεσης που θέλουμε να επιτύχουμε, με αντίστοιχη επιβάρυνση στην ποιότητα της εικόνας.

#### **143. Τι είναι Component video σήμα;**

Η σύνδεση Component χρησιμοποιεί κατά κανόνα τρία καλώδια (τρία βύσματα μορφής RCA – το ένα μεταφέρει την πληροφορία για την φωτεινότητα και τα άλλα δύο για την χρωματικότητα) για τη μεταφορά του σήματος της εικόνας και τη συναντούμε σε δύο βασικές παραλλαγές: την RGB και τη YUV. Στους υπολογιστές οι αναλογικές συνδέσεις εικόνας γίνονται με το πρότυπο RGBHV, το οποίο χρησιμοποιεί 5 καλώδια, από τα οποία τρία χρησιμοποιούνται για τη χρωματική πληροφορία (Red, Green, Blue) και δύο για σήματα συγχρονισμού.

Η σύνδεση Component Video είναι η μόνη αναλογικού τύπου που σας επιτρέπει να έχετε σήμα progressive (εφόσον τον υποστηρίζουν dvd και τηλεόραση) .

Η σύνδεση Component με καλής ποιότητας καλώδια και σωστή εγκατάσταση φτάνει άνετα τα 70 μέτρα χωρίς ανάγκη ενίσχυσης .Η εμπειρία πάντως δείχνει ότι μια καλή σύνδεση Component με πολύ καλά καλώδια στην πράξη δεν υπολείπεται σε τίποτα από τα DVI, HDMI



#### 144. Τι είναι Composite σήμα video;

Composite video (στα ελληνικά σύνθετο βίντεο) είναι μια μορφή σήματος αναλογικής εικόνας (και μόνο εικόνας) πριν αυτό το σήμα συνδυαστεί με ηχητικό σήμα και διαμορφωθεί σε σήμα μεταφοράς RF. Το Composite video συχνά συναντάται με τα αρχικά CVBS από το «Composite Video Blanking and Sync». Το Composite video μεταφέρει τα τυπικά συστήματα εικόνας, όπως NTSC, PAL και SECAM.

Είναι η σύνθεση (εξού και ονομάζεται σύνθετο video) τριών πηγών σημάτων εικόνας που ονομάζονται Y, U και V (μαζί αναφέρονται ως YUV) με παλμούς συγχρονισμού (sync). Το σήμα Y αντιπροσωπεύει την φωτεινότητα (luminance) της εικόνας με παλμούς συγχρονισμού (sync). Το σήμα Y από μόνο του θα μπορούσε να εμφανίσει μια μονόχρωμη εικόνα. Τα σήματα U και V αντιπροσωπεύουν την χροιά και τον κορεσμό (συνολικά chrominance) και μεταφέρουν τις πληροφορίες χρώματος. Το σήμα Y συνδυάζεται με τα σήματα UV και έχουμε το σύνθετο βίντεο που μεταφέρει την έγχρωμη εικόνα.

Οι περισσότερες καταναλωτικές αναλογικές συσκευές εικόνας ενσωματώνουν έξοδο composite video. Σε πολλές συσκευές υπάρχει και η δυνατότητα διαμόρφωσης του composite και σε RF (με εσωτερικό διαμορφωτή - modulator) ώστε να μπορούν να απεικονίσουν εικόνα και ήχο στις συχνότητες VHF ή UHF των τηλεοπτικών συσκευών. Η μεταφορά του composite video γίνεται συνήθως με ένα βύσμα RCA που για ευκολία αναγνώρισης από τα άλλα σήματα που μεταφέρουν τα RCA είναι κίτρινου χρώματος. (το κόκκινο βύσμα RCA μεταφέρει το δεξί και το άσπρο το αριστερό κανάλι ήχου). Σε επαγγελματικές συσκευές το composite video μεταφέρεται με ομοαξονικά καλώδια, βύσματα και υποδοχές BNC. Στην Ευρώπη, ή σύνδεση SCART χρησιμοποιείται συχνότερα από τις συνδέσεις RCA (και σε μικρότερο βαθμό η σύνδεση S-Video), και έτσι το σήμα RGB που μεταφέρεται με το SCART χρησιμοποιείται για συνδέσεις υπολογιστών, κονσόλες βιντεοπαιχνιδιών, και DVD players αντί για το composite video.

#### 145. Περιγράψτε τις διαφορές μεταξύ των σημάτων VGA και Βίντεο.

**VGA(Video Graphis Array).** Είναι σύνδεση αναλογικού τύπου, 15 ακροδεκτών, η οποία υπάρχει από την εποχή των οθονών CRT. Διαθέτει 3 ζεύγη αγωγών, ένα για κάθε βασικό χρώμα (κόκκινο, πράσινο, μπλέ), καθώς και επιπλέον αγωγούς για τα σήματα κατακόρυφου και οριζόντιου συγχρονισμού των πλαισίων της εικόνας. Σε κάθε ένα από τα ζευγάρια αγωγών που μεταφέρουν αυτά τα σήματα, υπάρχει ηλεκτρικό σήμα με τάση που αυξομειώνεται ανάλογα με την ένταση του αντίστοιχου χρώματος. Επειδή όμως οι οθόνες LCD λειτουργούν με ψηφιακό τρόπο, χρησιμοποιούν μετατροπέα αναλογικού σε ψηφιακό σήμα, προκειμένου να μπορέσουν να αξιοποιήσουν τα αναλογικά σήματα της υποδοχής VGA. Αντίστοιχα, το κύκλωμα γραφικών του υπολογιστή έχει ήδη μετατρέψει την ψηφιακή πληροφορία εικόνας σε αναλογική, όπως απαιτεί το πρότυπο VGA.

**Υποδοχή DVI-D (Digital Video Interface).** Είναι ψηφιακή σειριακή σύνδεση, 19 ακροδεκτών, η οποία κερδίζει συνεχώς έδαφος σε βάρος της VGA. Κι εδώ η οπτική πληροφορία μεταφέρεται από 3 ζεύγη αγωγών που μεταφέρουν τα βασικά χρώματα, όμως το ηλεκτρικό σήμα έχει δύο μόνο δυνατές στάθμες, φέροντας έτσι πληροφορία κωδικοποιημένη στο δυαδικό σύστημα. Σε κάθε ένα από τα τρία αυτά ζεύγη αγωγών χρώματος αντιστοιχεί ένας ζεύγος αγωγών με σήμα χρονισμού (clock). Το πλεονέκτημά της DVI-D είναι ότι δεν υπάρχουν απώλειες ποιότητας σε μετατροπές από αναλογικό σε ψηφιακό σήμα και αντίστροφα, καθώς η ροή πληροφορίας από το κύκλωμα γραφικών του υπολογιστή μέχρι το κύκλωμα οδήγησης των υγρών κρυστάλλων είναι καθαρά ψηφιακή.



**146. Ποια είναι τα βασικά μέρη του ψηφιακού βίντεο;**

VIDEO FRAME RATES. Είναι ο αριθμός των καρτέ που απαρτίζουν ένα βίντεο ώστε το ανθρώπινο μάτι να είναι ικανό να αντιληφθεί την κίνηση.

FRAME SIZE. Το μέγεθος των καρτέ σε ένα βίντεο καθορίζει το πλάτος και το μήκος της παραγωγής στην οθόνη οι παραγωγές οι οποίες θα εξαχθούν σε βιντεοκασέτα πρέπει να είναι δημιουργημένες σε ένα συγκεκριμένο μέγεθος καρτέ. Ενώ στο διαδίκτυο αυτό ποικίλει, όσο μεγαλύτερο είναι το μέγεθος των καρτέ τόσο ο φυλλομετρητής θα καθυστερήσει η φόρτωση στον υπολογιστή.

RGB ΧΡΩΜΑ ΚΑΙ ΒΑΘΟΣ BIT. Μια έγχρωμη εικόνα είναι συνδυασμένη από τρία βασικά χρώματα το κόκκινο, το πράσινο και το μπλε αυτά τα τρία χρώματα μας παρέχουν με εκατομμύρια διαφορετικούς συνδυασμούς.

ΣΥΜΠΙΕΣΗ. Όσο μεγάλο είναι το μέγεθος των καρτέ τόσο μεγαλύτερο είναι το βάθος των bit τόσο αυξάνονται τα καρτέ ανά δευτερόλεπτο το οποίο σημαίνει ότι είναι καλύτερη η ποιότητα του βίντεο.

**147. Σε τι διαφέρει το Video από το Animation και ποιο κοινό χαρακτηριστικό έχουν;**

Το animation είναι η τέχνη της σχεδίασης σκίτσων και στη συνέχεια η εμφάνισή τους σε μια σειρά καρτέ, ώστε να μοιάζει με ένα κινούμενο και ζωντανό πράγμα σε εμάς, ενώ ένα βίντεο είναι μια ηχογράφιση είτε ακίνητων είτε κινούμενων αντικειμένων. Έτσι, οι δύο τέχνες είναι χωρισμένοι από πόλους, αν και εξυπηρετούν τον ίδιο σκοπό, επιτρέποντας σε ένα άτομο να τους δει σαν κινηματογραφικές ταινίες. Οι ουσιαστικές διαφορές τους είναι:

- Ένα βίντεο δημιουργείται χρησιμοποιώντας βιντεοκάμερα, κινητό ή κινηματογραφική κάμερα και δεν απαιτείται προετοιμασία και μπορεί κανείς να χρησιμοποιήσει μόνο τη φωτογραφική μηχανή και να αρχίσει να φωτογραφίζει οποιοδήποτε αντικείμενο, ακίνητο ή κινούμενο με τη φωτογραφική μηχανή.

- Το animation δημιουργείται από έναν γελοιογράφο ή έναν καλλιτέχνη που σχεδιάζει μια σειρά απεικονίσεων σε διαφορετικές γωνίες που τροφοδοτούνται σε έναν υπολογιστή για να τις μετατρέψουν σε λειτουργία βίντεο που προσθέτει μουσική ή φωνές.

Η δημιουργία κινούμενων εικόνων είναι πιο δύσκολη από τη δημιουργία ενός βίντεο, αλλά μόλις μετατραπεί σε βίντεο. δεν υπάρχει ουσιαστικά καμία διαφορά μεταξύ των δύο.

**148. Εξηγήστε τις διαφορές μεταξύ γραμμικής (linear) και μη γραμμικής (non linear) επεξεργασίας video.**

Παλαιότερα υπήρχε διάκριση σε γραμμική (linear) και μη γραμμική (non-linear) επεξεργασία βίντεο. Η πρώτη μορφή χρησιμοποιούνταν στο μη αναλογικό βίντεο, όπου η επεξεργασία γινόταν με χρήση μηχανημάτων, που επεξεργάζονταν όλη τη μαγνητική ταινία, που περιείχε το βίντεο, από την αρχή έως το τέλος του. Στη μη γραμμική επεξεργασία το βίντεο εισάγεται απευθείας στον υπολογιστή, όπου και γίνεται επεξεργασία μεμονωμένων πλαισίων ή τμημάτων, εφαρμόζονται εφέ και επεξεργάζεται ο ήχος. Μπορεί να γίνει μετατροπή από μια μορφή σε μια άλλη και να χρησιμοποιηθεί διαφορετικός codec σε σχέση με τους αρχικούς. Επιπρόσθετα υποστηρίζεται η προσθήκη τίτλων και μενού και η εξαγωγή του βίντεο σε συγκεκριμένες μορφές ταινίας, π.χ. για DVD.

**149. Πότε χρησιμοποιούνται οι λειτουργίες προεπισκόπηση (preview video) και σύλληψη (capture video), κατά την επεξεργασία ψηφιακού video;**

Με τον όρο «σύλληψη» (capture) αναφερόμαστε στον τρόπο με τον οποίο καταφέρνουμε να εγγράψουμε το ψηφιακό βίντεο σε κάποιο μέσο. Ένας τρόπος είναι η μετατροπή του αναλογικού βίντεο σε ψηφιακό, σύμφωνα με τους τρόπους που μελετήσαμε στο τέλος της προηγούμενης ενότητας. Ο άλλος τρόπος είναι με χρήση προφανώς κάποιας κάμερας, που μπορεί να συλλάβει και να εγγράψει σε κάποιο μέσο το ψηφιακό βίντεο.

Η προεπισκόπηση χρησιμοποιείται για μια προτεινόμενη επιλογή χαρακτηριστικών πλάνων και καρέ, η οποία επιτρέπει την εύκολη και ταχεία πλοήγηση σε μεγάλες βάσεις δεδομένων οπτικοακουστικού υλικού και αποτελεί ισχυρό εργαλείο εξαγωγής περίληψης. Για παράδειγμα, έχει παρατηρηθεί ότι μία ακολουθία βίντεο διάρκειας 30 λεπτών αποτελείται συνήθως από 200 πλάνα. Επομένως, επιλέγοντας κατά μέσο όρο πέντε χαρακτηριστικά καρέ από κάθε πλάνο, απαιτούνται μόνο 1,000 από τα 45,000 συνολικά καρέ για την αναπαράσταση του οπτικού περιεχομένου.

**150. Τι είναι τα "Transition Effect" σε ένα πρόγραμμα επεξεργασίας video; Δώστε ένα παράδειγμα.**

Εφέ μετάβασης (transition effects) ονομάζονται οι τρόποι αλλαγής μεταξύ των πλάνων μιας ταινίας. Από παλιά οι μοντέρ χρησιμοποιούσαν διάφορες μεθόδους για να αλλάξουν πλάνο, όπως το απλό κόψιμο (cut), το σβήσιμο ενός πλάνου μέσα στο επόμενο, η εναλλαγή με μαύρο φόντο, κ.λ.π. Τα σύγχρονα προγράμματα είναι εξοπλισμένα με πολλές θεαματικές μεταβάσεις, οι οποίες μπορούν να χρησιμοποιηθούν και να εντυπωσιάσουν.

Οι μεταβάσεις πρέπει να ακολουθούν το πνεύμα της ταινίας και να μην μπερδεύουν το θεατή ή να του αποσπούν την προσοχή από την αφήγηση. Για παράδειγμα, μετά την έξοδο από μία αίθουσα κινηματογράφου αυτό που μένει στο θεατή είναι η πλοκή και σπανιότατα το εφέ κάποιας μετάβασης πλάνων. Η αλόγιστη χρήση εφέ μετάβασης έχει σαν αποτέλεσμα μια κουραστική και βαρετή για το θεατή ταινία και στη χειρότερη περίπτωση μπορεί να τον μπερδέψει. Απαιτείται αίσθηση μέτρου και λογικής. Στο παρελθόν έχει αποδειχτεί πολλές φορές ότι ένας κακός μοντέρ μπορεί να καταστρέψει έναν καλό σκηνοθέτη, ενώ ένας καλός συνάδελφός του μπορεί να σώσει μια αδιάφορη ταινία.

**151. Εξηγήστε τη λειτουργία των εργαλείων trimming (ψαλίδισμα) και cropping (ξάκρισμα), κατά την επεξεργασία μοντάζ ψηφιακού video.**

Κατά την επεξεργασία μοντάζ ψηφιακού video συχνά χρειάζεται να ψαλιδιστούν (trimming) οι ουρές των πλάνων στην αρχή και το τέλος ενός clip, γιατί συνήθως περιέχουν περιττό υλικό. Άλλες φορές ενδιαφέρει μόνο ένα συγκεκριμένο μέρος του κλιπ και όχι το σύνολό του, οπότε χρειάζεται να κοπεί το clip στα σημεία που ενδιαφέρουν και πεταχτεί το υπόλοιπο από τη γραμμή Timeline. Το ξάκρισμα (cropping) είναι η αφαίρεση ανεπιθύμητων περιοχών από ένα clip. Εκτελείται προκειμένου να αφαιρεθεί ένα ανεπιθύμητο θέμα ή άσχετες λεπτομέρειες από το clip, να αλλάξει ο λόγος διαστάσεων ή να βελτιωθεί η συνολική σύνθεση.

**152. Τι είναι ο υπέρτιτλος (Superimposed clip) στη διαδικασία μοντάζ ψηφιακού video;**

Η υπέρθεση (superimposing) περιγράφει τη διαδικασία της επικάλυψης και του συνδυασμού πολλαπλών εικόνων. Τα video clips είναι εντελώς αδιαφανή από προεπιλογή, αλλά η επικάλυψή τους απαιτεί διαφάνεια. Όταν κάνουμε διαφανή τα clips σε layers που βρίσκονται πιο πάνω στην ιεραρχία του timeline, αποκαλύπτονται clips από layers πιο χαμηλά στην ιεραρχία. Ο υπέρτιτλος είναι ένα clip το οποίο -συνήθως- περιέχει διαφάνεια και τοποθετείται σε υψηλότερο layer από κάποιο άλλο το οποίο παίζει το ρόλο του φόντου.

**153. Να περιγράψετε τα κύρια χαρακτηριστικά ενός ασυμπίεστου ψηφιακού αρχείου video.**

Τα κύρια χαρακτηριστικά ενός ασυμπίεστου ψηφιακού αρχείου video είναι τα εξής:

Ανάλυση: η οριζόντια και κάθετη διάσταση της εικόνας (pixels)

Αριθμός πλαισίων ανά δευτερόλεπτο ή αλλιώς συχνότητα πλαισίων: ο αριθμός των διαφορετικών καρέ (εικόνων-frames) που προβάλλονται ανά δευτερόλεπτο(fps)

Χρονική διάρκεια: ο χρόνος που διαρκεί η προβολή του βίντεο (δευτερόλεπτα).

Λόγος πλευρών: αναφέρεται στον λόγο της οριζόντιας διάστασης προς την κάθετη διάστασης.

Ρυθμός μεταφοράς bit: τα δεδομένα που περιέχονται σε ένα δευτερόλεπτο ενός αρχείου ή μιας ροής βίντεο (bit ανά δευτερόλεπτο).

**154. Τι είναι κωδικοποιητής ψηφιακών αρχείων video πραγματικού χρόνου (real time video encoders);**

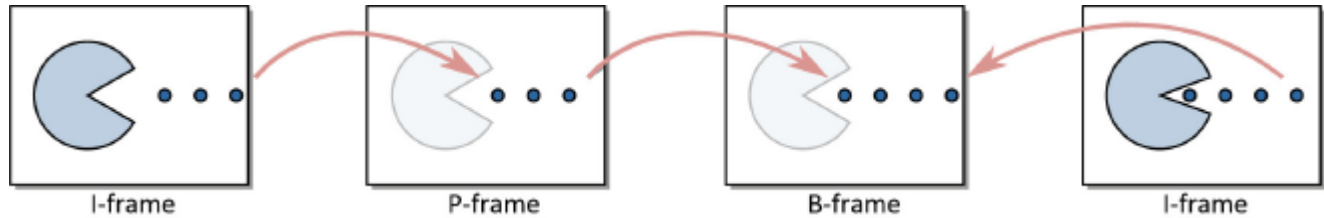
Ένας κωδικοποιητής ψηφιακών αρχείων video πραγματικού χρόνου επιτρέπει μετατροπή ή την επεξεργασία video από μια μορφή σχηματισμού ή δομής δεδομένων σε άλλη δομή μορφής ή δεδομένων όπου η διαδικασία λαμβάνει χώρα αμέσως ή μέσα σε πολύ μικρό χρονικό διάστημα μετά τη διάθεση των video για μετατροπή.

**155. Περιγράψτε την τεχνική συμπίεσης ψηφιακού video MPEG.**

Η συμπίεση της εικόνας βασίζεται κυρίως στη χρονική συμπίεση, δηλαδή τις ομοιότητες των καρέ του βίντεο. Τα περισσότερα από τα 30 καρέ, που μπορεί να περιέχει το βίντεο ανά δευτερόλεπτο, είναι όμοια μεταξύ τους. Στη χρονική συμπίεση των δεδομένων, η τεχνική MPEG αρχικά ταξινομεί τα

πλαίσια σε τρεις κατηγορίες: τα ενδοκωδικοποιημένα καρέ (Iframes), τα προβλεπόμενα καρέ (P-frames) και τα αμφίδρομα καρέ (B-frames).

Ένα I-frame (intra-coded) είναι στην ουσία μια πλήρης εικόνα, ενώ παίζει τον ρόλο του σημαντικού καρέ στη δημιουργία της ακολουθίας των εικόνων. Οι άλλοι τύποι καρέ βασίζονται στα προηγούμενα και στα επόμενα καρέ και έτσι διατηρούν μόνο τμήμα των πληροφοριών τους, επιτυγχάνοντας με αυτό τον τρόπο τη συμπίεση του βίντεο.



Ένα P-frame (predicted) αποθηκεύει πληροφορίες μόνο για τις αλλαγές στην εικόνα σε σχέση με το προηγούμενο καρέ. Στην εικόνα για παράδειγμα, βλέπουμε ότι το δεύτερο καρέ, δηλαδή το P-frame, μπορεί να βασιστεί στο προηγούμενο καρέ. Έτσι, αποθηκεύονται τελικά μόνο οι πληροφορίες για τα διαφορετικά pixels. Ένα B-frame (Bi-predictive) μπορεί να δημιουργηθεί από το προηγούμενο και το επόμενο καρέ.

**156. Περιγράψτε τα τυπικά βήματα της διαδικασίας Μοντάζ για την παραγωγή ενός ψηφιακού αρχείου video με τη χρήση ενός εργαλείου επεξεργασίας video (πχ. Premiere).**

1. Απόκτηση video
2. Αναγνώριση και εισαγωγή δεδομένων (capture)
3. Οργάνωση video clips
4. Κατασκευή και επεξεργασία timeline
5. Effects – Transitions
6. Τίτλοι – Γραφικά
7. Επεξεργασία ήχου
8. Εξαγωγή επεξεργασμένου video clip σε κατάλληλη μορφή

**157. Υπολογίστε το μέγεθος σε Kbytes από τον αποθηκευτικό χώρο ενός σκληρού δίσκου που δεσμεύεται κατά τη σύλληψη ασυμπίεστου ψηφιακού video χωρίς ήχο από βιντεοσυσκευή με σύνθετο βιντεοσήμα PAL, διάρκειας 2 λεπτών, με διαστάσεις 320 x 240 εικονοστοιχείων και βάθος χρώματος 16bit.**

$$320 \times 240 = 76800$$

$$76800 \times 25 = 1920000$$

$$1920000 \times 2 = 3840000 \text{ bytes}$$

$$3840000 \times 120 = 460800000 \text{ bytes} = 450000 \text{ Kb}$$

### 158. Αναφέρατε τους τύπους των symbols που υπάρχουν στο Flash και τα χαρακτηριστικά του καθενός από αυτούς.

Υπάρχουν τρία είδη συμβόλων: Graphic, Button και Movie clip.

#### Graphic

Είναι το πιο βασικό είδος συμβόλου. Τα σύμβολα γραφικών είναι τα λιγότερο ευέλικτα σύμβολα, επειδή δεν υποστηρίζουν ActionScript και δεν μπορείτε να εφαρμόσετε φίλτρα ή λειτουργίες ανάμειξης. Ωστόσο, είναι χρήσιμα όταν θέλετε να συγχρονιστεί μια κινούμενη εικόνα μέσα σε ένα γραφικό σύμβολο στην κύρια Timeline.

#### Button

Τα κουμπιά χρησιμοποιούνται σε μια εφαρμογή για να προσθέσουν διαδραστικότητα (interactivity) με το χρήστη. Ένα κουμπί μπορεί να αποτελείται από ένα μέχρι τέσσερα καρέ (key-frames) τα οποία ονομάζονται Up, Over, Down και Hit. Τα τρία πρώτα καρέ αντιστοιχούν στις τρεις διαφορετικές καταστάσεις (states) που μπορεί να έχει το κουμπί. Οι καταστάσεις αυτές εξαρτώνται από τη θέση και την κατάσταση του δείκτη (pointer) Το τέταρτο καρέ ορίζει την ενεργή περιοχή του κουμπιού και ονομάζεται Hit state.

Οι τρεις καταστάσεις (states) ενός κουμπιού είναι:

Up state (Όταν ο δείκτης βρίσκεται εκτός της ενεργής περιοχής του κουμπιού)

Over state (Όταν ο δείκτης βρίσκεται επάνω στην ενεργή περιοχή του κουμπιού)

Down state (Όταν ο δείκτης είναι πατημένος επάνω στην ενεργή περιοχή του κουμπιού)

#### Movie clip

Το movie clip είναι ένα μικρό movie το οποίο μπορεί να ενσωματωθεί σε ένα άλλο πιο μεγάλο movie. Κάθε movie clip έχει το δικό του timeline και το δικό του FrameRate και μπορεί να επεξεργαστεί όπως ένα κανονικό movie. Για να κρατήσει το δικό του FrameRate ένα movie clip πρέπει να φορτωθεί με την εντολή loadMovie. Σε διαφορετική περίπτωση το movie clip θα τρέξει με το FrameRate του movie στο οποίο έχει φορτωθεί ή έχει φιλοξενηθεί. Όταν δημιουργήσετε ένα movie clip μπορείτε να το αποθηκεύσετε στη βιβλιοθήκη και με μεταφορά και απόθεση (drag and drop) να παίρνετε όσα αντίγραφα θέλετε. Σε κάθε αντίγραφο (instance) μπορείτε να δώσετε ένα ξεχωριστό όνομα, έτσι ώστε να μπορείτε να διακρίνετε το ένα από το άλλο όταν χρειαστεί να αναφερθείτε σε ένα από αυτά μέσα από κάποια εντολή της ActionScript. Κάθε movie clip μπορεί να θεωρηθεί σαν αντικείμενο και σαν αντικείμενο έχει χαρακτηριστικά (properties) και λειτουργίες (methods). Τα χαρακτηριστικά αυτά και τις λειτουργίες μπορείτε να τις διαχειριστείτε προγραμματιστικά μέσα από την ActionScript για τη δημιουργία πιο σύνθετων εργασιών.

**159. Αναφέρατε τα πεδία του classic text που μπορούν να δημιουργηθούν από την εργαλειοθήκη του Flash και σε ποιες περιπτώσεις χρησιμοποιούνται.**

Τα πεδία του classic text είναι τα static, dynamic και input.

Static: Εμφανίζουν κείμενο που δεν αλλάζει δυναμικά.

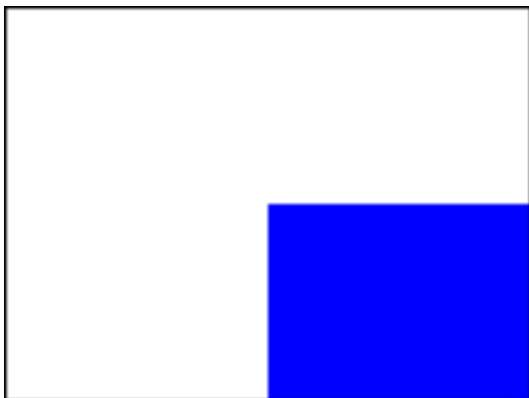
Dynamic: Το κείμενο που εμφανίζεται ενημερώνεται δυναμικά. Θα μπορούσε να εμφανίζει την τρέχουσα ημερομηνία ή το score σε κάποιο παιχνίδι.

Input: Επιτρέπει στους χρήστες να εισάγουν κείμενο. Χρήσιμο σε φόρμες ή έρευνες.

**160. Ποιες οι διαφορές μεταξύ keyframe, blanc keyframe και frame στο flash;**

Το frame είναι η μικρότερη μονάδα μεταβολής ενός αντικειμένου ή ενός συμβόλου στο timeline μιας ταινίας. Στις συμβατικές κινηματογραφικές ταινίες, το frame είναι μία από τις αναρίθμητες στατικές φωτογραφίες, οι οποίες, όταν προβάλλονται διαδοχικά και με τη σωστή ταχύτητα, δίνουν την αίσθηση της κίνησης. Τα keyframes είναι τα σημεία στα οποία μπορούμε να ορίσουμε εμείς οποιαδήποτε μεταβολή, όπως την αλλαγή του μεγέθους ή της πορείας ενός αντικειμένου και μετά να πούμε στο Flash να δημιουργήσει αυτόματα μόνο του τα ενδιάμεσα frames ανάμεσα στα keyframes. Η διαφορά μεταξύ της εισαγωγής ενός keyframe ή ενός blank keyframe είναι αν το Flash αντιγράφει ή όχι το περιεχόμενο του προηγούμενου keyframe στο νέο.

**161. Σε λευκό stage στο flash διαστάσεων 800\*600px υπάρχει ένα μπλε τετράγωνο διαστάσεων 400\*400px, το οποίο βρίσκεται στο σημείο 400,300 του stage. Περιγράψτε σχηματικά τι θα δει ο χρήστης όταν το movie γίνει publish.**



**162. Τι είναι τα code snippets;**

Τα code snippets είναι έτοιμα αποσπάσματα κώδικα που χρησιμοποιείται συχνά. Είναι ένας καλός τρόπος εκμάθησης κώδικα. Ο χρήστης μπορεί επίσης να δημιουργήσει το δικό του κομμάτι κώδικα και να το αποθηκεύσει σαν snippet με σκοπό την επαναχρησιμοποίησή του.



### 163. Πότε χρησιμοποιείται η εντολή trace bitmap;

Η εντολή trace bitmap χρησιμοποιείται όταν θέλουμε να μετατρέψουμε ένα Bitmap γραφικό σε διανυσματικό.

### 164. Ποια η διαφορά μεταξύ symbol και instance;

Το symbol είναι ένα γραφικό, ένα κουμπί ή ένα movie clip που δημιουργείται μία φορά στο Flash και μπορεί να επαναχρησιμοποιηθεί σε όλο το έγγραφό σας ή σε άλλα έγγραφα. Όλα τα σύμβολα αποθηκεύονται στην Βιβλιοθήκη. Το instance είναι ένα αντίγραφο ενός symbol που βρίσκεται στο Stage.

### 165. Πού βρίσκεται και πώς λειτουργεί το εργαλείο onion skin στο flash;

Με την λειτουργία onion skin, έχουμε την δυνατότητα να βλέπουμε ταυτόχρονα τα περιεχόμενα σε περισσότερα από ένα καρέ την φορά. Βρίσκεται στο κάτω μέρος του timeline.



Για να εμφανίσουμε πολλαπλά καρέ χρησιμοποιώντας την λειτουργία Onion Skin, στο διάγραμμα ροής χρόνου κάνουμε κλικ στο πλήκτρο Onion Skin. Θα εμφανισθεί ένα ζευγάρι από ενδείκτες δίπλα στην κεφαλή αναπαραγωγής στο διάγραμμα ροής χρόνου.

Το περιεχόμενο όλων των καρέ ανάμεσα στους δύο ενδείκτες εμφανίζεται τώρα στο σκηνικό και ορισμένα απ' αυτά τα καρέ είναι πριν από το τρέχον, ενώ μερικά είναι μετά. Σ' αυτή την κατάσταση είναι επεξεργάσιμα μόνο τα περιεχόμενα του τρέχοντος καρέ και το περιεχόμενο των μη-επεξεργάσιμων καρέ εμφανίζεται αμυδρό.

Μπορούμε να σύρουμε την κεφαλή αναπαραγωγής για να ξεφυλλίσουμε κι άλλα καρέ. Πρέπει να έχουμε υπόψη μας ότι δεν μπορούμε να βλέπουμε το περιεχόμενο των κλειδωμένων ή αόρατων επιπέδων όταν χρησιμοποιούμε την λειτουργία Onion Skin.

Για να εμφανίσουμε τα καρέ σαν περιγράμματα με την λειτουργία Onion Skin, στο διάγραμμα ροής χρόνου κάνουμε κλικ στο πλήκτρο Onion Skin Outlines. Όλο το περιεχόμενο των καρέ εκτός από το τρέχον εμφανίζεται σαν περίγραμμα. Μπορούμε να αναθέσουμε διαφορετικό χρώμα περιγράμματος σε κάθε επίπεδο για να μπορούμε να προσδιορίσουμε σε ποιο επίπεδο θέλουμε να κάνουμε αλλαγές.

Κανονικά, όταν χρησιμοποιούμε την λειτουργία Onion Skin μπορούμε να τροποποιήσουμε μόνο το περιεχόμενο του τρέχοντος καρέ. Για να κάνουμε επεξεργασία πολλαπλά καρέ, στο διάγραμμα ροής χρόνου κάνουμε κλικ στο πλήκτρο Edit Multiple Frames, οπότε γίνεται επεξεργάσιμο το περιεχόμενο όλων των καρέ που βρίσκονται ανάμεσα στους δύο ενδείκτες της λειτουργίας Onion Skin. Πρέπει να έχουμε υπόψη μας ότι μπορούμε να τροποποιούμε μόνο τα καρέ-κλειδιά σ' αυτή την κατάσταση.



Χρησιμοποιούμε του ενδείκτες της λειτουργίας Onion Skin για να καθορίζουμε την περιοχή των καρτέ που μπορούμε να ξεφυλλίζουμε και μπορούμε να αλλάζουμε τις θέσεις τους σύμφωνα με τις ανάγκες μας. Μπορούμε να προσαρμόζουμε τις θέσεις των ενδεικτών χειρωνακτικά ή μέσω των μενού.

Για να μετακινήσουμε τους ενδείκτες χειρωνακτικά, κάνουμε κλικ στη λαβή ενός ενδείκτη και τον σύρουμε στη νέα επιθυμητή θέση.

Για να τροποποιήσουμε τους ενδείκτες με το μενού, κάνουμε κλικ στο πλήκτρο Modify Onion Markers στο διάγραμμα ροής χρόνου και θα εμφανισθεί ένα μενού συντόμευσης με τις εξής επιλογές :

- Always Show Markers, οι ενδείκτες της λειτουργίας Onion Skin θα είναι πάντα ορατοί, ανεξάρτητα από το αν είναι ενεργή η λειτουργία αυτή.
- Anchor Onion, σταθεροποιεί ή κλειδώνει τους ενδείκτες της λειτουργίας Onion Skin στην τρέχουσα θέση τους, πράγμα που σημαίνει ότι θα παραμείνουν στατικοί αντί να διατηρούν τη σχετική τους θέση ως προς την κεφαλή αναπαραγωγής.
- Onion 2, ορίζει τους ενδείκτες σε θέσεις δύο καρτέ πριν και δύο καρτέ μετά από το τρέχον καρτέ, δηλ. τη θέση της κεφαλής αναπαραγωγής.
- Onion 5, ορίζει τους ενδείκτες σε θέσεις πέντε καρτέ πριν και πέντε καρτέ μετά από το τρέχον καρτέ.
- Onion All, ξεφυλλίζει όλα τα καρτέ της τρέχουσας σκηνής.

#### **166. Ποιος ο ρόλος του frame HIT στο timeline ενός button symbol; Τι πρόβλημα δημιουργείται με κουμπί που περιέχει μόνο κείμενο και πώς το αντιμετωπίζετε;**

Το frame HIT στο timeline ενός button symbol ελέγχει την περιοχή στην οποία μπορούμε να κάνουμε κλικ σε ένα κουμπί. Το frame δεν είναι ορατό, οπότε δεν έχει καμία επίδραση στην πραγματική εμφάνιση ενός κουμπιού. Όταν ένα κουμπί που περιέχει μόνο κείμενο υπάρχει κίνδυνος να μην μπορεί ο χρήστης να αλληλεπιδράσει μαζί του επειδή είναι δύσκολο να πετύχει τα pixels του κειμένου με το ποντίκι. Για να λύσουμε αυτό το πρόβλημα δημιουργούμε ένα ορθογώνιο στο frame HIT με μέγεθος όσο το κείμενο.

#### **167. Ποια είδη tweening υπάρχουν στο Flash και ποια είναι τα χαρακτηριστικά τους;**

Το tween ορίζεται ως η παραγωγή ή η δημιουργία ενδιάμεσων αντικειμένων ανάμεσα σε δύο διαφορετικές μορφές ενός αντικειμένου: Την αρχική και την τελική. Το Flash μπορεί να κάνει tween στα σχήματα (shapes) και στα σύμβολα (symbols). Στην περίπτωση που η κίνηση γίνεται ανάμεσα σε σχήματα το tween λέγεται shape tween και στην περίπτωση που η κίνηση γίνεται ανάμεσα σε σύμβολα λέγεται motion tween. Άρα υπάρχουν 2 είδη για tween animation. Shape tween και Motion tween. Γενικά, για να δημιουργήσουμε ένα tween, απαιτούνται δύο keyframes. Στο πρώτο keyframe αντιστοιχεί το αντικείμενο με την αρχική του μορφή (θέση, διαστάσεις, χρώμα, διαφάνεια κλπ) και στο δεύτερο keyframe αντιστοιχεί το ίδιο αντικείμενο με την τελική του μορφή δηλαδή με διαφορετικά χαρακτηριστικά. Η ιδέα του tween είναι ότι δημιουργούμε τις δύο ακραίες καταστάσεις ή

μορφές (αρχική και τελική) και το Flash αναλαμβάνει να δημιουργήσει όλες τις ενδιάμεσες μορφές έτσι ώστε να υπάρχει μια ομαλή μετάβαση από την αρχική κατάσταση στην τελική.

### Shape tween

Το shape tween εφαρμόζεται ανάμεσα σε σχήματα (shapes) και στη διάρκεια της κίνησης τα χαρακτηριστικά του σχήματος αλλάζουν σταδιακά.

Στη διάρκεια ενός shape tween ένα ή περισσότερα από τα χαρακτηριστικά που μπορεί να αλλάξουν είναι:

- position (θέση)
- size (διάσταση)
- transformation (παραμόρφωση)
- color (χρώμα)
- Διαφάνεια (Alpha)
- shape (σχήμα)

### Motion tween

Το motion tween εφαρμόζεται σε σύμβολα (graphics, buttons, movie clips) και όχι σε shapes. Στη διάρκεια ενός motion tween τα χαρακτηριστικά που μπορεί να αλλάξουν είναι:

- position (θέση)
- size (διάσταση)
- transformation (παραμόρφωση)
- color (χρώμα)
- rotation (περιστροφή)
- transparency (διαφάνεια)

Σε ένα motion tween μπορούμε να έχουμε ένα αντικείμενο το οποίο περιστρέφεται. Αυτό το χαρακτηριστικό δεν υπάρχει στο shape tween και επομένως αν θέλουμε ένα αντικείμενο να περιστρέφεται θα πρέπει πρώτα να το μετατρέψουμε σε σύμβολο και συνήθως σε σύμβολο τύπου graphic.

## 168. Πώς δουλεύει το εργαλείο Bone Tool;

Το Bone Tool χρησιμοποιείται στην δημιουργία προχωρημένων animation όπου θέλουμε τα αντικείμενα μας να έχουν την δική τους κίνηση και αλλαγή στην κατάστασή τους. Θα εξηγήσουμε πως δουλεύει με ένα παράδειγμα:

Δημιουργούμε ένα νέο έγγραφο Flash και επιλέξτε το ActionScript 3.0. Σχεδιάζουμε μια έλλειψη χρησιμοποιώντας το εργαλείο έλλειψης. Μετατρέπουμε το σχήμα έλλειψης σε σύμβολο κάνοντας δεξί κλικ στο σχήμα. Επιλέγουμε "convert to symbol" και επιλέγουμε "graphic" στο πλαίσιο Type. Διπλασιάζουμε το σύμβολο κρατώντας πατημένο το πλήκτρο Alt (Windows) ή Option (Mac OS) και σέρνοντας το σύμβολο σε μια νέα θέση. Το Flash θα αντιγράψει κάθε φορά που κάνουμε το σχήμα click and drag. Σχεδιάζουμε κάτι σαν την παρακάτω εικόνα.



Στη συνέχεια, επιλέγουμε το bone tool. Επιλέγουμε την πρώτη έλλειψη χρησιμοποιώντας το εργαλείο οστών (M), και την συνδέουμε το με τη δεύτερη έλλειψη. Κάνουμε το ίδιο για τα υπόλοιπα σύμβολα έλλειψης. Το σύνολο των συνδεδεμένων αντικειμένων ονομάζεται κινηματική αλυσίδα (Armature). Έπειτα επιλέγουμε το εργαλείο επιλογής (V) και σέρνουμε το τελευταίο οστό. Μπορούμε να χειριστούμε ολόκληρη την κινηματική αλυσίδα σε πραγματικό χρόνο σέρνοντας το τελευταίο οστό. Μπορούμε να δημιουργούμε ένα νέο keyframe που αντιγράφει τις ιδιότητες του προηγούμενου animation πατώντας το πλήκτρο F6 στο πληκτρολόγιό μας.

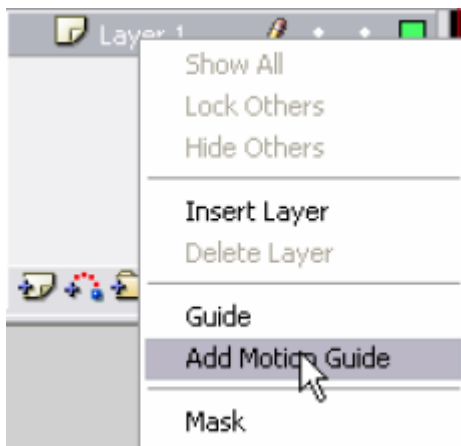
**169. Γράψτε τον κώδικα σε Actionscript 3 για ένα movie clip, το οποίο θα μετακινείται 5px στο stage ανάλογα με την πίεση των βελών του πληκτρολογίου (αριστερά-δεξιά-πάνω-κάτω).**

```
stage.addEventListener(KeyboardEvent.KEY_DOWN, myKeyDown);  
function myKeyDown(e:KeyboardEvent):void {  
    switch (e.keyCode) {  
        case Keyboard.LEFT :  
            my_mc.x-=5;  
            break;  
        case Keyboard.RIGHT :  
            my_mc.x+=5;  
            break;  
        case Keyboard.UP :  
            my_mc.y-=5;  
            break;  
        case Keyboard.DOWN :  
            my_mc.y+=5;  
            break;  
    }  
}
```

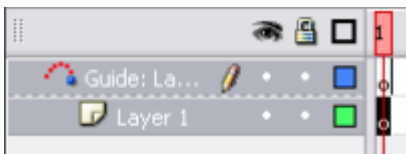
**170. Με ποιο τρόπο κάνουμε path animation ενός tween στο Flash;**

Η τεχνική κατά την οποία ένα αντικείμενο ακολουθεί σε ένα animation μια συγκεκριμένη διαδρομή ονομάζεται Path animation. Θα δούμε μέσω μιας δραστηριότητας με ποιο τρόπο την υλοποιούμε. Συγκεκριμένα θα φτιάξουμε ένα μικρό κύκλο που θα κινείται πάνω σε μια συγκεκριμένη διαδρομή που θα του υποδείξουμε εμείς.

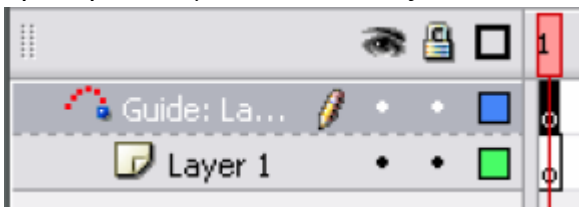
1. Αρχικά επιλέγουμε το layer 1 κάνοντας αριστερό κλικ με το ποντίκι πάνω σε αυτό. Στην συνέχεια κάνουμε δεξί κλικ και επιλέγουμε Add Motion Guide.



2. Μόλις το επιλέξουμε τότε θα παρατηρήσουμε πως αυτόματα το flash μετονομάζει το layer και από layer 1 το κάνει Guide: Layer 1, ενώ ταυτόχρονα δημιουργεί ένα νέο layer με όνομα layer 1.



3. Στην αρχή είναι και τα δύο layer επιλεγμένα. Εμείς επιλέγουμε μόνο το Guide Layer κάνοντας αριστερό κλικ με το ποντίκι σας πάνω σε αυτό.



4. Τώρα επιλέγουμε το pencil tool από την εργαλειοθήκη του flash και σχεδιάζουμε ένα τυχαίο μονοπάτι το οποίο θα μας χρησιμεύσει ως οδηγός για το πώς θα κινηθεί στην συνέχεια ο κύκλος πάνω σε αυτό. Φτιάχνουμε κάτι παρόμοιο με αυτό που φαίνεται παρακάτω:

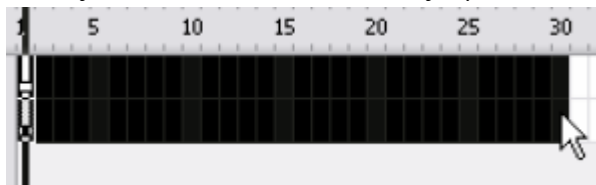


5. Είμαστε έτοιμοι να ζωγραφίσουμε ένα μικρό κύκλο, ο οποίος θα κινείται στο παραπάνω μονοπάτι που σχεδιάσαμε. Αρχικά επιλέγουμε το frame 1 του layer 1 κάνοντας αριστερό κλικ με το ποντίκι. Αυτό είναι σημαντικό μιας και σε αυτό το layer θα βάλουμε το κύκλο μας. Στην συνέχεια, κατά τα γνωστά, επιλέγουμε το εργαλείο οβάλ (oval tool) και σχεδιάζουμε έναν μικρό κύκλο πάνω στην σκηνή, όπως φαίνεται παρακάτω:

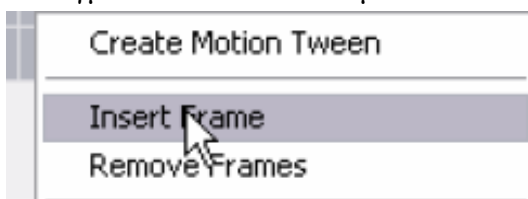


6. Σε αυτό το σημείο θα δημιουργήσουμε ένα motion tween όπως ακριβώς κάναμε παραπάνω. Συνεπώς κάνουμε διπλό κλικ πάνω στο κύκλο και εν συνεχεία κάνοντας δεξί κλικ επιλέγουμε Convert To Symbol. Από το παράθυρο που θα εμφανιστεί καθορίστε το Type ως graphic, δίνουμε ένα χαρακτηριστικό όνομα όπως π.χ my\_circle στο πεδίο name και πατήστε OK.

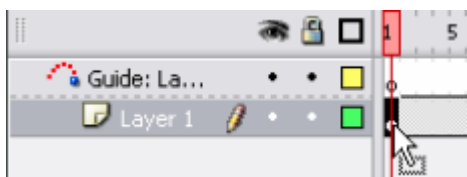
7. Τώρα πάμε να προσθέσουμε frames μέχρι και το frame 30 και στα δύο layer που έχουμε στην timeline. Για να γίνει αυτό κάνουμε αριστερό κλικ πάνω στο frame 1 του guide layer και σύρετε προς τα δεξιά και κάτω ωσότου επιλέξουμε τα πάνω 30 και τα κάτω 30 frames των δύο layer.



8. Αφήνουμε ελεύθερο το δείκτη του ποντικιού και στην συνέχεια κάνουμε δεξί κλικ σε ένα από τα επιλεγμένα frames. Από το μενού που θα εμφανιστεί επιλέγουμε Insert Frame.



9. Μόλις εισάγουμε τα frames στα δύο layers επιλέγουμε το frame 1 του layer 1 κάνοντας αριστερό κλικ πάνω σε αυτό.

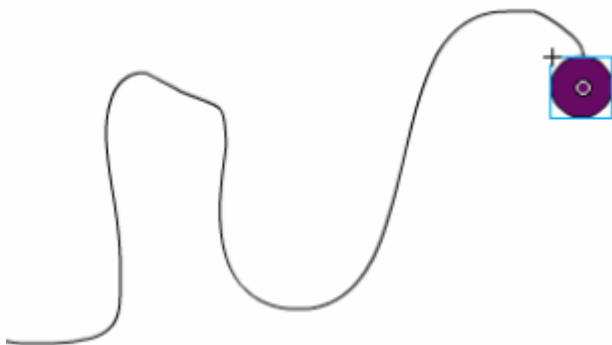


10. Σε αυτό το σημείο επιλέγουμε το κύκλο χρησιμοποιώντας το εργαλείο επιλογής (selection tool) κάνουμε ένα απλό κλικ πάνω σε αυτό και τοποθετούμε το πάνω στην άκρη της αριστερής άκρης του μονοπατιού σας όπως φαίνεται στο παρακάτω σχήμα. Παρατηρούμε επίσης πως όταν το κέντρο του κύκλου πλησιάσει στο αριστερό άκρο του μονοπατιού αυτόματα το ίδιο το flash το έλκει προς το μέρος του.

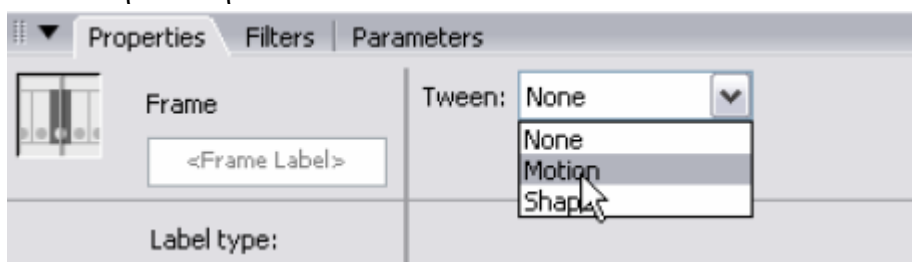


11. Στην συνέχεια επιλέγουμε το frame 30 του layer 1 και εισάγουμε σε αυτό ένα keyframe. Για να το κάνουμε αφού επιλέξουμε το frame 30 κάνουμε δεξί κλικ πάνω σε αυτό και επιλέγουμε Insert keyframe.

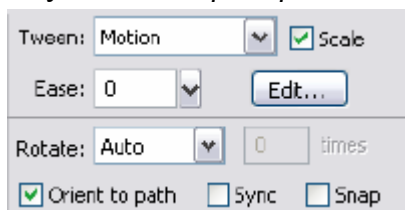
12. Τώρα επιλέγουμε και πάλι το κύκλο που βρίσκετε στην αριστερή άκρη του μονοπατιού και σύρετε τον τοποθετώντας τον τελικά στην δεξιά άκρη του μονοπατιού, όπως φαίνεται παρακάτω:



13. Τώρα θα φτιάξουμε ένα Motion Tween πάνω στο κύκλο υποδεικνύοντας του να κινηθεί δια μέσου του μονοπατιού. Για να το πετύχουμε επιλέγουμε ένα τυχαίο frame από το 2 έως και το 29 του layer 1. Προσοχή σε αυτό το layer έχουμε τον κύκλο εκεί θα φτιάξουμε το animation. Από το παράθυρο ιδιοτήτων (Properties Windows) που εμφανίζεται στο κάτω μέρος της σκηνής επιλέγουμε στο πεδίο Tween την ιδιότητα Motion.



14. Μόλις το επιλέξουμε αυτό, τότε στο ίδιο παράθυρο εμφανίζονται και κάποιες πρόσθετες επιλογές. Αυτή που χρειαζόμαστε είναι η επιλογή «orient to path». Μέσω της οποίας θα μπορέσουμε να συνδέσουμε την κίνηση του κύκλου με το μονοπάτι. επιλέγουμε το κουτί κάνοντας κλικ με το ποντίκι σας στο κουτί αριστερά του πεδίου «orient to path».



15. Τέλος επιλέγουμε Control | test Movie από το κύριο μενού του Flash για να δούμε το κύκλο να κινείται επαναληπτικά πάνω ακριβώς στην διαδρομή που ορίσαμε.

**171. Δημιουργήστε σε AS3 την function DistanceTwoPoints που θα υπολογίζει την απόσταση μεταξύ δύο σημείων πάνω στο stage.**

```
function DistanceTwoPoints(x1:Number, x2:Number, y1:Number, y2:Number): Number {
    var dx:Number = x1-x2;
    var dy:Number = y1-y2;
    return Math.sqrt(dx * dx + dy * dy);
}
```

**172. Περιγράψτε πώς επιδρά μια μάσκα σε ένα layer που «μασκάρεται» στο Flash. Πώς θα δημιουργούσατε έναν τίτλο που να δείχνει ότι κυματίζει, όπως το νερό της πισίνας;**

Η μάσκα χρησιμοποιείται για να καλύπτει ένα αντικείμενο και να επιτρέπει να εμφανίζεται μόνο το τμήμα του αντικειμένου που καλύπτει η μάσκα. Οι μάσκες βρίσκονται σε ειδικά επίπεδα τα οποία λέγονται mask layers. Το επίπεδο ή τα επίπεδα που βρίσκονται κάτω από το mask layer λέγονται masked layers. Μια μάσκα δεν είναι τίποτα άλλο παρά ένα σχήμα (shape) ή ένας συνδυασμός σχημάτων ή ένα σύμβολο (graphic, movie clip ή button). Ο χώρος που καλύπτουν αυτά τα αντικείμενα είναι ο διάφανος χώρος (τρύπες) μέσα από τον οποίο φαίνονται τα περιεχόμενα των masked layers. Δεν μπορούμε να έχουμε μάσκα μέσα σε ένα κουμπί. Μπορούμε όμως να έχουμε μάσκα σε ένα movie clip. Σε μια μάσκα μπορούμε να έχουμε ένα tween animation ή ένα shape animation για πιο εντυπωσιακά εφέ. Έτσι για να δημιουργήσουμε έναν τίτλο που να δείχνει ότι κυματίζει, όπως το νερό της πισίνας θα τοποθετήσουμε στο masked layer ένα animation με νερό και στο mask layer τον τίτλο που θέλουμε.

**173. Περιγράψτε τα αντικείμενα που πρέπει να δημιουργηθούν και τον τρόπο που πρέπει να τοποθετηθούν μέσα στην εφαρμογή, έτσι ώστε να έχουμε έναν κόκκινο κύκλο να διασχίζει οριζόντια το stage, φέρνοντας μαζί του ένα άλλο μικρότερο μπλε κύκλο σαν δορυφόρο του (δηλ. να διαγράφει κυκλική τροχιά γύρω από αυτόν) (στο Flash).**

Δημιουργούμε 2 κυκλικά σχήματα στο πρώτο frame του timeline. Ένα με χρώμα κόκκινο και ένα μπλέ με διάμετρο μικρότερη του κόκκινου. Μετατρέπουμε το κόκκινο σχήμα σε MovieClip και του δίνουμε όνομα orbiter. Μετατρέπουμε το μπλέ σχήμα σε MovieClip και του δίνουμε όνομα orbit. Κάνουμε click στο κόκκινο MovieClip και στην καρτέλα properties δίνουμε instance name orbiterMC. Κάνουμε click στο μπλέ MovieClip και στην καρτέλα properties δίνουμε instance name orbitMC. Δημιουργούμε ένα νέο layer, το ονομάζουμε AS3 και στο πρώτο frame τοποθετούμε τον παρακάτω κώδικα ActionScript 3

```
var orbitAngle:Number = 0; // Αρχική γωνία
var orbitSpeed:Number = 10; //Αριθμός εικονοστοιχείων ανά τροχιά
var orbitradius:Number = 90; // Τρέχουσα απόσταση από την προέλευση
var orbiterSpeed:Number = 5; //πόσο γρήγορα κινείται το orbiterMC
stage.addEventListener(Event.ENTER_FRAME, loop);
function loop(e:Event):void
{
    var rad:Number = orbitAngle * (Math.PI / 180); // Μετατροπή μοίρες -> ακτίνια
    orbitMC.x = orbiterMC.x + orbitradius * Math.cos(rad);
```



```

orbitMC.y = orbiterMC.y + orbitradius * Math.sin(rad);
orbitAngle+=orbitSpeed;
orbiterMC.x += orbiterSpeed; //σε κάθε frame μετατοπίζουμε το movieclip mc_1 στον
οριζόντιο άξονα προσθέτοντας την τιμή της mc_1SpeedX
if(orbiterMC.x >= stage.stageWidth+orbiterMC.width/2){ //έλεγχος του αν το x position
στην δεξιά πλευρά του mc_1 είναι μικρότερο ή ίσο με την δεξιά πλευρά της οθόνης
    orbiterMC.x = -orbiterMC.width/2; //τοποθέτηση του mc_1 στην αριστερή
στην πλευρά της οθόνης
}
}

```

**174. Δώστε ένα παράδειγμα (στο Flash) που γίνεται collision detection μεταξύ του movieclip mc\_1 και του mc\_2. Το movieclip mc\_1 κινείται αρχικά με βήμα 5px στον οριζόντιο άξονα του stage(800\*600). Εάν εξέλθει του stage από μια πλευρά εισέρχεται από την άλλη, διατηρώντας την κατεύθυνσή του. Εάν συναντήσει το movieclip mc\_2 η κατεύθυνσή του αναστρέφεται.**

```

var mc_1SpeedX:int = 5; //πόσο γρήγορα κινείται το mc_1
stage.addEventListener(Event.ENTER_FRAME, loop);
function loop(e:Event):void
{
    mc_1.x += mc_1SpeedX; //σε κάθε frame μετατοπίζουμε το movieclip mc_1 στον οριζόντιο
    άξονα προσθέτοντας την τιμή της mc_1SpeedX
    if (mc_1.hitTestObject(mc_2)) // collision detection μεταξύ του movieclip mc_1 και
    του mc_2
    {
        mc_1SpeedX*= -1; // αναστρέφουμε την κατεύθυνση του mc_1
    }
    if(mc_1.x >= stage.stageWidth-mc_1.width/2){ //έλεγχος του αν το x position στην
    δεξιά πλευρά του mc_1 είναι μικρότερο ή ίσο με την δεξιά πλευρά της οθόνης
        mc_1.x = mc_1.width/2; //τοποθέτηση του mc_1 στην αριστερή στην πλευρά της οθόνης
    }
}

```

## 4. Πρακτικό Μέρος: Κατάλογος Στοχοθεσίας Πρακτικών Ικανοτήτων και Δεξιοτήτων (Στοχοθεσία Εξεταστέας Ύλης Πρακτικού Μέρους)

Για την πιστοποίηση της επαγγελματικής ικανότητας, κατά το Πρακτικό Μέρος, οι υποψήφιοι της ειδικότητας **ΤΕΧΝΙΚΟΣ ΕΦΑΡΜΟΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ (ΠΟΛΥΜΕΣΑ/WEBDESIGNER-DEVELOPER/VIDEOGAMES)** εξετάζονται σε γενικά θέματα επαγγελματικών γνώσεων και ικανοτήτων και επίσης σε ειδικές επαγγελματικές γνώσεις και ικανότητες, που περιλαμβάνονται αποκλειστικά στη στοχοθεσία του πρακτικού μέρους της ειδικότητας.

### **ΘΕΜΑΤΑ ΕΞΕΤΑΣΗΣ ΠΡΑΚΤΙΚΟΥ ΜΕΡΟΥΣ ενδεικτικά:**

Σχεδιασμός ολοκληρωμένου έργου με συνδυασμό τουλάχιστον τριών προγραμμάτων, βάσει δοθέντος ενδεικτικού προτύπου σε μορφή jpg ή pdf ή κατάλληλης περιγραφής των ζητούμενων. Το έργο δύναται να περιλαμβάνει σχεδιασμό γραφικών, επεξεργασία κειμένων, πινάκων, εικόνων κλπ. Τα απαιτούμενα κείμενα δίνονται σε μορφή doc-docx και οι εικόνες σε μορφή jpg. Οι εξεταζόμενοι καλούνται να δημιουργήσουν και να παραδώσουν προς αξιολόγηση με τη βοήθεια κατάλληλων λογισμικών το ζητούμενο έργο ολοκληρωμένο σε ψηφιακή μορφή.

Ενδεικτικά ζητούμενα έργα:

- Δημιουργία δυναμικού website (πχ e-shop).
- Ανάπτυξη εφαρμογών, με χρήση γλώσσας προγραμματισμού Java.
- Δημιουργία διαδραστικής εφαρμογής για κινητές υπολογιστικές μικροσυσκευές (tablet, smartphones κλπ.), συμβατής με κάθε λειτουργικό σύστημα (πχ χρήση γλώσσας SWIFT2) .
- Δημιουργία διαδικτυακής πλατφόρμας υποστήριξης και παρουσίασης για κινητές υπολογιστικές μικροσυσκευές και υπολογιστές γραφείου.
- Εφαρμογή για «Εξυπνο» Σπίτι (Απομακρυσμένη πρόσβαση σε φωτισμούς, θέρμανση, συναγερμό ασφαλείας κλπ) με τη χρήση σύγχρονων γλωσσών προγραμματισμού, που αφορούν στη χρήση «Εξυπνων Συσκευών».
- Σχεδιασμός τρισδιάστατου αντικειμένου με συγκεκριμένα δεδομένα, φωτισμό κλπ.
- Δημιουργία παιχνιδιού / animation.

Στα ανωτέρω προτεινόμενα Θέματα Εξέτασης Πρακτικού Μέρους, δύνανται να ζητούνται: επεξεργασία εικόνων, δημιουργία / ενσωμάτωση video ή /και animation, βάσεων δεδομένων κλπ και κάθε άλλο στοιχείο βάσει των μαθησιακών αποτελεσμάτων του οδηγού σπουδών.