

ΤΕΧΝΙΚΟΣ ΕΦΑΡΜΟΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ (ΠΟΛΥΜΕΣΑ / WEBDESIGNER - DEVELOPER / VIDEOGAMES)

ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΘΕΩΡΗΤΙΚΗΣ ΕΞΕΤΑΣΗΣ ΕΟΠΠΕΠ

ΑΠΟ ΤΟΝ ΓΙΩΡΓΟ ΚΑΜΠΕΡΟΓΙΑΝΝΗ

cambersa@gmail.com

ΟΜΑΔΑ Α. ΓΕΝΙΚΕΣ ΕΡΩΤΗΣΕΙΣ

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

Ένας ηλεκτρονικός υπολογιστής αποτελείται από τα εξής βασικά τμήματα:

Κεντρική Μονάδα Επεξεργασίας (CPU):

Η CPU είναι ο "εγκέφαλος" του υπολογιστή. Εκτελεί όλες τις εντολές και επεξεργάζεται τα δεδομένα.

Μνήμη:

Αποθηκεύει προσωρινά ή μόνιμα τα δεδομένα και τις εντολές που χρειάζεται η CPU για την επεξεργασία.

Υπάρχουν 2 κατηγορίες:

- Μνήμη RAM: Χρησιμοποιείται για προσωρινή αποθήκευση των δεδομένων κατά την εκτέλεση προγραμμάτων. Διαγράφεται όταν ο υπολογιστής απενεργοποιείται.
- Μνήμη ROM: Περιέχει δεδομένα και εντολές που δεν αλλάζουν, όπως το BIOS, και είναι απαραίτητη για την εκκίνηση του συστήματος.

Μονάδες Αποθήκευσης:

Αποθηκεύουν δεδομένα και αρχεία μακροπρόθεσμα.

Σ' αυτή την κατηγορία ανήκουν: Σκληροί δίσκοι, USB sticks, εξωτερικοί δίσκοι, CD-ROM κλπ.

Συσκευές Εισόδου:

Επιτρέπουν στον χρήστη να εισάγει δεδομένα και εντολές στον υπολογιστή.

Σ' αυτή την κατηγορία ανήκουν: Πληκτρολόγια, ποντίκια, σαρωτές, κάμερες κλπ

Συσκευές Εξόδου:

Επιτρέπουν στον υπολογιστή να εμφανίζει αποτελέσματα και πληροφορίες στον χρήστη.

Σ' αυτή την κατηγορία ανήκουν: Οθόνες, εκτυπωτές, ηχεία κλπ.

Μητρική Πλακέτα:

Η μητρική πλακέτα συνδέει όλα τα παραπάνω τμήματα και επιτρέπει την επικοινωνία μεταξύ τους. Περιέχει τις υποδοχές για τη CPU, τη μνήμη, τις συσκευές αποθήκευσης και άλλες μονάδες, καθώς και διασυνδέσεις για εξωτερικές συσκευές.

Τροφοδοτικό:

Παρέχει την απαραίτητη ηλεκτρική ενέργεια για τη λειτουργία όλων των εξαρτημάτων του υπολογιστή, μετατρέποντας το ρεύμα από το δίκτυο στη σωστή τάση και ένταση για κάθε τμήμα.

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

Εκτυπωτές Ψεκασμού (Inkjet):

Χρήση: Ιδανικοί για οικιακή χρήση και για γραφεία με μέτριες απαιτήσεις εκτύπωσης. Προσφέρουν καλή ποιότητα εκτύπωσης φωτογραφιών και γραφικών.

Χαρακτηριστικά: Ψεκάζουν μικροσκοπικές σταγόνες μελάνης στο χαρτί. Κατάλληλοι για έγχρωμες εκτυπώσεις υψηλής ανάλυσης αλλά σχετικά αργοί σε σύγκριση με άλλους τύπους.

Εκτυπωτές Laser:

Χρήση: Καλύτεροι για επιχειρήσεις ή γραφεία με ανάγκες μεγάλου όγκου εκτυπώσεων. Προσφέρουν γρήγορη και οικονομική εκτύπωση κειμένων και γραφικών.

Χαρακτηριστικά: Χρησιμοποιούν λέιζερ και τύμπανο toner για τη δημιουργία εικόνων. Είναι οικονομικοί για μεγάλους όγκους εκτυπώσεων και προσφέρουν υψηλή ταχύτητα.

Εκτυπωτές Θερμικής Μεταφοράς (Thermal Printers):

Χρήση: Χρησιμοποιούνται για την εκτύπωση αποδείξεων, ετικετών και barcode. Συχνά βρίσκονται σε POS συστήματα και ταμειακές μηχανές.

Χαρακτηριστικά: Επικαλυμμένο χαρτί που αντιδρά σε θερμότητα ή χρήση μελανιού με θερμική μεταφορά. Προσφέρουν αντοχή σε υγρασία και φθορά, αλλά είναι περιορισμένοι σε ποιότητα έγχρωμης εκτύπωσης.

Εκτυπωτές Ακίδων (Dot Matrix):

Χρήση: Κατάλληλοι για περιβάλλοντα με ειδικές ανάγκες, όπως η εκτύπωση πολλαπλών αντιγράφων ή τιμολογίων, λόγω της δυνατότητας διαδοχικών εκτυπώσεων μέσω καρμπόν.

Χαρακτηριστικά: Χρησιμοποιούν κεφαλές εκτύπωσης με βελόνες για να χτυπούν ένα μελανωμένο φύλλο. Είναι ανθεκτικοί, αλλά η ποιότητα και η ταχύτητα εκτύπωσης είναι χαμηλές σε σύγκριση με άλλους τύπους.

Πολυμηχανήματα:

Τα πολυμηχανήματα είναι εκτυπωτές laser ή ψεκασμού που μπορούν να εκτελέσουν κι άλλες εργασίες όπως σάρωση, φωτοτυπίες, φαξ κλπ. Συνδυάζουν πολλές δυνατότητες καταλαμβάνοντας μικρό χώρο, καθιστώντας τους κατάλληλους για μικρές και μεσαίες επιχειρήσεις.

Εκτυπωτές 3D:

Χρήση: Χρησιμοποιούνται σε πρωτότυπη κατασκευή, στην ιατρική, την αρχιτεκτονική και τη μηχανολογία για την εκτύπωση τρισδιάστατων αντικειμένων.

Χαρακτηριστικά: Χρησιμοποιούν υλικά όπως πλαστικό, ρητίνη ή μέταλλο και δημιουργούν αντικείμενα στρώση-στρώση. Οι εκτυπωτές αυτοί είναι αργοί αλλά προσφέρουν δυνατότητες για δημιουργία πρωτότυπων και προσαρμοσμένων προϊόντων.

Κριτήρια Αξιολόγησης Εκτυπωτών

Ποιότητα εκτύπωσης (dpi resolution), δυνατότητα έγχρωμης εκτύπωσης, ταχύτητα εκτύπωσης (ppm), κόστος ανά σελίδα, συνδεσιμότητα (USB, Ethernet, Wi-Fi), πρόσθετες λειτουργίες (σάρωση, φωτοτυπικό), μέγεθος, κόστος συντήρησης, αξιοπιστία.

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

Μαγνητικά Μέσα:

Αυτά τα μέσα χρησιμοποιούν μαγνητισμό για την αποθήκευση δεδομένων. Τα δεδομένα αποθηκεύονται σε μαγνητικούς δίσκους ή ταινίες που περιστρέφονται, ενώ οι κεφαλές διαβάζουν και γράφουν τα δεδομένα.

Οπτικά Μέσα:

Βασίζονται στη χρήση φωτεινής δέσμης λέιζερ για την ανάγνωση και εγγραφή δεδομένων. Τα δεδομένα είναι αποθηκευμένα σε μια ειδική επιφάνεια δίσκου, την οποία το λέιζερ διαβάζει, ή σε ορισμένες περιπτώσεις, γράφει.

Μνήμες Flash (Solid-State Storage):

Αυτή η κατηγορία περιλαμβάνει αποθηκευτικά μέσα που χρησιμοποιούν μνήμη flash για την αποθήκευση δεδομένων, χωρίς κινούμενα μέρη.

Οι σκληροί δίσκοι (HDD) ανήκουν στα μαγνητικά μέσα, οι δίσκοι SSD στις μνήμες Flash ενώ οι οπτικοί δίσκοι (CD, DVD) στα οπτικά μέσα.

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

Ο σαρωτές είναι συσκευές που φωτογραφίζουν ένα έγγραφο και το ψηφιοποιούν μεταφέροντας το σαν εικόνα στον υπολογιστή. Τα κριτήρια αξιολόγησης ενός σαρωτή είναι η ποιότητα σάρωσης, η ταχύτητα, τα υποστηριζόμενα μεγέθη (A4/A3) και η τιμή.

Τα προγράμματα OCR διαβάζουν εικόνες και εάν περιέχουν γράμματα προσπαθούν να τα μετατρέψουν σε χαρακτήρες κειμένου ώστε να μπορούμε να τα επεξεργαστούμε με κάποιον επεξεργαστή κειμένου όπως το Microsoft Word;

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

Η άδεια χρήσης λογισμικού καθορίζει τους όρους χρήσης του λογισμικού και τα νομικά δικαιώματα που απορρέουν μετά την απόκτηση του. Όταν αγοράζουμε ένα λογισμικό δεν αποκτούμε την πλήρη κυριότητα αλλά μας παραχωρείται το δικαίωμα να το χρησιμοποιήσουμε. Πως ακριβώς μπορούμε να το χρησιμοποιήσουμε αναφέρεται στους όρους χρήσης (πχ αν μπορούμε να το αντιγράψουμε σε άλλες συσκευές μας, να το δώσουμε σε άλλους χρήστες κλπ). Αν παραβούμε κάποιον από τους όρους χρήσης τότε καταπατούμε τα πνευματικά δικαιώματα του κατασκευαστή.

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;

Οι επιθέσεις άρνησης υπηρεσίας (DOS) είναι επιθέσεις που έχουν σαν σκοπό να καταστήσουν ένα υπολογιστή, υπηρεσία ή δίκτυο ανίκανο να εξυπηρετήσει άλλους χρήστες. Αυτό προσπαθούν να το επιτύχουν στέλνοντας χιλιάδες αιτήματα που υπερφορτώνουν τον διακομιστή ώστε να προκαλέσουν την κατάρρευση του ή εκμεταλλευόμενοι αδυναμίες του συστήματος.

Άμυνα στις επιθέσεις DOS προσφέρουν τα firewalls, συστήματα IPS, CDN (διανομή του περιεχομένου σε πολλούς servers σε διαφορετικές τοποθεσίες), η ενίσχυση της υποδομής, ο πλεονασμός (redundancy) κλπ.

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

Οι ιοί είναι κακόβουλα προγράμματα που μπορούν να αναπαράγονται μέσω των αποθηκευτικών μέσων ή του δικτύου και να μολύνουν άλλους υπολογιστές. Προκαλούν βλάβες στους υπολογιστές που μολύνουν διαγράφοντας αρχεία, υποκλέπτοντας δεδομένα κλπ.

Κύριες κατηγορίες:

Ιοί εκκίνησης: Προσβάλλουν το boot-sector του σκληρού δίσκου.

Ιοί αρχείων: Προσβάλλουν τα αρχεία του υπολογιστή.

Ιοί μακροεντολών: Είναι γραμμένοι σε μακροεντολές που χρησιμοποιούνται από προγράμματα όπως το Microsoft Office και προσβάλλουν τα έγγραφά τους.

Ιοί συστήματος ή μνήμης: Φορτώνονται στη μνήμη του υπολογιστή όταν εκκινείται το σύστημα και μολύνουν άλλα προγράμματα.

Πολυμορφικοί ιοί: Αυτοί αλλάζουν τον κώδικά τους όταν μολύνουν ένα νέο υπολογιστή καθιστώντας δύσκολη την ανίχνευσή τους.

Trojan horses: Αυτά δεν είναι ιοί (με την έννοια ότι δεν αναπαράγονται), εμφανίζονται σαν χρήσιμα προγράμματα αλλά στην πραγματικότητα εκτελούν κακόβουλες ενέργειες στο παρασκήνιο.

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

Οι αρχές πιστοποίησης είναι οργανισμοί που εκδίδουν ψηφιακά πιστοποιητικά τα οποία επιβεβαιώνουν την ταυτότητα ενός οργανισμού ή ενός ατόμου. Τα ψηφιακά πιστοποιητικά συνήθως χρησιμοποιούν κρυπτογράφηση για να βεβαιώσουν ότι μια ιστοσελίδα είναι αυθεντική και παρέχουν εμπιστοσύνη στις συναλλαγές μέσω διαδικτύου.

Η διαδικασία για την έκδοση πιστοποιητικού ενός ηλεκτρονικού καταστήματος είναι:

Αίτηση: Το κατάστημα υποβάλλει αίτηση σε μια αρχή πιστοποίησης. Η αίτηση περιλαμβάνει στοιχεία όπως το όνομα του καταστήματος, διεύθυνση URL κλπ.

Δημιουργία κλειδιών: Δημιουργείται ένα ζεύγος κλειδιών (ιδιωτικό και δημόσιο)

Αίτημα πιστοποίησης: Αποστέλλεται στην αρχή πιστοποίησης το δημόσιο κλειδί και τα στοιχεία του ιστότοπου.

Επαλήθευση: Η αρχή πιστοποίησης επαληθεύει τα στοιχεία του αιτούντα.

Έκδοση πιστοποιητικού: Η αρχή εκδίδει το ψηφιακό πιστοποιητικό αναφέροντας το δημόσιο κλειδί, τα στοιχεία του αιτούντα και τα στοιχεία της αρχής πιστοποίησης.

Εγκατάσταση του πιστοποιητικού: Ο διαχειριστής του καταστήματος εγκαθιστά το πιστοποιητικό στον ιστότοπο εξασφαλίζοντας έτσι ότι η μεταφορά δεδομένων και οι συναλλαγές θα είναι ασφαλείς και κρυπτογραφημένες.

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

Υπάρχει η κανονική και η γρήγορη διαμόρφωση.

Η κανονική διαμόρφωση δημιουργεί ένα κενό ευρετήριο αρχείων και ελέγχει κάθε τμήμα (sector) του δίσκου ότι είναι κατάλληλο για εγγραφή. Για το σκοπό αυτό γράφει δεδομένα και τα διαβάζει

πίσω επαληθεύοντας ότι γράφτηκαν σωστά. Αν υπάρχουν τμήματα που δεν επαληθεύτηκαν σωστά απενεργοποιούνται (bad sectors).

Η γρήγορη διαμόρφωση δημιουργεί μόνο ένα κενό ευρετήριο αρχείων απαλείφοντας έτσι τυχόν πληροφορίες που προϋπήρχαν.

Η κανονική διαμόρφωση προσφέρει μεγαλύτερη ασφάλεια γιατί είμαστε σίγουροι ότι όλα τα τμήματα του δίσκου είναι κατάλληλα προς εγγραφή. Επίσης αν προϋπήρχαν δεδομένα αυτά διαγράφονται οριστικά κατά την επαλήθευση του κάθε sector. Όμως διαρκεί πολύ περισσότερο χρόνο από τη γρήγορη διαμόρφωση, ιδιαίτερα σε σκληρούς δίσκους με μεγάλη χωρητικότητα.

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

Χρειαζόμαστε κεντρική μονάδα επεξεργασίας (CPU), μνήμη (RAM), μονάδα αποθήκευσης (hard disk), μονάδα εξόδου (οθόνη) και μονάδα εισόδου (πληκτρολόγιο, ποντίκι). Η διασύνδεση όλων αυτών επιτυγχάνεται με τη μητρική πλακέτα (motherboard). Επίσης χρειαζόμαστε ρεύμα, δηλαδή κάποιο τροφοδοτικό.

Η διασύνδεση CPU, RAM και motherboard επιτυγχάνεται με διαύλους επικοινωνίας ενώ για τις μονάδες εισόδου-εξόδου χρησιμοποιούνται καλώδια.

Τη μητρική πλακέτα, CPU, RAM, τροφοδοτικό τα τοποθετούμε σε κάποιο κουτί κατάλληλο για υπολογιστές ώστε να είναι προστατευμένα.

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

Κύριο έγγραφο: Είναι το αρχικό κείμενο που θα χρησιμοποιηθεί.

Αρχείο προέλευσης δεδομένων: Περιλαμβάνει όλα τα απαραίτητα στοιχεία για συγχώνευση με το κύριο έγγραφο. Μπορεί να είναι για παράδειγμα ένα αρχείο excel, μια απλή λίστα σ' ένα έγγραφο κειμένου (.txt) ή να το δημιουργήσουμε μέσα από το word.

Συγχωνευμένο έγγραφο: Είναι το αποτέλεσμα της ένωσης του κύριου εγγράφου με το αρχείο δεδομένων. Μπορούμε να το επεξεργαστούμε, εκτυπώσουμε ή αποθηκεύσουμε.

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

- Οικονομικές (Financial)
- Λογικές (Logical)
- Κειμένου (Text)
- Ημερομηνίας και Ώρας (Date & Time)
- Αναζήτησης και Αναφοράς (Lookup & Reference)
- Μαθηματικές και Τριγωνομετρικές (Math & Trig)
- Στατιστικές (Statistical)
- Μηχανολογικές (Engineering)
- Βάσης Δεδομένων (Database)
- Κύβου (Cube)
- Πληροφοριών (Information)

- Συναρτήσεις χρήστη (User defined)

Η συνάρτηση IF ελέγχει μια συνθήκη. Αν η συνθήκη είναι αληθής εκτελείται μια σειρά εντολών ενώ αν είναι ψευδής εκτελείται μια άλλη.

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

Διάταξη εκτύπωσης: Προβάλλει το έγγραφο όπως περίπου θα εκτυπωθεί στον εκτυπωτή.

Ανάγνωση Πλήρους οθόνης: Το έγγραφο καταλαμβάνει όλη την οθόνη χωρίς να εμφανίζονται τα μενού και η γραμμή κορδέλας του Word.

Διάταξη Web: Εμφανίζει το κείμενο σαν ιστοσελίδα χωρίς περιθώρια και άλλους περιορισμούς σελίδων.

Διάρθρωση: Παρέχει τη δυνατότητα να προσαρμόσουμε κατάλληλα τις επικεφαλίδες και να αναδιατάξουμε το περιεχόμενο.

Πρόχειρη: Η πρόχειρη μοιάζει με τη διάταξη εκτύπωσης αλλά εμφανίζει λιγότερα στοιχεία γραφικού περιβάλλοντος.

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

Οι αναφορές επιτρέπουν τη χρήση των δεδομένων ενός κελιού για τον υπολογισμό τιμών.

Υπάρχουν 3 είδη αναφορών:

Σχετικές αναφορές

Δηλώνονται με τη στήλη και τη γραμμή που βρίσκεται το κελί, πχ. A1. Αν αντιγράψουμε τον υπολογισμό που αναφέρεται σ' αυτό το κελί σε ένα άλλο κελί τότε και η αναφορά στο κελί θα προσαρμοστεί στην αντίστοιχη θέση. Πχ αν στο κελί D1 γράψουμε $= A1 * 2$ και στη συνέχεια αντιγράψουμε αυτό τον τύπο στο κελί E2 τότε το κελί θα έχει τιμή $= B2 * 2$

Απόλυτες αναφορές

Δηλώνονται με το σύμβολο \$ μπροστά από τη στήλη και γραμμή του κελιού, πχ. \$A\$1. Αν αντιγράψουμε τον υπολογισμό που αναφέρεται σ' αυτό το κελί τότε η αναφορά θα παραμείνει ίδια. Πχ αν στο κελί D1 γράψουμε $= \$A\$1 * 2$ και το αντιγράψουμε στο κελί E2 τότε το κελί θα έχει πάλι τιμή $= \$A\$1 * 2$.

Μικτές αναφορές

Βάζουμε το σύμβολο \$ μόνο στη στήλη ή τη γραμμή, πχ A\$1. Τότε θα παραμείνει σταθερή μόνο η στήλη ή γραμμή.

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

Τα βασικά χαρακτηριστικά είναι η ποιότητα εκτύπωσης (dpi), η ταχύτητα (ppm), η δυνατότητα έγχρωμης εκτύπωσης, η συνδεσιμότητα (USB, Wi-Fi, Ethernet).

Διακρίνουμε τις εξής κατηγορίες εκτυπωτών:

Ακίδων ή κρουστικοί (Dot Matrix)

Χρησιμοποιούν μια ακίδες που χτυπούν μια μελανοταινία επάνω στο χαρτί. Δεν έχουν καλή ποιότητα εκτύπωσης, είναι θορυβώδεις και δεν έχουν δυνατότητα εκτύπωσης γραφικών. Μπορούν όμως να εκτυπώσουν διπλότυπα με τη χρήση καρμπόν. Χρησιμοποιούνται συνήθως σε λογιστήρια.

Ψεκασμού (Inkjet)

Χρησιμοποιούν μικροσκοπικά ακροφύσια για να ψεκάσουν μελάνη στο χαρτί. Έχουν καλή ποιότητα ιδιαίτερα στις έγχρωμες εκτυπώσεις, είναι σχετικά αθόρυβοι αλλά αργοί. Χρησιμοποιούνται συνήθως από οικιακούς χρήστες λόγω του προσιτού κόστους τους. Έχουν ακριβό κόστος εκτύπωσης.

Laser

Έχουν πολύ καλή ποιότητα εκτύπωσης και κάποια μοντέλα προσφέρουν έγχρωμη εκτύπωση. Είναι αθόρυβοι, γρήγοροι με φθινό κόστος εκτύπωσης. Είναι ακριβότεροι από τους inkjet και χρησιμοποιούνται συνήθως από επιχειρήσεις.

Θερμικής μεταφοράς (Thermal printers)

Η κεφαλή τους ζεσταίνει στα κατάλληλα σημεία ένα ειδικό χαρτί κι έτσι σχηματίζονται οι χαρακτήρες. Είναι αθόρυβοι. Χρησιμοποιούνται σε ταμειακές μηχανές, εκτυπώσεις ετικετών κλπ.

Πολυμηχανήματα (Multifunction)

Μπορούν να είναι είτε ψεκασμού είτε laser. Προσφέρουν εκτός του εκτυπωτή και άλλες δυνατότητες όπως σάρωση, εκτύπωση φωτοτυπιών, φάξ κλπ.

Εκτυπωτές 3D

Δημιουργούν πραγματικά αντικείμενα συνήθως κατασκευασμένα από πλαστικό αλλά τα πιο ακριβά μοντέλα μπορούν να τυπώσουν μέταλλα ή ξύλο.

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

Ο κώδικας χαρακτήρων είναι ένα πρότυπο που αντιστοιχίζει ένα σύνολο (γράμματα, αριθμούς, σύμβολα) σε μοναδικές αριθμητικές τιμές ώστε να είναι δυνατή η επεξεργασία, μετάδοση και αποθήκευση τους από έναν υπολογιστή.

- Ο κώδικας ASCII χρησιμοποιεί 7 bit (8 bit ο Extended ASCII) και μπορεί να αντιστοιχίσει 127 χαρακτήρες σε αριθμητικές τιμές (ή 256 ο Extended ASCII). Έτσι αντιστοιχίζεται όλο το αγγλικό αλφάβητο, οι αριθμοί, κάποια σύμβολα και χαρακτήρες ελέγχου (αλλαγή γραμμής, tab κλπ).
- Ο κώδικας Unicode χρησιμοποιεί 8, 16 ή 32 bit (UTF-8, UTF-16, UTF-32), είναι η πιο σύγχρονη τυποποίηση και μπορεί να αντιστοιχίσει όλες τις γλώσσες του κόσμου.

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

Δημιουργία, Μετονομασία, Διαγραφή, Αντιγραφή, Μετακίνηση, Αλλαγή δικαιωμάτων πρόσβασης, Ανάγνωση, Επεξεργασία, Αποθήκευση.

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

Τα δεδομένα είναι ακατέργαστα στοιχεία ή γεγονότα που δεν έχουν αποκτήσει ακόμη κάποιο νόημα ή σημασία.

Η πληροφορία είναι το αποτέλεσμα της επεξεργασίας των δεδομένων που αποκτά πλέον νόημα και χρησιμοποιείται για λήψη αποφάσεων ή κατανόηση.

Παράδειγμα δεδομένων: Ημερήσιες θερμοκρασίες, 18, 22, 24, 19 κλπ.

Παράδειγμα πληροφορίας: Η μέση θερμοκρασία Νοεμβρίου ήταν 22 βαθμοί.

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

Η μνήμη RAM (Random Access Memory) ή Μνήμη Τυχαίας προσπέλασης είναι οι θέσεις μνήμης που καταλαμβάνει το λειτουργικό σύστημα και οι εφαρμογές που τρέχουμε. Στη RAM επίσης γίνεται ή επεξεργασία των δεδομένων. Τα περιεχόμενα της RAM σβήνονται όταν κλείσουμε τον υπολογιστή. Η μνήμη ROM (Read Only Memory) περιέχει συνήθως το BIOS του υπολογιστή. Δεν μπορούμε να γράψουμε στοιχεία στη ROM και τα περιεχόμενά της παραμένουν ακόμη και όταν σβήνουμε τον υπολογιστή. Στη ROM περιέχεται κώδικας που ελέγχει αν όλα τα συστήματα εκκίνησης λειτουργούν σωστά και στη συνέχεια αναλαμβάνει να φορτώσει το λειτουργικό σύστημα.

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

Η μνήμη cache είναι ταχύτερη μνήμη που βρίσκεται ανάμεσα στη CPU και τη μνήμη RAM (περιέχεται στο τσιπ του επεξεργαστή). Χρησιμοποιείται για να αποθηκεύει τα πρόσφατα δεδομένα ή τα δεδομένα που είναι πιθανόν να ζητηθούν σύντομα. Αν τα στοιχεία βρίσκονται στη μνήμη cache ο επεξεργαστής τα διαβάζει από εκεί και όχι από τη μνήμη RAM που είναι πολύ πιο αργή βελτιώνοντας έτσι την απόκριση του συστήματος.

Υπάρχουν 3 είδη cache L1, L2, L3 και αντίστοιχα κάθε μία έχει ταχύτερη προσπέλαση αλλά και μικρότερο μέγεθος.

Μνήμη cache επίσης χρησιμοποιούν και άλλα υποσυστήματα όπως ο σκληρός δίσκος.

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

Διαιρούμε συνεχώς τον αριθμό με το 2 μέχρις ότου να μη μπορεί να διαιρεθεί άλλο. Ο δυαδικός αριθμός προκύπτει από τα υπόλοιπα των διαιρέσεων με αντίστροφη σειρά.

120 | 2
 0 | 60 | 2
 0 | 30 | 2
 0 | 15 | 2
 1 | 7 | 2
 1 | 3 | 2
 1 | 1 | 2
 1 | 0

Επομένως $120_{(10)} = 1111000_{(2)}$

Για να μετατρέψουμε ένα δεκαδικό σε οκταδικό ακολουθούμε την ίδια διαδικασία αλλά διαιρούμε με 8.

120		8				
0		15		8		
		7		1		8
				1		0

Επομένως $120_{(10)} = 170_{(8)}$

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

Ο επεξεργαστής (CPU – Central Processing Unit) είναι ο εγκέφαλος ενός υπολογιστή. Επεξεργάζεται τα δεδομένα, και εκτελεί τις εντολές των προγραμμάτων.

Εκτελεί αριθμητικές και λογικές πράξεις, μεταφέρει δεδομένα από και προς τη μνήμη και ελέγχει την επικοινωνία των υποσυστημάτων.

Υπάρχουν επεξεργαστές γενικής χρήσης (πχ Intel Core i-9, AMD Ryzen), γραφικών (πχ Nvidia Geforce, AMD Radeon), κινητών τηλεφώνων (πχ Qualcomm Snapdragon), server (πχ Intel Xeon) κλπ.

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

Ο interpreter μεταφράζει τις εντολές του προγράμματος σε γλώσσα μηχανής κατά την εκτέλεση της εφαρμογής.

Ο compiler μεταφράζει πρώτα όλο τον κώδικα σε γλώσσα μηχανής και μετά εκτελείται η εφαρμογή. Σαν συνέπεια τα προγράμματα που μεταγλωττίζονται με interpreter είναι πιο αργά στη λειτουργία τους γιατί κάθε εντολή μεταφράζεται την ώρα που εκτελείται. Επίσης μπορεί να υπάρχουν λάθη στον κώδικα τα οποία επισημαίνονται κατά την εκτέλεση της εφαρμογής γιατί τότε μεταγλωττίζονται οι εντολές. Όμως το πρόγραμμα μεταγλωττίζεται πολύ πιο γρήγορα από ένα compiled και μπορούμε να δούμε άμεσα το αποτέλεσμα του κώδικά μας.

24. Ποιες είναι οι διαφορές ανάμεσα στην επαναληπτική εντολή "όσο....επανάλαβε" (while ... do) και την επαναληπτική εντολή "αρχή_επανάληψης...μέχρις_ότου" (repeat...until);

Στην πρώτη περίπτωση ο έλεγχος εκτέλεσης του βρόγχου γίνεται στην αρχή (while... do) ενώ στη δεύτερη (repeat... until) στο τέλος.

Αυτό σημαίνει ότι το while... do μπορεί να μην εκτελεστεί καθόλου ενώ το repeat until θα εκτελεστεί τουλάχιστον μία φορά.

Επίσης το while εκτελείται όσο η συνθήκη είναι αληθής ενώ το repeat μέχρις ότου η συνθήκη γίνει ψευδής.

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

Φυσική γλώσσα

Ο αλγόριθμος αναπαρίσταται στην κανονική ομιλούμενη γλώσσα. Είναι εύκολα κατανοητός από ανθρώπους που δεν γνωρίζουν προγραμματισμό αλλά μπορεί να οδηγήσει σε λάθη και παρερμηνείες όταν μετατραπεί σε κώδικα.

Ψευδογλώσσα

Ο αλγόριθμος γράφεται σε γλώσσα που μοιάζει με φυσική γλώσσα και προσομοιάζει τα βήματα μιας γλώσσας προγραμματισμού. Έτσι είναι εύκολο να γραφεί στη συνέχεια κώδικας σε οποιαδήποτε γλώσσα προγραμματισμού.

Διάγραμμα ροής

Αποτελείται από γεωμετρικά σχήματα και βέλη. Τοποθετώντας κατάλληλα τα σχήματα και ενώνοντας με τα βέλη αναπαριστούμε οπτικά τα βήματα εκτέλεσης του αλγόριθμου.

Προγραμματιστικός κώδικας

Γράφουμε τον αλγόριθμο κατευθείαν σε κάποια γλώσσα προγραμματισμού.

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

Τα συντακτικά λάθη είναι λάθη στη σύνταξη των εντολών και συνήθως εντοπίζονται από τον compiler πριν ξεκινήσει η μεταγλώττιση.

Τα λογικά λάθη είναι πιο δύσκολο να εντοπιστούν. Ενώ ο κώδικας είναι συντακτικά σωστός, η υλοποίηση του αλγόριθμου είναι εσφαλμένη.

```
int a = 10    // ΣΥΝΤΑΚΤΙΚΟ ΛΑΘΟΣ. ΛΕΙΠΕΙ ΤΟ ; ΣΤΟ ΤΕΛΟΣ ΤΗΣ ΕΝΤΟΛΗΣ

// Το παρακάτω είναι λογικό λάθος. Ο βρόγχος θα εκτελείται
// συνέχεια γιατί δεν αυξάνουμε το count μέσα στο while
int count = 5;
while (count < 10) {
    printf("%d", count);
}
```

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

Ο όρος LIFO απαντάται σε δομές στοίβας. Στις στοίβες τα δεδομένα προστίθενται στο τέλος της στοίβας και εξάγονται πάλι από το τέλος. Δηλαδή το στοιχείο που προστέθηκε τελευταίο θα εξαχθεί πρώτο.

Το FIFO απαντάται σε δομές ουράς. Στις ουρές τα δεδομένα προστίθενται στο τέλος της ουράς αλλά εξάγονται από την αρχή της. Δηλαδή εξάγουμε το στοιχείο που προστέθηκε πρώτο στην ουρά.

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

Οι στατικές δομές δεδομένων έχουν προκαθορισμένο μέγεθος που δηλώνεται κατά τη δημιουργία τους. Καταλαμβάνουν συνεχόμενες θέσεις μνήμης και η προσπέλαση τους είναι γρήγορη. Αν όμως δεν χρησιμοποιούμε όλο το προκαθορισμένο μέγεθος, σπαταλούμε μνήμη.

Οι δυναμικές δομές μπορούν να αλλάξουν μέγεθος κατά την εκτέλεση του κώδικα ανάλογα με τις ανάγκες του προγράμματος. Έτσι έχουμε βέλτιστη εκμετάλλευση μνήμης. Λόγω της επιπλέον διαχείρισης μνήμης κατά κανόνα η προσπέλαση τους είναι πιο αργή από τις στατικές δομές και επίσης έχουν μεγαλύτερη πολυπλοκότητα στην υλοποίηση.

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>
#include <stdlib.h>

int main()
{
    int n = 0;
    unsigned long long factorial = 1;

    printf("Enter a number ");
    scanf("%d", &n);

    if (n < 0)
    {
        printf("Number must be positive\n");
        return 1;
    }

    for (int i = 1; i <= n; i++)
    {
        factorial *= i;
    }
    printf("Factorial of %d is %llu\n", 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.

Βασικοί τύποι δεδομένων στη C:

char

Καταλαμβάνει 1 byte και παίρνει τιμές -128 έως 127

Παράδειγμα: char a = 'A';

int

Καταλαμβάνει 4 bytes και παίρνει τιμές $-256^4 / 2$ έως $256^4 / 2$

Παράδειγμα: int a = 10;

float

Καταλαμβάνει 4 bytes. Είναι δεκαδικός απλής ακρίβειας με 6 δεκαδικά ψηφία

Παράδειγμα: float a = 15.2;

double

Καταλαμβάνει 8 bytes. Είναι δεκαδικός διπλής ακρίβειας με 15 δεκαδικά ψηφία

_Bool (από τη C99 και μετά)

Καταλαμβάνει 1 byte και παίρνει τιμές true ή false

Παράδειγμα: _Bool a = true;

pointer

Χρησιμοποιούνται για να αποθηκεύουν διευθύνσεις μνήμης

Παράδειγμα: `int a = 10; int *ptr = &a;` (δείχνει τη διεύθυνση μνήμης που αποθηκεύτηκε η μεταβλητή `int a`).

Ο τύπος `int` μπορεί να συνοδεύεται από τροποποιητές `unsigned`, `long`, `long long` και ο τύπος `double` από `long` για να κρατούν μεγαλύτερους αριθμούς.

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. $35 = 3 * 3 * 3 * 3 * 3$ (5 φορές) β) Περιγράψτε σχηματικά πως δουλεύει για $\alpha=3$ και $\beta=5$
 Αναδρομική είναι μια συνάρτηση που καλεί τον εαυτό της. Μέσα στη συνάρτηση πρέπει να υπάρχει μια συνθήκη διακοπής ώστε να μην καλεί τον εαυτό της επ' άπειρο.

A)

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

B)

Η αναδρομικές συναρτήσεις λειτουργούν σαν στοίβες LIFO.

Στον πάτο της στοίβας μπαίνει η αρχική κλήση και από πάνω της με τη σειρά όλες οι επόμενες.

Για την εξαγωγή υπολογίζεται πρώτα η τελευταία κλήση, η οποία βγαίνει από τη στοίβα και περνάει το αποτέλεσμά της στην προηγούμενη κλήση.

ΣΕΙΡΑ ΕΙΣΑΓΩΓΗΣ	ΕΙΣΑΓΩΓΗ ΣΕ ΣΤΟΙΒΑ	ΣΕΙΡΑ ΕΞΑΓΩΓΗΣ	ΕΞΑΓΩΓΗ ΑΠΟ ΣΤΟΙΒΑ
5	<code>return 3</code>	1	3
4	<code>return 3 * power(3, 1)</code>	2	$3 * 3$
3	<code>return 3 * power(3, 2)</code>	3	$3 * (3 * 3)$
2	<code>return 3 * power(3, 3)</code>	4	$3 * (3 * 3 * 3)$
1	<code>return 3 * power(3, 4)</code>	5	$3 * (3 * 3 * 3 * 3)$
ΑΡΧΙΚΗ ΚΛΗΣΗ	<code>return power(3, 5)</code>	ΑΠΟΤΕΛΕΣΜΑ	$3 * 3 * 3 * 3 * 3$

32. Να γραφεί αλγόριθμος ο οποίος να διαβάζει τα ονόματα 10 πωλητών μιας εταιρείας και να τα αποθηκεύει σε ένα πίνακα Α, και τις αντίστοιχες πωλήσεις αυτών των πωλητών το 2005 και να τις αποθηκεύει σε ένα πίνακα Β. 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

Αλγόριθμος ΠΩΛΗΣΕΙΣ

```
min <- 1
max <- 1
sum <- 0
average <- 0
n <- 10
```

```
Για i Από 1 Μεχρι n
  Γράψε "Όνομα: "
  Διάβασε A[i]
  Γράψε "Πωλήσεις:"
  Διάβασε B[i]
```

```
sum <- sum + B[i]
```

```
!Αν η τιμή του στοιχείου είναι η μικρότερη
!ενημερώνουμε τον δείκτη προς το στοιχείο
Αν B[i] < B[min] Τότε
  min <- i
Τέλος_Αν
```

```
!Αν η τιμή του στοιχείου είναι η μεγαλύτερη
!ενημερώνουμε τον δείκτη προς το στοιχείο
Αν B[i] > B[max] Τότε
  max <- i
Τέλος_Αν
```

```
Τέλος_Επανάληψης
```

```
average <- sum / n
Εκτύπωσε "Μέσος όρος πωλήσεων", average
Εκτύπωσε "Χαμηλότερες πωλήσεις", A[min], B[min]
Εκτύπωσε "Υψηλότερες πωλήσεις", A[max], B[max]
```

```
Τέλος ΠΩΛΗΣΕΙΣ
```

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;
}
```

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

Α. Τυπώνει τους αριθμούς 50 έως 59 με τους αντίστοιχους λογαρίθμους τους.

Β. Επειδή το n παίρνει αρχική τιμή 50 και στη συνέχεια αυξάνεται, η συνθήκη $n \neq 40$ θα είναι πάντα αληθής. Επομένως θα τυπώνονται οι αριθμοί 50 έως άπειρο με τους αντίστοιχους λογαρίθμους.

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);
```

Σταδιακά οι μεταβλητές θα πάρουν τιμές:

$n = -3, i = 0, k = 0$

$x = 3.0, y = 0.0$

$i = -3, y = 7.0$

$k = 0$

$x = -2.0, n = -2$

$x = 6.0$

$n = -1, k = -1$

$y = -1\% -3 = -1.0$

Όταν $x < y$ τότε $x\%y = x$ (οι αρνητικοί αριθμοί δεν παίζουν ρόλο). Άρα $1 < 3$ και $-1\% -3 = -1$.

Οπότε θα τυπωθεί: -1 -3 -1 6.000000 -1.000000

Παρατήρηση:

Το modulus $a \% b$ επιστρέφει αν $a > b$ το υπόλοιπο της διαίρεσης a / b , αν $a < b$ επιστρέφει a .

Για αρνητικούς αριθμούς αγνοούμε τα πρόσημα και ισχύει το παραπάνω. Απλά αν $a < 0$ θα επιστραφεί αρνητικός αριθμός.

35. Να γραφεί πρόγραμμα που να δέχεται ένα διψήφιο ακέραιο αριθμό, και να επιστρέφει τον αριθμό ανεστραμμένο καθώς και το άθροισμα των ψηφίων του αριθμού (αν π.χ. ο χρήστης δώσει τον αριθμό 53, το πρόγραμμα πρέπει να επιστρέψει 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>
#include <stdlib.h>
```

```
int main()
{
    int num = 0;
```



```
do
{
    printf("Enter a 2-digit integer: ");
    scanf("%d", &num);
    if (num < 10 || num > 99)
    {
        printf("Number is not a 2-digit integer\n");
    }
} while (num < 10 || num > 99);

// Σπάννουμε τον αριθμό σε μονάδες και δεκάδες
// για να τον αντιστρέψουμε
int ones = num % 10;
int tens = num / 10;
int inverse = ones * 10 + tens;
int sum = ones + tens;

printf("Inverse of %d is %d\n", num, inverse);
printf("%d has digit sum of %d\n", num, sum);

return 0;
}
```

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

Η quick sort ακολουθεί την τεχνική διαίρει και βασίλευε για να σπάσει το πρόβλημα σε μικρότερα υποπροβλήματα.

Υπάρχουν 3 κύρια στάδια στην υλοποίηση του αλγόριθμου:

1. Επιλογή pivot

Επιλέγουμε ένα οποιοδήποτε στοιχείο του πίνακα σαν pivot.

2. Χωρισμός πίνακα

Χωρίζουμε τον πίνακα σε 2 υποπίνακες έχοντας σαν σημείο αναφοράς το pivot. Βάζουμε τα στοιχεία που είναι μικρότερα από το pivot στον αριστερό πίνακα και όσα είναι μεγαλύτερα στον δεξιό. Έτσι το pivot θα είναι ταξινομημένο στη σωστή θέση.

3. Αναδρομική κλήση

Τα βήματα 1 και 2 καλούνται αναδρομικά στους 2 υποπίνακες έως ότου οι υποπίνακες περιέχουν ένα ή κανένα στοιχείο.

Στο παρακάτω πρόγραμμα C επιλέγουμε σαν pivot το τελευταίο στοιχείο του πίνακα.

Το βήμα 3 υλοποιείται με τη συνάρτηση quickSort και το βήμα 2 με την arrangePartition.

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

```
// ΑΝΤΑΛΛΑΞΕΙ ΤΙΣ ΤΙΜΕΣ 2 ΣΤΟΙΧΕΙΩΝ
void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

```
// Επιλέγει το τελευταίο στοιχείο του πίνακα σαν pivot
// Επιστρέφει το στοιχείο που θα είναι το νέο pivot
int partition(int arr[], int low, int high)
{
    // Δείκτης για τα μικρότερα στοιχεία.Υποδεικνύει την προσωρινή
    // θέση pivot (βρίσκεται στα δεξιά του) που έχει βρεθεί μέχρι τώρα
    int index = low - 1;
    int pivot = high;

    // Διατρέχουμε τον πίνακα arr[low...high] και μετακινούμε όλα τα
    // μικρότερα στοιχεία στην αριστερή μεριά. Τα στοιχεία από το
    // low έως το index θα είναι τα μικρότερα σε κάθε επανάληψη
    for (int i = low; i < pivot; i++)
    {
        if (arr[i] < arr[pivot])
        {
            index++;
            swap(&arr[index], &arr[i]);
        }
    }

    // Μεταφέρουμε το pivot μετά από όλα τα μικρότερα στοιχεία
    // και επιστρέφουμε τη θέση του
    swap(&arr[index + 1], &arr[pivot]);
    pivot = index + 1;
    return pivot;
}

void quickSort(int arr[], int low, int high)
{
    if (low < high)
    {
        // Βρίσκουμε το pivot
        int pivot = partition(arr, low, high);

        // Αναδρομικές κλήσεις για τα υποδιαστήματα
        quickSort(arr, low, pivot - 1); //Υποδιάστημα αριστερά του pivot
        quickSort(arr, pivot + 1, high); //Υποδιάστημα δεξιά του pivot
    }
}

// Εκτυπώνει απλά τον πίνακα (δεν χρησιμοποιείται από την quickSort)
void printArray(int arr[], int len)
{
    for (int i = 0; i < len; i++)
    {
        printf("%d\t", arr[i]);
    }
    printf("\n");
}

// Παράδειγμα χρήσης της quickSort
int main()
{
    int array[] = {10, 30, 80, 70, 60, 90};
```

```
int len = sizeof(array) / sizeof(array[0]);  
printf("Original array:\n");  
printArray(array, len);  
printf("Sorted with quick sort:\n");  
quickSort(array, 0, len - 1);  
printArray(array, len);  
return 0;  
}
```

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

Στη selection sort διατρέχουμε όλα τα στοιχεία του πίνακα και βάζουμε στην αρχή το μικρότερο στοιχείο. Επαναλαμβάνουμε για τα επόμενα στοιχεία του πίνακα. Στο τέλος ο πίνακας θα είναι ταξινομημένος:

0-> **6** 7 9 **3** 5 8 // Αλλαγή θέσης 6 - 3
1-> 3 **7** 9 **6** 5 8 // Αλλαγή θέσης 7 - 5
2-> 3 5 **9** **6** 7 8 // Αλλαγή θέσης 9 - 6
3-> 3 5 6 **9** **7** 8 // Αλλαγή θέσης 9 - 7
4-> 3 5 6 7 **9** **8** // Αλλαγή θέσης 9 - 8
5-> 3 5 6 7 8 9

Στη bubble sort συγκρίνουμε κάθε στοιχείο με το επόμενο. Αν είναι μεγαλύτερο αντιμετωπίζουμε τη θέση τους. Έτσι στο τέλος θα υπάρχει ο μεγαλύτερος αριθμός. Επαναλαμβάνουμε για τα υπόλοιπα στοιχεία εκτός του τελευταίου. Έτσι τώρα ο προτελευταίος θα είναι στη θέση του. Αν ο πίνακας έχει N στοιχεία μετά από N-1 επαναλήψεις θα είναι ταξινομημένα.

0 -> 6 7 9 3 5 8 // Αρχικός πίνακας. Αλλαγή θέσης 9-3, 9-5, 9-8
1 -> 6 7 3 5 8 9 // Αλλαγή θέσης 7-3, 7-5
2 -> 6 3 5 7 8 9 // Αλλαγή θέσης 6-3, 6-5
3 -> 3 5 6 7 8 9
4 -> 3 5 6 7 8 9
5 -> 3 5 6 7 8 9 // Ταξινομημένος πίνακας

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>  
#include <stdlib.h>  
  
int main()  
{  
    int len = 5;  
    int arr[len][len];  
  
    for (int i = 0; i < len; i++)  
    {  
        for (int j = 0; j < len; j++)
```

```
{
    printf("Enter array element [%d][%d]: ", i, j);
    scanf("%d", &arr[i][j]);
}

// Εκτύπωση όλων των στοιχείων του πίνακα
printf("Array elements:\n");
for (int i = 0; i < len; i++)
{
    for (int j = 0; j < len; j++)
    {
        printf("%d\t", arr[i][j]);
    }
    printf("\n");
}

// Εκτύπωση διαγωνίου (πάνω αριστερά προς κάτω δεξιά)
printf("Main diagonal:\n");
for (int i = 0; i < len; i++)
{
    printf("%d\t", arr[i][i]);
}
printf("\n");

// Εκτύπωση ανάστροφης διαγωνίου (πάνω δεξιά προς κάτω αριστερά)
printf("Reverse diagonal:\n");
for (int i = 0; i < len; i++)
{
    printf("%d\t", arr[i][len - 1 - i]);
}
printf("\n");

return 0;
}
```

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

Αλγόριθμος είναι μια ακολουθία πεπερασμένων ενεργειών που επιλύουν ένα πρόβλημα. Ένας αλγόριθμος δεν είναι απαραίτητο να είναι γραμμένος σε κάποια γλώσσα προγραμματισμού. Πρόγραμμα είναι μια ακολουθία εντολών σε κάποια γλώσσα προγραμματισμού που μπορεί να περιλαμβάνει πολλούς αλγόριθμους και παράγει ένα εκτελέσιμο αρχείο στον υπολογιστή.

40. Τι ονομάζουμε δομή δεδομένων;

Δομή δεδομένων ονομάζουμε έναν οργανωμένο τρόπο διαχείρισης και αποθήκευσης δεδομένων ώστε να είναι εύκολη η πρόσβαση, προσθήκη, επεξεργασία και διαγραφή τους.

Παραδείγματα δομών δεδομένων είναι οι πίνακες, συνδεδεμένες λίστες, στοίβες, ουρές κλπ.

Οι δομές δεδομένων μαζί με τους αλγόριθμους είναι θεμελιώδεις στην ανάπτυξη λογισμικού. Οι δομές δεδομένων καθορίζουν τον τρόπο που θα οργανωθούν τα δεδομένα ενώ οι αλγόριθμοι καθορίζουν τον τρόπο με τον οποίο θα επεξεργαστούν.

- 41. Σε ποιες κατηγορίες διακρίνονται τα προβλήματα με κριτήριο το είδος επίλυσης τους;**
Προβλήματα απόφασης: Αποφασίζουν αν μια δήλωση είναι αληθής ή ψευδής.
Προβλήματα υπολογισμού: Βρίσκουν ένα αποτέλεσμα με την εκτέλεση υπολογιστικών πράξεων.
Προβλήματα βελτιστοποίησης: Αναζητούν τη βέλτιστη λύση με βάση καθορισμένα κριτήρια.
- 42. Ποιες είναι οι βασικές λειτουργίες (ή πράξεις) επί των δομών δεδομένων;**
Εισαγωγή: Εισαγωγή ενός στοιχείου στη δομή.
Διαγραφή: Διαγραφή ενός στοιχείου.
Ενημέρωση: Τροποποίηση ενός στοιχείου που υπάρχει ήδη στη δομή δεδομένων.
Πρόσβαση: Ανάκτηση ενός στοιχείου από τη δομή χωρίς να το τροποποιήσουμε.
Αναζήτηση: Αναζήτηση ενός στοιχείου της δομής δεδομένων.
Ταξινόμηση: Ταξινόμηση της δομής με βάση προκαθορισμένα κριτήρια.
Συγχώνευση: Ένωση δύο ή περισσότερων δομών σε μία ενιαία δομή.
Διαχωρισμός: Διάσπαση μιας δομής δεδομένων σε μικρότερες.
- 43. Τι εννοούμε με τον όρο "πρόβλημα" και σε τι αναφερόμαστε με τον όρο "Ανάλυση Προβλήματος";**
Πρόβλημα είναι μια κατάσταση που πρέπει να επιλυθεί ή δε λύση της δεν είναι γνωστή ούτε προφανής.
Με τον όρο Ανάλυση Προβλήματος εννοούμε τον χωρισμό του προβλήματος σε μικρότερα υποπροβλήματα που είναι ευκολότερο να επιλυθούν. Με τον τρόπο αυτό κατανοούμε καλύτερα τη δομή του προβλήματος και είναι ευκολότερη η επίλυσή του.
- 44. Να αναφέρετε τις πιο συνηθισμένες τεχνικές σχεδίασης αλγορίθμων.**
Διαιρεί και βασίλευε: Χωρίζουμε το πρόβλημα σε μικρότερα υποπροβλήματα, επιλύουμε τα υποπροβλήματα και συνδυάζουμε τις λύσεις τους για να δώσουμε τη συνολική λύση.
Δυναμικός προγραμματισμός: Χωρίζουμε το πρόβλημα σε επαναλαμβανόμενα υποπροβλήματα και αποθηκεύουμε τις λύσεις τους ώστε να μη χρειάζεται να τις υπολογίσουμε εκ νέου.
Άπληστη μέθοδος: Σε κάθε βήμα του αλγόριθμου προσπαθούμε να βρούμε τη βέλτιστη λύση ελπίζοντας ότι στο τέλος η συνολική λύση θα είναι βέλτιστη.
- 45. Ποια είναι τα χαρακτηριστικά ενός αλγορίθμου;**
Είσοδος: Μπορεί να δέχεται από μηδέν έως πολλά δεδομένα στην είσοδο.
Έξοδος: Πρέπει να επιστρέφει τουλάχιστον ένα αποτέλεσμα κατά την έξοδο.
Καθοριστικότητα: Κάθε βήμα του αλγόριθμου πρέπει να είναι σαφώς καθορισμένο.
Περατότητα: Ο αλγόριθμος πρέπει να ολοκληρώνεται σε πεπερασμένο αριθμό βημάτων.
Αποτελεσματικότητα: Κάθε εντολή του αλγόριθμου πρέπει να είναι απλή και εκτελέσιμη.
Γενικότητα: Πρέπει να επιλύει μια ολόκληρη κατηγορία προβλημάτων και όχι μόνο μια συγκεκριμένη περίπτωση.
Σωστή λειτουργία: Ο αλγόριθμος πρέπει να παράγει σωστό αποτέλεσμα για οποιαδήποτε είσοδο.

46. Με ποιους τρόπους πραγματοποιείται η περιγραφή ενός αλγορίθμου;

Φυσική γλώσσα

Χρησιμοποιούμε μια ομιλούμενη γλώσσα για να περιγράψουμε τα βήματα του αλγόριθμου. Με τον τρόπο αυτό ο αλγόριθμος γίνεται εύκολα κατανοητός από ανθρώπους που δεν γνωρίζουν προγραμματισμό αλλά μπορεί να οδηγήσει σε λάθη ή παρερμηνείες κατά την υλοποίηση του.

Ψευδογλώσσα

Χρησιμοποιούμε μια γλώσσα που μοιάζει με γλώσσα προγραμματισμού αλλά δεν είναι. Έτσι τα βήματα του αλγόριθμου είναι σαφώς καθορισμένα και μπορεί να γραφεί στη συνέχεια σε κάποια κατάλληλη γλώσσα προγραμματισμού.

Διάγραμμα ροής

Χρησιμοποιούμε γεωμετρικά σύμβολα για να οπτικοποιήσουμε τα βήματα του αλγόριθμου.

Ενώνουμε τα βήματα μεταξύ τους με βέλη. Έτσι ο αλγόριθμος γίνεται εύκολα οπτικά κατανοητός.

Προγραμματιστικά

Γράφουμε απευθείας τις εντολές του αλγόριθμου σε κάποια γλώσσα προγραμματισμού.

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

Είσοδος: Μπορεί να δέχεται από μηδέν έως πολλά δεδομένα στην είσοδο.

Έξοδος: Πρέπει να επιστρέφει τουλάχιστον ένα αποτέλεσμα κατά την έξοδο.

Καθοριστικότητα: Κάθε βήμα του αλγόριθμου πρέπει να είναι σαφώς καθορισμένο.

Περατότητα: Ο αλγόριθμος πρέπει να ολοκληρώνεται σε πεπερασμένο αριθμό βημάτων.

Αποτελεσματικότητα: Κάθε εντολή του αλγόριθμου πρέπει να είναι απλή και εκτελέσιμη.

Γενικότητα: Πρέπει να επιλύει μια ολόκληρη κατηγορία προβλημάτων και όχι μόνο μια συγκεκριμένη περίπτωση.

Σωστή λειτουργία: Ο αλγόριθμος πρέπει να παράγει σωστό αποτέλεσμα για οποιαδήποτε είσοδο.

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

Απλή δομή επιλογής: Εκτελείται μια σειρά εντολών αν μία συνθήκη είναι αληθής. Υλοποιείται με την εντολή IF... THEN...

Σύνθετη δομή επιλογής: Εκτελείται μια σειρά εντολών αν μία συνθήκη είναι αληθής αλλιώς εκτελείται μια άλλη σειρά εντολών. Υλοποιείται με την εντολή IF... THEN... ELSE...

Πολλαπλή δομή επιλογής: Ανάλογα με την τιμή μιας συνθήκης εκτελείται η αντίστοιχη επιλογή. Υλοποιείται με τις εντολές IF... THEN... ELSEIF... ELSE... ή SWITCH/CASE.

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

Έχουμε 3 κύριες δομές επανάληψης:

- Δομή επανάληψης όπου ο έλεγχος επανάληψης βρίσκεται στην αρχή της δομής:
WHILE (έλεγχος) {εκτέλεση εντολών}
- Δομή επανάληψης όπου ο έλεγχος επανάληψης βρίσκεται στο τέλος της δομής:
DO {εκτέλεση εντολών} WHILE (έλεγχος) ή REPEAT {εκτέλεση εντολών} UNTIL (έλεγχος)
- Δομή επανάληψης με προκαθορισμένο αριθμό επαναλήψεων:
FOR (αριθμός επαναλήψεων) {εκτέλεση εντολών}

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

Η στοίβα είναι μια δομή δεδομένων LIFO (Last In First Out). Τα δεδομένα εισάγονται στο τέλος της στοίβας το ένα μετά το άλλο. Μπορούμε να διαβάσουμε μόνο το στοιχείο που προστέθηκε τελευταίο, δηλαδή αυτό που βρίσκεται στην κορυφή της στοίβας. Οι βασικές της λειτουργίες είναι:

push: Εισάγει ένα στοιχείο στο τέλος της στοίβας.

pop: Επιστρέφει και αφαιρεί το τελευταίο στοιχείο της στοίβας.

isEmpty: Ελέγχει αν η στοίβα είναι άδεια.

isFull: Ελέγχει αν η στοίβα είναι γεμάτη.

Παράδειγμα: Θέλουμε να αναστρέψουμε τη συμβολοσειρά "ΑΒΓ". Για το σκοπό αυτό εισάγουμε ένα-ένα τα στοιχεία της σε μια στοίβα και τα διαβάζουμε ξανά από τη στοίβα.

```
push('A')
```

```
push('B')
```

```
push('Γ') // Η στοίβα τώρα περιέχει Α, Β, Γ
```

```
pop()    // Επιστρέφει Γ
```

```
pop()    // Επιστρέφει Β
```

```
pop()    // Επιστρέφει Α
```

Έτσι μετατρέπουμε την 'ΑΒΓ' σε 'ΓΒΑ'

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

Αναζήτηση:

- Σειριακή ή Γραμμική Αναζήτηση (Linear Search): Η απλούστερη μορφή αναζήτησης. Χρησιμοποιείται σε αταξινομήτους πίνακες. Διατρέχουμε ένα-ένα τα στοιχεία του πίνακα μέχρι να βρούμε αυτό που αναζητούμε.
- Δυαδική Αναζήτηση (Binary Search): Χρησιμοποιείται σε ταξινομημένους πίνακες. Βασίζεται στον αλγόριθμο Διαιρεί και Βασίλευε. Συγκρίνουμε το στοιχείο που αναζητούμε με το μεσαίο στοιχείο του πίνακα. Αν είναι μεγαλύτερο χωρίζουμε τον πίνακα και επαναλαμβάνουμε για το δεξιό τμήμα του (μέση έως τέλος), ενώ αν είναι μικρότερο για το αριστερό τμήμα του (αρχή έως μέση). Συνεχίζουμε τη διαίρεση μέχρι να βρούμε το στοιχείο ή να μη μπορεί να διαιρεθεί άλλο ο πίνακας οπότε το στοιχείο δεν υπάρχει στον πίνακα.
- Αναζήτηση Παρεμβολής (Interpolation Search): Δεν ψάχνουμε όπως στη δυαδική αναζήτηση στη μέση, μετά στο $\frac{1}{2}$, μετά στο $\frac{1}{4}$ κλπ. Μοιάζει με την αναζήτηση σε ένα λεξικό. Αν ψάχνουμε για παράδειγμα μια λέξη που αρχίζει από Α ανοίγουμε το λεξικό στην αρχή, Αν αρχίζει από Ω ανοίγουμε στο τέλος ενώ αν αρχίζει από Κ ανοίγουμε κάπου στη μέση.

Ταξινόμηση:

- Ταξινόμηση Επιλογής (Selection Sort): Διατρέχουμε τον πίνακα και τοποθετούμε το μικρότερο στοιχείο στην αρχή. Επαναλαμβάνουμε για τα υπόλοιπα στοιχεία.
- Ταξινόμηση Φυσαλίδας (Bubble Sort): Συγκρίνουμε τα στοιχεία ανά ζευγάρια και τοποθετούμε το μεγαλύτερο στοιχείο στο τέλος του ζευγαριού. Επαναλαμβάνουμε για όλα τα στοιχεία του πίνακα.

- Ταξινόμησης Εισαγωγής (Insertion Sort): Εισάγει ένα στοιχείο στη σωστή θέση σε ένα ήδη ταξινομημένο πίνακα.
- Ταξινόμηση Συγχώνευσης (Merge Sort): Διαιρούμε τον πίνακα σε δύο τμήματα. Ταξινομούμε τα τμήματα αναδρομικά και μετά συγχωνεύουμε ξανά τα τμήματα.
- Γρήγορη Ταξινόμηση (Quick Sort): Παίρνουμε ένα στοιχείο του πίνακα σαν σημείο αναφοράς (Pivot Point). Μεταφέρουμε τα μεγαλύτερα στοιχεία δεξιά του και τα μικρότερα αριστερά. Επαναλαμβάνουμε για τα τμήματα που δημιουργήθηκαν.
- Ταξινόμηση Σωρού (Heap Sort): Χρησιμοποιούμε μια δομή σωρού για την τοποθέτηση των στοιχείων σε σωστή σειρά.

52. Τι είναι RAM και πώς είναι δομημένη;

Η RAM (Random Access Memory) ή Μνήμη Τυχαίας Προσπέλασης είναι η μνήμη που κρατούνται τα δεδομένα και τα προγράμματα προς εκτέλεση. Από τη RAM τα δεδομένα μεταφέρονται στη CPU για επεξεργασία. Επίσης από τη RAM μεταφέρονται οι εντολές των προγραμμάτων στη CPU προς εκτέλεση. Τα περιεχόμενα της RAM διαγράφονται όταν κλείσουμε τον υπολογιστή.

Η RAM είναι μνήμη τυχαίας προσπέλασης, δηλαδή μπορούμε να απευθυνθούμε κατευθείαν σε οποιαδήποτε θέση μνήμης χωρίς να απαιτείται σειριακή αναζήτηση.

Διακρίνουμε δύο είδη RAM. Τη Δυναμική (DRAM) και τη Στατική (SRAM).

Η SRAM χρησιμοποιεί flip-flop κυκλώματα για αποθήκευση των στοιχείων. Είναι γρηγορότερη αλλά πιο ακριβή και απαιτεί μεγαλύτερη ενέργεια. Χρησιμοποιείται σαν μνήμη cache.

Η DRAM χρησιμοποιεί πυκνωτές για τη αποθήκευση των στοιχείων και χρειάζεται συνεχώς ανανέωση (refresh). Είναι ο πιο κοινός τύπος RAM γιατί είναι φθηνή. Με την πάροδο του χρόνου εμφανίζονται γρηγορότερες τέτοιες μνήμες με μικρότερη κατανάλωση και μεγαλύτερη χωρητικότητα. Έτσι έχουμε διάφορες γενιές DDR, DDR2, DDR3, DDR4 κλπ.

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

- Καταχωρητές Γενικής Χρήσης και Δεδομένων (General Purpose Registers): Χρησιμοποιούνται για την εκτέλεση αριθμητικών πράξεων και τη μεταφορά δεδομένων.
- Καταχωρητές Διευθύνσεων (Address Registers): Αποθηκεύουν διευθύνσεις μνήμης για την προσπέλαση δεδομένων.
- Καταχωρητές Δείκτη Εντολών (Instruction Pointer): Περιέχουν τη διεύθυνση μνήμης της επόμενης εντολής που θα εκτελεστεί.
- Καταχωρητές Κατάστασης (Status ή Flag Registers): Περιέχουν τιμές που δείχνουν την κατάσταση της CPU ή τα αποτελέσματα προηγούμενων πράξεων.
- Καταχωρητές Κινητής Υποδιαστολής (Floating Point Registers): Αποθηκεύουν δεδομένα για τις πράξεις αριθμών με δεκαδικά ψηφία.
- Καταχωρητές Τμήματος (Segment Registers): Χρησιμοποιούνται για την αποθήκευση μιας περιοχής μνήμης.

Καταχωρητές Γενικής Χρήσης:

- AX (Συσσωρευτής – Accumulator)
- BX (Βάσης – Base)
- CX (Μετρητής – Counter)
- DX (Δεδομένων – Data)

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

Το bit είναι η μικρότερη μονάδα πληροφορίας στο δυαδικό σύστημα. Μπορεί να πάρει τιμές 0 ή 1. Το byte αποτελείται από 8 bits και μπορεί να πάρει τιμές από -128 έως 128 ή 0-255. Είναι η βασική μονάδα αποθήκευσης στους περισσότερους υπολογιστές.

Το word εξαρτάται από την αρχιτεκτονική του επεξεργαστή και μπορεί να έχει μέγεθος 16-bit, 32-bit ή 64-bit (δηλαδή 2, 4 ή 8 bytes αντίστοιχα).

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

Άμεση Διευθυνσιοδότηση

Απευθυνόμαστε άμεσα στη διεύθυνση που περιέχει τα δεδομένα.

Παράδειγμα: MOV AX, [1000]

Το περιεχόμενο της διεύθυνσης μνήμης 1000 φορτώνεται στον καταχωρητή AX.

Έμμεση Διευθυνσιοδότηση

Απευθυνόμαστε στη διεύθυνση μνήμης που περιέχει τη διεύθυνση μνήμης των δεδομένων.

Παράδειγμα: MOV AX, [BX]

Το περιεχόμενο της διεύθυνσης μνήμης που υπάρχει στον καταχωρητή BX φορτώνεται στον καταχωρητή AX.

Σχετική Διευθυνσιοδότηση

Απευθυνόμαστε σε μια διεύθυνση σχετική με τη διεύθυνση μνήμης που υπάρχει στον καταχωρητή.

Παράδειγμα: MOV SI, 4

MOV BX, 1000h

MOV AX, [BX+SI]

Φορτώνεται στον καταχωρητή AX η τιμή της διεύθυνσης 1004.

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

Από την αρχιτεκτονική του υπολογιστή, το λειτουργικό σύστημα και τις απαιτήσεις των εφαρμογών.

Σε συστήματα με μικρή μνήμη χρησιμοποιούνται τεχνικές εικονικής μνήμης και τα δεδομένα αποθηκεύονται προσωρινά στο σκληρό δίσκο ώστε να είναι δυνατή η επεξεργασία περισσότερων δεδομένων από τη φυσική διαθέσιμη μνήμη.

Το λειτουργικό σύστημα είναι υπεύθυνο για τη διαχείριση μνήμης. Σε κάποιες απαιτητικές εφαρμογές ή εφαρμογές που χρησιμοποιούνται συχνά μπορεί να κρατάει τα δεδομένα σε μνήμη cache ώστε να είναι ταχύτερη η εκτέλεση τους.

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

Από την αρχιτεκτονική του επεξεργαστή και το λειτουργικό σύστημα.

Για παράδειγμα ένας επεξεργαστής 32-bit με λειτουργικό σύστημα 32-bit μπορεί να απευθυνθεί σε 2^{32} θέσεις μνήμης (4GB). Ένας επεξεργαστής και λειτουργικό 64-bit μπορεί να απευθυνθεί σε 2^{64} θέσεις μνήμης.

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

Γιατί μπορεί να φορτωθούν στη μνήμη περισσότερες ή μεγαλύτερες εφαρμογές και δεδομένα αποφεύγοντας έτσι τη χρήση εικονικής μνήμης που είναι πολύ πιο αργή από τη φυσική μνήμη RAM. Με τον τρόπο αυτό επιταχύνεται η εκτέλεση εφαρμογών αλλά και γενική απόκριση του συστήματος.

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

Αριθμητική και Λογική Μονάδα (ALU): Στο τμήμα αυτό εκτελούνται οι αριθμητικές και λογικές πράξεις.

Καταχωρητές (Registers): Είναι κελιά μνήμης στο εσωτερικό του επεξεργαστή για τη φύλαξη των δεδομένων ή εντολών προς επεξεργασία.

Μονάδα Ελέγχου και Χρονισμού (Control Unit and Timing): Ελέγχει και συντονίζει την επικοινωνία με τις μονάδες εισόδου/εξόδου.

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

Το ψηφίο ισοτιμίας (parity bit) προστίθεται σε ένα μπλοκ δεδομένων για να ελέγξουμε αν η πληροφορία που μεταδίδεται έχει μεταφερθεί σωστά. Ένα bit μπορεί να έχει τιμή 0 ή 1. Στον έλεγχο ισοτιμίας μετράμε τα bit που έχουν τιμή 1.

Έχουμε 2 είδη ισοτιμίας:

Even-parity: Αν στο μπλοκ δεδομένων τα bit με τιμή 1 είναι μονός αριθμός, προσθέτουμε ένα bit με τιμή 1 ώστε τα συνολικά 1-bit μαζί με το parity να είναι ζυγά (αλλιώς προσθέτουμε 0).

Odd-parity: Αν στο μπλοκ δεδομένων τα bit με τιμή 1 είναι ζυγός αριθμός, προσθέτουμε ένα bit με τιμή 1 ώστε τα συνολικά 1-bit μαζί με το parity να είναι μονά (αλλιώς προσθέτουμε 0).

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

Υπάρχει η κανονική διαμόρφωση και η γρήγορη διαμόρφωση.

Στην κανονική διαμόρφωση δημιουργείται ένα άδειο ευρετήριο αρχείων και ελέγχεται κάθε τμήμα (sector) του δίσκου ότι είναι κατάλληλο για εγγραφή δεδομένων.

Στη γρήγορη διαμόρφωση δημιουργείται μόνο ένα άδειο ευρετήριο αρχείων.

Η κανονική διαμόρφωση δίνει μεγαλύτερη ασφάλεια καθώς όλα τα τμήματα (sectors) του δίσκου έχουν ελεγχθεί ότι είναι κατάλληλα για την αποθήκευση δεδομένων. Όμως είναι πολύ πιο αργή από τη γρήγορη διαμόρφωση και ιδιαίτερα σε δίσκους με μεγάλη χωρητικότητα η διαδικασία μπορεί να διαρκέσει πολλές ώρες.

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

Το I/O bus είναι ένας δίαυλος ή σύστημα καλωδίων που επιτρέπει την επικοινωνία του επεξεργαστή με τη μνήμη και τις μονάδες εισόδου/εξόδου.

Το interrupt είναι ένα σήμα που αποστέλλεται από μια περιφερειακή συσκευή ή λογισμικό για να ενημερώσει τον επεξεργαστή ότι είναι έτοιμη να στείλει ή να δεχθεί δεδομένα. Σκοπός του interrupt είναι να διακοπεί η κανονική ροή εκτέλεσης του επεξεργαστή ώστε να διαχειριστεί ένα γεγονός.

Ο controller είναι μια συσκευή ή ένα ηλεκτρονικό κύκλωμα που λειτουργεί ως διαμεσολαβητής. Ελέγχει την επικοινωνία και τη μεταφορά δεδομένων μεταξύ του επεξεργαστή και των συσκευών εισόδου/εξόδου.

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

Το ρολόι ενός υπολογιστή είναι ένα ηλεκτρονικό κύκλωμα που παράγει ηλεκτρικούς παλμούς. Είναι απαραίτητο για να συντονίζεται η επικοινωνία μεταξύ των τμημάτων του επεξεργαστή, της μνήμης και των μονάδων εισόδου/εξόδου.

Τα βασικά χαρακτηριστικά του ρολογιού είναι η συχνότητα (clock frequency) δηλαδή πόσοι παλμοί δημιουργούνται το δευτερόλεπτο και ο κύκλος (clock cycle) που καθορίζει το διάστημα μεταξύ των παλμών.

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

Χωρίς έλεγχο περιεχομένου:

```
MOV AH, 01h      ; Υπορουτίνα για εισαγωγή χαρακτήρα
INT 21h          ; Κλήση διακοπής του DOS
MOV [CHAR], AL   ; Αποθήκευση του χαρακτήρα στη μνήμη
```

Με έλεγχο περιεχομένου:

```
MOV AH, 01h      ; Υπορουτίνα για εισαγωγή χαρακτήρα
INT 21h          ; Κλήση διακοπής του DOS
CMP AL, '0'      ; Έλεγχος αν είναι μικρότερος από '0'
JB INVALID_INPUT ; Αν ναι, μετάβαση σε σήμανση λάθους
CMP AL, '9'      ; Έλεγχος αν είναι μεγαλύτερος από '9'
JA INVALID_INPUT ; Αν ναι, μετάβαση σε σήμανση λάθους
```

```
; Αν ο χαρακτήρας είναι αριθμητικός, μετατροπή σε αριθμό
SUB AL, '0'      ; Μετατροπή ASCII χαρακτήρα σε ακέραιο
MOV [DIGIT], AL  ; Αποθήκευση του αριθμού στη μνήμη
```

```
INVALID_INPUT:
; Κώδικας για χειρισμό μη έγκυρης εισόδου
```

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

Εμφάνιση χαρακτήρα στην οθόνη μέσω κλήσης BIOS

```
MOV AH, 0Eh      ; Υπορουτίνα για εμφάνιση χαρακτήρα
MOV AL, 'A'      ; Χαρακτήρας που θα εμφανιστεί
INT 10h          ; Κλήση της διακοπής του BIOS
```

Εμφάνιση χαρακτήρα στην οθόνη με άμεση εγγραφή στη μνήμη video

```
MOV AX, 0xB800 ; βάση διεύθυνσης μνήμης βίντεο
MOV DS, AX      ; φόρτωση στο segment register
MOV DI, 0       ; Offset για την αρχή της οθόνης
MOV AL, 'A'     ; Χαρακτήρας που θα εμφανιστεί
MOV [DI], AX    ; Εγγραφή στη μνήμη βίντεο
```

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

Ο κατάλογος (directory) είναι μια ειδική μορφή δεδομένων που χρησιμοποιείται για την οργάνωση των αρχείων ενός αποθηκευτικού μέσου. Σε ένα αποθηκευτικό μέσο υπάρχει ένας ριζικός κατάλογος (root directory) που περιέχει αρχεία ή άλλους υποκαταλόγους (sub directories). Με τη σειρά τους οι υποκατάλογοι μπορεί να περιέχουν άλλους υποκαταλόγους ή αρχεία δημιουργώντας έτσι μια ιεραρχική δενδρική δομή οργάνωσης των αρχείων κάνοντας εύκολη την αναζήτηση τους. Οι υποστηριζόμενες λειτουργίες ενός καταλόγου είναι: Δημιουργία (md), Διαγραφή (rd), Μετονομασία (ren), Αντιγραφή (copy), Μετακίνηση (move), Εμφάνιση περιεχομένων (dir).

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

Η Διαδοχική οργάνωση αρχείων είναι ένας τρόπος αποθήκευσης δεδομένων όπου οι εγγραφές αποθηκεύονται στο αποθηκευτικό μέσο ή τη μνήμη σε σειρά ή μία μετά την άλλη. Για να προσπελάσουμε μια εγγραφή θα πρέπει να διαβάσουμε όλες τις προηγούμενες. Χρησιμοποιείται σε μέσα που δεν έχουν τη δυνατότητα άμεσης προσπέλασης, όπως μαγνητικές ταινίες. Η Άμεση οργάνωση επιτρέπει την τυχαία προσπέλαση οποιαδήποτε εγγραφής. Μπορούμε να γράψουμε, διαβάσουμε ή τροποποιήσουμε μια εγγραφή απευθείας. Χρησιμοποιείται σε αποθηκευτικά μέσα που επιτρέπουν τέτοια οργάνωση όπως σκληροί δίσκοι.

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

Στο αρχείο system.ini αποθηκεύονται πληροφορίες όπως θέση προγραμμάτων οδήγησης, ρυθμίσεις συστήματος κλπ. Φορτώνονται με την εκκίνηση του λειτουργικού συστήματος. Χρησιμοποιήθηκε στις παλαιότερες εκδόσεις των windows (1.0 έως Windows Me) αλλά πλέον αντικαταστάθηκε από τη registry. Στα windows 11 υπάρχει ακόμη για λόγους συμβατότητας με 16-bit εφαρμογές.

69. Ποια είναι τα δομικά μέρη του UNIX;

Ο πυρήνας (kernel): Είναι η καρδιά του Unix. Διαχειρίζεται τους πόρους του υπολογιστή και λειτουργεί σαν ενδιάμεσο στρώμα ανάμεσα στις εφαρμογές και το υλικό (CPU, RAM, αποθηκευτικά μέσα κλπ).

Το κέλυφος (shell): Το κέλυφος είναι η διεπαφή του χρήστη με τον πυρήνα. Μεταφράζει τις εντολές του χρήστη σε λειτουργίες του πυρήνα.

Βοηθητικά Προγράμματα (utilities): Είναι εφαρμογές που εκτελούν διάφορες εργασίες όπως διαχείριση αρχείων, επεξεργασία κειμένου, διαχείριση εργασιών κλπ.

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

Το FAT (File Allocation Table) χρονολογείται από την εποχή του MS-DOS και είναι ένα απλό σύστημα διαχείρισης αρχείων. Για λόγους συμβατότητας το υποστηρίζουν όλες οι εκδόσεις των Windows έως σήμερα και έχει βελτιωθεί με την ονομασία FAT32. Αυτό το σύστημα αρχείων υποστηρίζεται από πολλές άλλες συσκευές όπως βιντεοκάμερες και άλλες φορητές συσκευές.

Χαρακτηριστικά:

- Σε περίπτωση που καταστραφεί το ευρετήριο (File Allocation Table) τα δεδομένα δεν μπορούν να ανακτηθούν.
- Δεν υποστηρίζει δικαιώματα πρόσβασης σε επίπεδο αρχείου.
- Το μέγιστο μέγεθος ενός αρχείου είναι 4GB (αυτός ο περιορισμός δεν ισχύει για το exFAT).
- Είναι συμβατό με πολλές συσκευές διαφόρων κατασκευαστών.

Το NTFS (New Technology File System) παρουσιάστηκε με την εμφάνιση των Windows NT. Είναι ένα πολύ πιο σύγχρονο σύστημα διαχείρισης αρχείων και χρησιμοποιείται έως σήμερα σαν το βασικό σύστημα διαχείρισης των Windows.

Χαρακτηριστικά:

- Μετά από αστοχία τμημάτων του υλικού τα δεδομένα μπορούν ανακτηθούν.
- Υποστηρίζει δικαιώματα πρόσβασης σε επίπεδο αρχείου και κρυπτογράφηση.
- Μπορεί να διαχειριστεί πολύ μεγάλα αρχεία.
- Μικρότερος κατακερματισμός αρχείων.
- Πέραν των Windows πολύ λίγα λειτουργικά το αναγνωρίζουν σωστά

71. Τί είναι ανάκτηση πληροφοριών και πόσα είδη υπάρχουν;

Ανάκτηση πληροφοριών είναι ο κλάδος της πληροφορικής που ασχολείται με την εξαγωγή σχετικών πληροφοριών από μεγάλες συλλογές δεδομένων όπως κείμενα, πολυμέσα, βάσεις δεδομένων ή τον παγκόσμιο ιστό. Ο στόχος είναι να παρέχει τα πιο χρήσιμα και σχετικά αποτελέσματα στο χρήστη μετά από κάποιο ερώτημα.

Η ανάκτηση πληροφοριών χρησιμοποιεί διάφορες τεχνικές ανάλογα με το είδος των δεδομένων και τις απαιτήσεις του συστήματος. Έτσι μπορεί να χρησιμοποιεί τεχνικές μηχανικής μάθησης, γλωσσολογίας, πιθανολογίας, στατιστικής, βάσεων δεδομένων κλπ για την αποτελεσματική εξαγωγή και παρουσίαση των πληροφοριών.

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

Ένα κατακερματισμένο αρχείο χρησιμοποιεί έναν αλγόριθμο κατακερματισμού (hashing) για την αποθήκευση και ανάκτηση εγγραφών.

Κάθε εγγραφή έχει ένα μοναδικό κλειδί. Ο αλγόριθμος κατακερματισμού υπολογίζει με βάση αυτό το κλειδί τη θέση που θα αποθηκευτεί η εγγραφή. Η πρόσβαση στις εγγραφές είναι πολύ γρήγορη γιατί γίνεται απευθείας.

Ένα δεικτοδοτούμενο αρχείο χρησιμοποιεί ένα πίνακα δεικτών για την αποθήκευση των εγγραφών. Κάθε εγγραφή περιέχει ένα μοναδικό κλειδί. Ένας δείκτης αποτελείται από ένα ζευγάρι κλειδιού και

διεύθυνσης. Η πρόσβαση σε μία εγγραφή γίνεται διαβάζοντας από τον πίνακα δεικτών σε ποιά θέση είναι αποθηκευμένη με βάση το κλειδί της.

- 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 ποιάς/ποιών εντολής/ών θα το κάνετε;**

```
tar -xvf Backup.tar mad.ben
```

Όπου:

- x = Εξαγωγή αρχείου
- v = Εμφάνιση διαδικασίας στο τερματικό
- f = Ορισμός του αντίγραφου ασφαλείας (Backup.tar)
- mad.ben = Το αρχείο που θα εξαχθεί.

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

Μπορούμε να εισάγουμε μια αλυσίδα εντολών με τη χρήση του συμβόλου ; Οι εντολές θα εκτελεστούν η μία μετά την άλλη.

```
echo "Hello"; echo " world"; echo " from Linux"
```

Με το σύμβολο && μπορούμε να εισάγουμε εντολές υπό συνθήκη. Η εντολή μετά το && εκτελείται αν η προηγούμενη ήταν αληθής.

```
echo "Starting process" && ls && echo "Done"
```

Αν έχουμε μια δέσμη εντολών σε κάποιο αρχείο κειμένου μπορούμε να την εκτελέσουμε με την εντολή bash

```
bash commands.txt
```

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

Στο full backup αντιγράφονται όλα τα αρχεία. Στο partial (differential) backup αντιγράφονται μόνο τα αρχεία που προστέθηκαν ή τροποποιήθηκαν μετά το τελευταίο full backup.

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

Χρειαζόμαστε 2 partitions. Ένα primary partition και ένα swap partition.

Το primary partition κρατάει όλα τα αρχεία του λειτουργικού.

Το swap partition που είναι επέκταση της RAM στο δίσκο.

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

Το spooling είναι η προσωρινή αποθήκευση δεδομένων στη μνήμη ή το σκληρό δίσκο πριν

αποσταλούν σε μία συνήθως αργή συσκευή όπως ο εκτυπωτής. Έτσι ο επεξεργαστής ελευθερώνεται

για να εκτελέσει άλλες διεργασίες. Ο χρήστης δεν χρειάζεται να περιμένει να τελειώσει μια εκτύπωση για να στείλει μια άλλη.

78. Αναφέρετε τί 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;

Υπάρχουν 3 κατηγορίες χρηστών:

Ιδιοκτήτης (Owner): Είναι αυτός που έχει δημιουργήσει το φάκελο ή αρχείο και έχει πλήρη δικαιώματα σ' αυτό.

Ομάδα (Group): Είναι οι χρήστες που ανήκουν στην ομάδα του ιδιοκτήτη.

Άλλοι (Others): Όλοι οι υπόλοιποι.

Οι φάκελοι και τα αρχεία έχουν 3 επίπεδα προστασίας:

Ανάγνωση (Read): Ο χρήστης έχουν μόνο δικαίωμα να εμφανίσουν τα περιεχόμενα τα αρχείου ή φακέλου. Επίσης μπορούν να τον αντιγράψουν.

Εγγραφή (Write): Ο χρήστης έχει δικαίωμα δημιουργίας αρχείων μέσα στο φάκελο. Επίσης μπορεί να τροποποιήσει, μετονομάσει, μετακινήσει ή να διαγράψει αρχεία.

Εκτέλεση (Execute): Ο χρήστης μπορεί να τρέξει το πρόγραμμα αν είναι εκτελέσιμο.

Τα επίπεδα δηλώνονται με ένα αντίστοιχο αγγλικό γράμμα:

- r = Read
- w = Write
- x = Execute

Αν αντί γράμμα υπάρχει – σημαίνει ότι ο χρήστης δεν έχει το αντίστοιχο δικαίωμα.

Συνολικά βλέπουμε σε κάθε φάκελο ή αρχείο 10 γράμματα που υποδηλώνουν τα επίπεδα προστασίας. Το πρώτο γράμμα είναι d αν αφορά φάκελο ή – αν αφορά αρχείο. Οι υπόλοιπες 3 τριάδες αφορούν το επίπεδο προστασίας για κάθε κατηγορία χρηστών.

Παράδειγμα: drwxr-xr—

Σημαίνει ότι αφορά φάκελο, ο ιδιοκτήτης έχει πλήρη δικαιώματα, η ομάδα έχει δικαιώματα ανάγνωσης και εκτέλεσης ενώ οι λοιποί χρήστης έχουν μόνο δικαίωμα ανάγνωσης.

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

d = directory (φάκελος)

r = read (δικαίωμα εμφάνισης περιεχομένων και αντιγραφής)

w = write (δικαίωμα δημιουργίας αρχείων, τροποποίησης, μετακίνησης, διαγραφής)

x = execute (δικαίωμα εκτέλεσης αν πρόκειται για εκτελέσιμο αρχείο)

- = αφαιρεί το αντίστοιχο δικαίωμα

Σε κάθε αρχείο ή φάκελο καθορίζεται το δικαίωμα πρόσβασης για κάθε μία από τις 3 κατηγορίες χρηστών στο Linux (Ιδιοκτήτης, Ομάδα, Άλλοι). Υπάρχουν δηλαδή 10 γράμματα. Το 1^ο συμβολίζει αν πρόκειται για φάκελο (d) ή αρχείο (-). Τα άλλα 9 είναι 3 τριάδες με τα δικαιώματα της κάθε κατηγορίας χρηστών (rwx)

drwxrwxrwx = Σημαίνει ότι πρόκειται για φάκελο και έχουν πλήρη δικαιώματα rwx όλες οι κατηγορίες χρηστών

drwxr-xr-- = Σημαίνει ότι ο Ιδιοκτήτης έχει πλήρη δικαιώματα (read, write, execute), η Ομάδα έχει δικαίωμα read και execute ενώ οι Άλλοι μόνο δικαίωμα read.

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

Το λειτουργικό σύστημα είναι λογισμικό που παρεμβάλλεται ανάμεσα στις εφαρμογές που χρησιμοποιεί ο χρήστης και το υλικό του υπολογιστή (CPU, RAM, μονάδες εισόδου/εξόδου, μονάδες αποθήκευσης).

Μεταφράζει τις εντολές των προγραμμάτων σε εντολές που μπορεί να κατανοήσει το υλικό.

Διαχειρίζεται τους πόρους του υπολογιστή, τα δικαιώματα πρόσβασης των χρηστών, την επικοινωνία με άλλα συστήματα, την εύρυθμη λειτουργία των εφαρμογών κλπ.

Τα πρώτα χρόνια εμφάνισης των υπολογιστών δεν υπήρχαν λειτουργικά συστήματα. Ο χειρισμός ήταν μια επίπονη και χρονοβόρα διαδικασία καθώς έπρεπε να ανοιγοκλείνουν διακόπτες για να δώσουν εντολές στον υπολογιστή ώστε να εκτελεστεί ένα πρόγραμμα.

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

Ο πυρήνας (kernel) είναι η καρδιά ενός λειτουργικού συστήματος. Βρίσκεται στο χαμηλότερο επίπεδο ανάμεσα στο υλικό του υπολογιστή και τις υπόλοιπες εφαρμογές. Αναλαμβάνει να μεταφράσει τις εντολές των εφαρμογών σε εντολές κατάλληλες προς το υλικό. Διαχειρίζεται τους πόρους του συστήματος και ελέγχει την καλή του λειτουργία. Επικοινωνεί με τις μονάδες εισόδου/εξόδου και συντονίζει τις διεργασίες που εκτελούνται ώστε η απόκριση του συστήματος να είναι όσο το δυνατό ταχύτερη και η εκμετάλλευση πόρων βέλτιστη.

Το Σύστημα Αρχείων

Η Διεπαφή Χρήστη (User Interface): Μπορεί να είναι ένα γραφικό περιβάλλον ή ένας διερμηνευτής εντολών. Επιτρέπει στο χρήστη να χειρίζεται αρχεία, να εκτελεί προγράμματα και να προβαίνει σε ρυθμίσεις του συστήματος.

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

FTP (File Transfer Protocol)

Είναι ένα πρωτόκολλο αποστολής και λήψης αρχείων. Είναι ένα ευρέως χρησιμοποιούμενο πρωτόκολλο για τη μεταφορά αρχείων στο διαδίκτυο. Τα αρχεία είναι αποθηκευμένα σε κάποιον διακομιστή. Οι υπολογιστές που έχουν δικαίωμα πρόσβασης μπορούν ανεβάσουν, κατεβάσουν, μετονομάσουν ή διαγράψουν αρχεία με FTP. Για παράδειγμα με αυτό το πρωτόκολλο μπορούμε να ανεβάσουμε μια ιστοσελίδα που έχουμε φτιάξει σε κάποιον server.

HTTP (Hyper Text Transfer Protocol)

Είναι ένα πρωτόκολλο μετάδοσης και εμφάνισης πληροφοριών όπως κείμενο, εικόνες και πολυμέσα. Αυτό είναι το πρωτόκολλο που χρησιμοποιούν οι browsers για να επικοινωνήσουν με ένα διακομιστή και να εμφανίσουν μια ιστοσελίδα στο χρήστη.

- 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;**

Στην απλή (simplex) επικοινωνία έχουμε μονοσήμαντη μετάδοση από ένα πομπό σε ένα δέκτη.

Στη half duplex επικοινωνία και τα δύο σημεία μετάδοσης είναι πομποί αλλά και δέκτες. Η επικοινωνία είναι αμφίδρομη δεν μπορεί όμως να είναι ταυτόχρονη. Όταν το ένα σημείο Α είναι πομπός το άλλο σημείο Β θα είναι δέκτης. Όταν ολοκληρωθεί η μετάδοση το σημείο Β μπορεί να γίνει πομπός και το Α δέκτης.

Στη full duplex έχουμε ταυτόχρονη αμφίδρομη επικοινωνία. Δύο σημεία μπορούν να εκπέμπουν αλλά και να δέχονται ταυτόχρονα δεδομένα.

- 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.**

Στις υπηρεσίες με σύνδεση απαιτείται η σύνδεση των 2 μερών πριν ξεκινήσει η μετάδοση πληροφοριών. Δημιουργείται κατά κάποιο τρόπο ένας σωλήνας που ενώνει τα 2 μέρη. Τα πακέτα δεδομένων φεύγουν από το ένα σημείο το ένα μετά το άλλο και καταλήγουν με την ίδια σειρά στο άλλο σημείο. Παράδειγμα αυτής της επικοινωνίας είναι η τηλεφωνική σύνδεση.

Στις υπηρεσίες χωρίς σύνδεση τα πακέτα φεύγουν από τον αποστολέα προς κάθε κατεύθυνση, αναγράφουν όμως επάνω τους τη διεύθυνση του παραλήπτη. Τα πακέτα δεν φθάνουν με την ίδια σειρά που αποστάλθηκαν και θα πρέπει ο παραλήπτης να τα βάλει σε σωστή σειρά. Παράδειγμα το ταχυδρομείο.

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

Η μεταγωγή κυκλώματος απαιτεί τη φυσική σύνδεση μεταξύ δύο τερματικών σταθμών για τη μετάδοση πληροφορίας. Αυτό σημαίνει ότι κανένας άλλος σταθμός δεν μπορεί να χρησιμοποιήσει αυτό το φυσικό μονοπάτι, παρόλο που οι δύο σταθμοί μπορεί να μη το χρησιμοποιούν εκείνη τη στιγμή.

Η μεταγωγή πακέτου λύνει αυτό το πρόβλημα. Η πληροφορία χωρίζεται σε τμήματα που ονομάζονται πακέτα. Τα δίκτυα αυτού του είδους δεν αποκαθιστούν κάποια φυσική σύνδεση μεταξύ των κόμβων αλλά διεκπεραιώνουν τη δρομολόγηση των πακέτων. Έτσι μπορούν να χρησιμοποιούν τις ίδιες φυσικές συνδέσεις πολλοί τερματικοί σταθμοί.

Οι μέθοδοι προώθησης είναι:

Datagram: Σε κάθε πακέτο επισυνάπτεται η διεύθυνση του παραλήπτη. Ο ένας σταθμός αποστέλλει τα πακέτα προς όποια σύνδεση είναι διαθέσιμη και τα πακέτα δρομολογούνται προς τον παραλήπτη μέσω διάφορων ελεύθερων κόμβων.

Νοητό κύκλωμα: Δημιουργείται ένα εικονικό μονοπάτι μεταξύ των 2 σταθμών. Τα πακέτα που δρομολογούνται σ' αυτό το μονοπάτι απευθύνονται όλα στον ίδιο παραλήπτη.

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

Το Frame Relay είναι μια τεχνική επικοινωνίας που βασίζεται στη μεταγωγή πακέτων. Τα δεδομένα σπάνε σε πακέτα και μεταφέρονται μέσω ενός νοητού κυκλώματος. Η μετάδοση είναι πιο γρήγορη από άλλες τεχνικές μεταγωγής πακέτων γιατί έχουν αφαιρεθεί αρκετές λειτουργίες ελέγχου που δεν είναι απαραίτητες σε ένα ψηφιακό ασφαλές περιβάλλον. Επίσης είναι πιο οικονομικό από μια μισθωμένη γραμμή γιατί δεν απαιτείται η φυσική σύνδεση μεταξύ των τερματικών σταθμών. Ήταν μια πολύ διαδεδομένη τεχνική επικοινωνίας, σήμερα όμως τείνει να αντικατασταθεί από άλλες μορφές (πχ VPN).

87. Τι ονομάζουμε επικοινωνία δεδομένων;

Επικοινωνία δεδομένων είναι η ανταλλαγή πληροφοριών ή δεδομένων μεταξύ δύο ή περισσότερων συσκευών ή υπολογιστών μέσω ενός καναλιού επικοινωνίας. Τέτοια κανάλια επικοινωνίας μπορεί να είναι καλώδια χαλκού, ραδιοκύματα, οπτικές ίνες. Είναι θεμελιώδης για τη λειτουργία δικτύων υπολογιστών, το διαδίκτυο, συστήματα τηλεπικοινωνιών και πληροφορικής.

88. Ποια είναι τα κύρια στοιχεία μετάδοσης των δεδομένων;

- Πηγή: Δημιουργία πληροφορίας
- Κωδικοποιητής: Μετατροπή πληροφορίας σε μορφή κατάλληλη για το σύστημα μετάδοσης.
- Πομπός: Εκπομπή της διαμορφωμένης πληροφορίας
- Μέσο μετάδοσης: Το κανάλι επικοινωνίας
- Δέκτης: Λήψη της διαμορφωμένης πληροφορίας
- Αποκωδικοποιητής: Μετατροπή της διαμορφωμένης πληροφορίας σε μορφή κατάλληλη για τον παραλήπτη.
- Παραλήπτης: Αποδέκτης της πληροφορίας.

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

- UTP (Unshielded Twisted Pair): Τα συνεστραμμένα ζεύγη δεν έχουν κάποια θωράκιση.
- FTP (Foiled Twisted Pair): Η δέσμη των καλωδίων έχει θωράκιση φύλλων αλουμινίου μέσα από το κάλυμμα.
- STP (Shielded Twisted Pair): Τα ζεύγη έχουν θωράκιση μεταξύ τους.
- SFTP (Shielded Foiled Twisted Pair): Υπάρχει θωράκιση τόσο μεταξύ των ζευγών όσο και γύρω από τη δέσμη καλωδίων.

Τα UTP χρησιμοποιούνται σε εσωτερικούς χώρους όπου δεν υπάρχει ανάγκη θωράκισης ενώ τα FTP, STP, SFTP σε χώρους που χρειαζόμαστε προστασία από ηλεκτρομαγνητικές παρεμβολές. Τα UTP είναι πιο οικονομικά και ευκολότερα στη διαχείριση γιατί είναι πιο εύκαμπτα.

Επίσης τα καλώδια UTP, FTP κλπ κατατάσσονται σε κατηγορίες CAT ανάλογα με την απόδοσή τους

- CAT5: Ταχύτητα έως 100Mbps
- CAT5e: Ταχύτητα έως 1000Mbps
- CAT6: Ταχύτητα έως 10Gbps

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

- Ηλεκτρομαγνητικές παρεμβολές από ηλεκτρικά καλώδια τροφοδοσίας συσκευών
- Παρεμβολές ραδιοσυχνοτήτων από κεραίες εκπομπής.
- Ενδογενής θόρυβος από τις συσκευές μετάδοσης.
- Θόρυβος από το φυσικό περιβάλλον (θερμοκρασία, υγρασία).

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

Το αναλογικό σήμα είναι μια συνεχής κυματομορφή που μπορεί να πάρει άπειρες τιμές ανά μονάδα χρόνου. Κυριότερη παράμετρος ενός αναλογικού σήματος είναι η συχνότητα Hz. Είναι ευαίσθητο σε παρεμβολές και η ποιότητα του μειώνεται με την απόσταση. Είναι δύσκολο να αποθηκευτεί και να μεταδοθεί. Χαρακτηριστικό παράδειγμα είναι η φωνή μας που είναι απλά μεταβολές στην πίεση του αέρα.

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

Τα ψηφιακά σήματα παίρνουν διακριτές (συγκεκριμένες) τιμές ανά μονάδα χρόνου. Τα ψηφιακά δυαδικά σήματα μπορούν να πάρουν μόνο 2 τιμές. Η αποθήκευση και η μετάδοση τους είναι εύκολη και χρησιμοποιούνται κατά κόρον σε ηλεκτρονικά και υπολογιστικά συστήματα.

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

Μετατροπή από ψηφιακό σήμα σε αναλογικό απαιτείται όταν το σήμα είναι ψηφιακό αλλά πρόκειται να μεταδοθεί από αναλογικό μέσο. Παράδειγμα τα ηχεία. Ένα ψηφιακό αρχείο mp3 για να μεταδοθεί από τα ηχεία θα πρέπει να μετατραπεί σε αναλογικό σήμα για να παραχθεί ήχος. Το αντίθετο ισχύει για παράδειγμα στο μικρόφωνο. Λαμβάνει το αναλογικό σήμα της φωνής μας και το σήμα πρέπει να μετατραπεί σε ψηφιακό για να αποθηκευτεί σε αρχείο mp3.

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

Στη σύγχρονη μετάδοση οι πληροφορίες ομαδοποιούνται σε πακέτα και αποστέλλονται συνεχώς. Απαιτείται σήμα χρονισμού (clock) ανάμεσα στον αποστολέα και τον παραλήπτη.

Στην ασύγχρονη μετάδοση αποστέλλεται ένας χαρακτήρας (byte) κάθε φορά. Στην αρχή και το τέλος κάθε χαρακτήρα προστίθεται ένα bit για συγχρονισμό.

Η σύγχρονη μετάδοση είναι ταχύτερη αλλά και πιο ακριβή στην υλοποίηση λόγω του μηχανισμού σήματος χρονισμού. Χρησιμοποιείται σε δίκτυα υψηλών ταχυτήτων όπως Ethernet, τηλεπικοινωνιακά δίκτυα και οπτικές ίνες.

Η ασύγχρονη μετάδοση χρησιμοποιείται για κοντινές αποστάσεις και όπου δεν υπάρχουν υψηλές απαιτήσεις ταχύτητας, όπως πληκτρολόγια, ποντίκια κλπ.

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

Στη μεταγωγή μηνύματος ο τερματικός σταθμός αποστέλλει το μήνυμα προσαρτώντας τη διεύθυνση του παραλήπτη. Τα δίκτυα αυτού του είδους δεν αποκαθιστούν κάποια φυσική σύνδεση μεταξύ

αποστολέα και παραλήπτη. Φροντίζουν μόνο για τη διεκπεραίωση του μηνύματος. Έτσι μπορούν και άλλοι τερματικοί σταθμοί να χρησιμοποιούν τις ίδιες φυσικές συνδέσεις.

Η μεταγωγή πακέτου μοιάζει με τη μεταγωγή μηνύματος αλλά το μήνυμα σπάει σε πακέτα. Τα πακέτα μπορεί να μη φτάσουν στον παραλήπτη με την ίδια σειρά που στάλθηκαν και ο παραλήπτης θα πρέπει να τα τοποθετήσει ξανά στη σωστή σειρά για να συνθέσει το μήνυμα.

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

Ο πομπός αποστέλλει ένα διαμορφωμένο σήμα προς τον δέκτη μέσω ενός μέσου μετάδοσης. Στην ασύγχρονη μετάδοση αποστέλλεται διαδοχικά ένα χαρακτήρας (byte) κάθε φορά. Σε κάθε χαρακτήρα προστίθεται 1 bit στην αρχή και 1 στο τέλος για συγχρονισμό. Οι χαρακτήρες δεν μεταδίδονται σε τακτά χρονικά διαστήματα. Η ασύγχρονη μετάδοση χρησιμοποιείται για κοντινές αποστάσεις και όπου δεν υπάρχει ανάγκη υψηλών ταχυτήτων όπως πληκτρολόγιο, ποντίκι κλπ. Στη σύγχρονη μετάδοση τα δεδομένα αποστέλλονται σε πακέτα. Μεταξύ του πομπού και του δέκτη παράγεται ένα σήμα χρονισμού (clock). Έτσι δεν χρειάζονται bit συγχρονισμού και τα πακέτα αποστέλλονται συνεχώς σε τακτά διαστήματα. Είναι ταχύτερη από την ασύγχρονη αλλά και πιο ακριβή στην υλοποίηση λόγω του μηχανισμού σημάτων χρονισμού. Χρησιμοποιείται σε δίκτυα υψηλών ταχυτήτων όπως Ethernet, οπτικές ίνες κλπ. Ο ρυθμός εκπομπής bit ανά μονάδα χρόνου ονομάζεται ρυθμός μετάδοσης και εκφράζεται σε bits per second (bps, Mbps, Gbps).

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 ;

`Επικοινωνία`

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

Θα εμφανίσει μια σελίδα στον browser με τίτλο Άσκηση και ένα παράθυρο με το μήνυμα "Αυτό είναι το μήνυμα 1".

Ωστόσο ο κώδικας δεν είναι ολοκληρωμένος και μπορεί το παράθυρο να μην εμφανιστεί γιατί λείπουν τα closing tags `</SCRIPT></HEAD></HTML>`

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;

Παράδειγμα εισαγωγής εικόνας:

``

Στο παραπάνω παράδειγμα θα εμφανιστεί η εικόνα με όνομα αρχείου image.jpg και θα έχει ύψος (height) 100 pixels και πλάτος (width) 50 pixels.

Η εικόνα δεν είναι απαραίτητο να βρίσκεται στον ίδιο υπολογιστή. Θα πρέπει όμως να καθορίσουμε την πλήρη διαδρομή του αρχείου ή URL στο src.

Η παράμετρος align καθορίζει πως θα στοιχηθεί η εικόνα σε σχέση με το περιβάλλον κείμενο. Στο παράδειγμα θα εμφανιστεί αριστερά. Πλέον αποφεύγεται από τη σύγχρονη HTML και χρησιμοποιούμε CSS.

Η παράμετρος alt περιέχει κείμενο που θα εμφανιστεί σε περίπτωση που δεν είναι δυνατή η εμφάνιση της εικόνας. Η παράμετρος CAPTION δεν συνδέεται απευθείας με την εικόνα. Αν θέλαμε να βάλουμε κάποια λεζάντα που να περιγράφει την εικόνα θα γράφαμε:

```
<figure>
  
  <figcaption>Τοπίο με βουνά</figcaption>
</figure>
```

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

Μπορούμε να εισάγουμε κείμενο απευθείας στο <body> μιας σελίδας χρησιμοποιώντας ετικέτες όπως <p> για εισαγωγή παραγράφου ή <h1>...<h6> για διάφορες μορφές επικεφαλίδων.

Μπορούμε επίσης να εισάγουμε κείμενο με χρήση JavaScript κάτι που είναι πολύ συνηθισμένο σε δυναμικές σελίδες.

Με χρήση φορμών μπορεί επίσης να εισάγει ένας χρήστης κείμενο σε μια σελίδα.

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

- CAPTION: Εμφανίζει μια λεζάντα για τον πίνακα
- TH: Ορίζει το τμήμα επικεφαλίδας του πίνακα. Το κείμενο που βρίσκεται ανάμεσα στις ετικέτες <TH></TH> εμφανίζεται με έντονη γραφή.
- COLSPAN: Ακολουθείται από έναν αριθμό που καθορίζει πόσες στήλες θα καταλάβει ένα κελί.
- ROWSPAN: Ακολουθείται από έναν αριθμό που καθορίζει πόσες σειρές θα καταλάβει ένα κελί.
- ALIGN: Στοιχίζει τα περιεχόμενα των κελιών οριζόντια.
- VALIGN: Στοιχίζει τα περιεχόμενα των κελιών κάθετα.

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

Οι σημάνσεις επικεφαλίδας εμφανίζουν το κείμενο με μεγαλύτερο ύψος γραμμάτων και έντονη γραφή. Έτσι ξεχωρίζει από τις υπόλοιπες παραγράφους και κάνει το κείμενο πιο ευανάγνωστο χωρίζοντάς το σε ενότητες που διαβάζονται πιο εύκολα.

Οι μηχανές αναζήτησης δίνουν μεγάλη βαρύτητα στις επικεφαλίδες, θεωρώντας ότι περιγράφουν το θέμα που διαπραγματεύεται η σελίδα. Έτσι θα εμφανίσουν πιο ψηλά στην κατάταξη μια ιστοσελίδα που περιέχει σε επικεφαλίδα κείμενο σχετικό με αυτό που αναζητάει ο χρήστης.

Το heading 1 έχει μεγαλύτερο μέγεθος από το 6.

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 σελίδες περιέχουν σταθερό κείμενο και γραφικά που δεν αλλάζει συχνά. Η ίδια σελίδα θα εμφανιστεί σε όλους τους χρήστες. Παράδειγμα μια σελίδα εταιρικής παρουσίασης.

Οι δυναμικές σελίδες παρέχουν δεδομένα που συνήθως αντλούνται από κάποια βάση δεδομένων. Έτσι το περιεχόμενό τους αλλάζει ανάλογα με τις προτιμήσεις του χρήστη. Παράδειγμα ένα e-shop.

Για κατασκευή στατικών σελίδων: HTML, CSS, JavaScript.

Για κατασκευή δυναμικών σελίδων: PHP, SQL, JavaScript, .NET.

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

`Send mail`

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.

i	Μόνο έντονα	Παράδειγμα
ii	Έντονα και υπογραμμισμένη	<u>Παράδειγμα</u>
iii	Υπογραμμισμένη	<u>Παράδειγμα</u>
iv	Έντονα, πλάγια και υπογραμμισμένη	<i><u>Παράδειγμα</u></i>

106. Τι κάνει η εντολή <BODY BACKGROUND = "bgimage.gif">; Πώς μπορούμε να αλλάξουμε το χρώμα του 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% μπλε.

Η εντολή θα εμφανίσει την ιστοσελίδα με την εικόνα bgimage.gif σαν φόντο. Θα πρέπει να αποφεύγεται στη σύγχρονη HTML και να χρησιμοποιείται CSS.

Για να αλλάξουμε το φόντο μιας HTML σελίδας χρησιμοποιούμε CSS αλλάζοντας την ιδιότητα background-color:

Πράσινο φόντο: `<body style="background-color: green;">`

50% Πράσινο – 50% Μπλε φόντο: `<body style="background-color: rgb(0, 128, 128);">`

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

Η div ομαδοποιεί τα στοιχεία που περιέχει (μπορεί να είναι κείμενο, γραφικά, φόρμες κλπ). Έτσι μπορούμε να εφαρμόσουμε ένα ενιαίο στυλ εμφάνισης για όλο το περιεχόμενό που περιέχεται μέσα στα tags `<div> ... </div>`.

align:

Στοιχίζει το περιεχόμενο της div.

class:

Ορίζει ένα αναγνωριστικό που μπορεί να χρησιμοποιηθεί από τη CSS για μορφοποίηση της div. Μπορεί και άλλες ετικέτες να έχουν το ίδιο όνομα κλάσης. Η μορφοποίηση θα εφαρμοστεί σε όλες.

Παράδειγμα:

```
<div class="my-div">
```

Στη συνέχεια αναφερόμαστε στη CSS για αυτή την κλάση με μια τελεία μπροστά από το όνομά της:
.my-div {color: red;} /* Το κείμενο όσων ανήκουν σ' αυτή την κλάση θα εμφανίζεται κόκκινο */

id:

Ορίζει ένα μοναδικό αναγνωριστικό που μπορεί να χρησιμοποιηθεί από τη JavaScript ή CSS για μορφοποίηση της div.

Παράδειγμα: <div id="my-id">

Στη CSS αναφερόμαστε στο id βάζοντας # μπροστά από το όνομα. Το id είναι μοναδικό για κάθε ετικέτα:

```
#my-id {color: red;} /* Το κείμενο της div θα είναι κόκκινο */
```

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

Ιανουάριος	1		
Φεβρουάριος		2	
Μάρτιος			3

```
<html>
<head>
  <style>
    table {
      border: 1px solid black;
      border-collapse: collapse;
      width: 400px;
      height: 150px;
    }
    td {
      text-align: center;
      width: 100px;
      border: 1px solid black;
    }
    .month {
      text-align: left;
      width: 100px;
      padding-left: 20px;
    }
  </style>
</head>
<body>
```

```
<table>
  <tr>
    <td class="month">Ιανουάριος</td><td>1</td><td></td><td></td>
  </tr>
  <tr>
    <td class="month">Φεβρουάριος</td><td></td><td>2</td><td></td>
  </tr>
  <tr>
    <td class="month">Μάρτιος</td><td></td><td></td><td>3</td>
  </tr>
</table>
</body>
</html>
```

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

Με τις επώνυμες άγκυρες ορίζουμε σημεία μέσα στην ιστοσελίδα που μπορούμε να μεταβούμε από άλλα σημεία της ή από άλλες σελίδες με χρήση υπερσυνδέσμων.

Παράδειγμα:

```
<a href="#unit1">Go to Unit 1</a>
<p> Long text follows </p>
<h1 id="#unit1">This is unit 1</h1>
```

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

Με τη χρήση CSS και να ορίσουμε την εικόνα ως background-image.

Παράδειγμα:

```
<html>
<head>
  <style>
    .container {
      background-image: url("pic1.jpg");
      background-size: cover;
      background-position: center;
      height: 300px;
      width: 300px;
    }
  </style>
</head>
<body>
  <div class="container"> Text over image </div>
</body>
</html>
```

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

Το CSS είναι μια γλώσσα που μας επιτρέπει να μορφοποιήσουμε σελίδες HTML.

- Μπορεί να ενσωματωθεί inline σε μια ετικέτα με την παράμετρο style.
<p style="color: red;">Red text</p>
- Μπορεί να είναι στο ίδιο αρχείο με τον κώδικα HTML.
<html>

```
<head>
  <style>
    .paragraph {
      color: red;
    }
  </style>
</head>
<body>
  <p class="paragraph">Red text</p>
</body>
</html>
```

- Μπορεί να είναι σε ξεχωριστό αρχείο. Σ' αυτή την περίπτωση πρέπει να ορίσουμε στο head της σελίδας τη θέση και το όνομα του αρχείου.

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

Συνιστάται να είναι η CSS σε ξεχωριστό αρχείο. Έτσι αλλάζοντας τη μορφοποίηση κάποιας ετικέτας ή κλάσης στη CSS θα αλλάξει η εμφάνιση όλων των σελίδων που τη χρησιμοποιούν προσφέροντας έτσι ενιαία εμφάνιση του ιστότοπου χωρίς να χρειάζεται να αλλάξουμε κάθε σελίδα χωριστά.

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.

Η υπερσύνδεση είναι μια αναφορά σε κάποιο άλλο σημείο της σελίδας ή μιας άλλης σελίδας του ίδιου ή άλλου ιστότοπου. Ο χρήστης κάνοντας κλικ στον υπερσύνδεσμο μεταφέρεται στην άλλη σελίδα.

_self: Ανοίγει τη σελίδα στο πλαίσιο που περιέχει την υπερσύνδεση.

_parent: Ανοίγει τη σελίδα στο γονικό πλαίσιο του υπερσυνδέσμου.

_blank: Ανοίγει τη σελίδα σε νέο παράθυρο του φυλλομετρητή.

_top: Ανοίγει τη σελίδα στο τρέχον παράθυρο.

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

Ο χάρτης εικόνας είναι μια τεχνική που μας επιτρέπει να ορίσουμε ενεργές περιοχές (hot spots) σε μια εικόνα. Κάθε hot spot λειτουργεί ως υπερσύνδεσμος ή μπορεί να εκτελέσει άλλες λειτουργίες καλώντας μια συνάρτηση JavaScript.

Τα hot spot μπορεί να είναι ορθογώνιες περιοχές, κυκλικές ή πολυγωνικές.

Για να σχεδιάσουμε ένα ακανόνιστο σχήμα θα χρησιμοποιήσουμε το πολυγωνικό σχήμα εισάγοντας τις συντεταγμένες x, y κάθε κορυφής του. Παράδειγμα:

```
<area shape="poly" coords="10, 10, 20, 25, 30, 25, 15, 15" href="mypage.html">
```


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

Με το cell padding ορίζουμε το εσωτερικό περιθώριο που θα έχει το περιεχόμενο του κελιού από τις πλευρές του.

Με το cell spacing ορίζουμε την απόσταση που θα έχουν τα κελιά μεταξύ τους.

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.

- GIF: Υποστηρίζει μόνο 256 χρώματα οπότε δεν είναι κατάλληλο για φωτογραφίες. Έχει συμπίεση χωρίς απώλειες οπότε οι λεπτομέρειες των γραμμάτων δεν χάνονται και είναι κατάλληλο για σχεδίαση λογότυπων. Υποστηρίζει απλή διαφάνεια και κινούμενες εικόνες οπότε είναι κατάλληλο και για σχεδίαση κινούμενων banner.
- JPEG: Έχει συμπίεση με απώλειες. Είναι κατάλληλο για φωτογραφίες γιατί υποστηρίζει 16 εκατ. χρώματα και τα αρχεία έχουν μικρό μέγεθος χωρίς οι εικόνες να χάνουν ιδιαίτερα σε ευκρίνεια. Δεν είναι κατάλληλο όμως για λογότυπα γιατί οι όποιες απώλειες φαίνονται πιο ευδιάκριτα. Δεν υποστηρίζει διαφάνεια.
- PNG: Έχει συμπίεση χωρίς απώλειες οπότε τα αρχεία έχουν μεγαλύτερο μέγεθος από JPG. Για το λόγο αυτό δεν συνίσταται για φωτογραφίες εκτός κι αν θέλουμε τέλεια απεικόνιση. Υποστηρίζει 16 εκατ. χρώματα και είναι ιδανικό για σχεδίαση λογότυπων, banner κλπ. Στα πλεονεκτήματά του ότι υποστηρίζει πολλά επίπεδα διαφάνειας.

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.

Η συνάρτηση swap_image() θα χρησιμοποιηθεί για την αλλαγή μιας εικόνας όταν το ποντίκι βρίσκεται πάνω σε αυτήν (rollover effect).

Το αντικείμενο που καλεί τη συνάρτηση είναι ένα στοιχείο εικόνας (). Χρησιμοποιεί τα συμβάντα onmouseover και onmouseout για να καλέσει τη συνάρτηση swap_image().

```
<html>
<head>
  <script>
    function swap_image(id, path) {
      document.getElementById(id).src = path;
    }
  </script>
</head>
<body>
  
</body>
</html>
```

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.

```
<html>
<head>
  <script>
    function check_form() {
      const form = document.getElementById("form1");
      if (form.username.value == "" || form.password.value == "") {
        alert("User Name or Password must not be blank");
      }
    }
  </script>
</head>
<body>
  <form id="form1">
    User Name: <input type="text" id="username">
    Password: <input type="password" id="password">
    <input type="button" value="Submit" onclick="check_form()">
  </form>
</body>
</html>
```

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

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

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.

```
<html>
<head>
</head>
<body>
  <script>
    let value = 1;
```

```
let table = "<table style='margin: 0 auto; border: 1px solid black;' +  
            \" border-collapse: collapse; text-align: center;'>\";  
for (let row = 0; row < 7; row++) {  
    table += "<tr>\";  
    for (let col = 0; col < 6; col++) {  
        table += "<td style='padding: 4px; border: 1px solid black;\";  
        if (value % 7 == 0) {  
            table += \" background-color: red; font-weight: bold;\";  
        }  
        table += \"'>\" + value + "</td>\";  
        value++;  
    }  
    table += "</tr>\";  
}  
table += "</table>\";  
document.body.innerHTML += table;  
</script>  
</body>  
</html>
```

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.

```
<?php  
$query = "SELECT * FROM subscribers\";  
$result = mysqli_query($conn, $query); // $conn η σύνδεση με DB  
while ($row = mysqli_fetch_assoc($result)) {  
    $body = "Your username is \" . $row["username"] .  
            \" and your password is \" . $row["password\";\";  
    mail($row["email"], "Login info", $body);  
}  
?>
```

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 κωδικό σήμανσης για την κάθε μια από τις τέσσερις περιπτώσεις.

- α) <table>
- β) <tr>
- γ) <td>
- δ) <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;

Όταν ορίζουμε το CSS σε εξωτερικό αρχείο μπορούμε να χρησιμοποιήσουμε τα στυλ σε πολλές ιστοσελίδες χωρίς να χρειάζεται να τα ξαναγράψουμε για κάθε ιστοσελίδα. Κάθε αλλαγή στο εξωτερικό CSS θα επηρεάσει όλες τις σελίδες που αναφέρονται σε αυτό κι έτσι μπορούμε εύκολα να τροποποιήσουμε την εμφάνιση των σελίδων μας καθολικά. Επίσης επειδή δεν ενσωματώνουμε τον κώδικα σε κάθε σελίδα έχουμε μικρότερο μέγεθος αρχείων και γρηγορότερη φόρτωση.

Όταν χρησιμοποιούμε inline στυλ αλλάζουμε το στυλ μόνο στο συγκεκριμένο element. Αν θέλουμε να το χρησιμοποιήσουμε σε άλλο element θα πρέπει να το ξαναγράψουμε. Έτσι μεγαλώνει πολύ το αρχείο μας και αυξάνουν οι πιθανότητες λαθών.

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: Επιδρά στα στοιχεία της σελίδας που έχουν το συγκεκριμένο HTML tag. Για παράδειγμα αν θέλουμε όλες οι επικεφαλίδες h2 να έχουν κόκκινα γράμματα θα γράψουμε:

```
h2 {color: red;}
```

Class selector: Επιδρά στα στοιχεία κλάσης. Για να ορίσουμε μια κλάση στον HTML κώδικα γράφουμε class="όνομα κλάσης". Πχ για να ορίσουμε μια κλάση με όνομα redText σε παράγραφο θα γράψουμε <p class="redText">This text is red</p>. Στο CSS αναφερόμαστε σ' αυτή την κλάση βάζοντας τελεία μπροστά από το όνομα της κλάσης. Το παρακάτω θα εμφανίσει με κόκκινα γράμματα όλα τα elements που έχουν κλάση redText:

```
.redText {colour: red;}
```

ID selector: Επιδρά στο στοιχείο με το συγκεκριμένο id. Στον HTML κώδικα για να ορίσουμε id σε μια ετικέτα div γράφουμε πχ <div id="redText">This text is red</div>. Στο CSS αναφερόμαστε στο συγκεκριμένο id βάζοντας στην αρχή #:

```
#redText {colour: red;}
```

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.

Μπορούμε να χρησιμοποιήσουμε κάποιον επεξεργαστή όπως το notepad ή το Visual Studio Code και να γράψουμε τον παρακάτω κώδικα. Μετά αποθηκεύουμε το αρχείο με κατάληξη .html.

```
<html>
<head>
  <style>
    table {
      margin: 0 auto;
      border: 1px solid black;
      border-collapse: collapse;
    }
    td {
      border: 1px solid black;
```

```
        text-align: center;
    }
    .col1 {width: 100px;}
    .col2 {width: 300px;}
</style>
</head>
<body>
    <table>
        <tr>
            <td class="col1">1,1</td><td class="col2">1,2</td>
        </tr>
        <tr>
            <td class="col1">2,1</td><td class="col2">2,2</td>
        </tr>
        <tr>
            <td class="col1">3,1</td><td class="col2">3,2</td>
        </tr>
    </table>
</body>
</html>
```

- 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;**

Image Hotspots: Μπορούμε να χρησιμοποιήσουμε κάποιο πρόγραμμα γραφικών όπως το GIMP για να βρούμε τις συντεταγμένες των hotspots. Το hotspot μπορεί να είναι τετράγωνο, κύκλος ή πολύγωνο. Πχ. για τετράγωνο στην html σελίδα γράφουμε στο body:

```

<map name="selection1">
    <area shape="rect" coords="10,10,50,50" href="hspot1.html">
</map>
```

Java Applet: Χρησιμοποιούμε την ετικέτα applet. Πχ. για να εισάγουμε το java applet bubble.class γράφουμε:

```
<applet code="bubble.class" width="300px" height="300px">Draw
bubbles</applet>
```

Αριθμημένη λίστα: Στο body της html σελίδας γράφουμε:

```
<ol>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ol>
```

Αυτό θα εισάγει μια λίστα με δεκαδική αρίθμηση.

Μπορούμε όμως να ορίσουμε στο αρίθμηση με κεφαλαία ή πεζά γράμματα (type="A" ή type="a") ή κεφαλαίους/μικρούς λατινικούς αριθμούς (type="I" ή type="i").

Μπορούμε επίσης να εισάγουμε μη αριθμημένη λίστα. Στο παρακάτω παράδειγμα θα εμφανιστούν οι γραμμές με bullets μπροστά:

```
<ul>
    <li>Item 1</li>
```

```
</li>Item 2</li>
<li>Item 3</li>
</ul>
```

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

Τα ερωτήματα είναι εντολές προς τη βάση δεδομένων για ανάκτηση, εισαγωγή, επεξεργασία ή διαγραφή δεδομένων. Με απλά λόγια είναι ο τρόπος που επικοινωνεί ο χρήστης με τη βάση δεδομένων.

127. Τι είναι ένα σύστημα διαχείρισης βάσης δεδομένων;

Είναι ένα λογισμικό που επιτρέπει στους χρήστες να δημιουργούν, οργανώνουν, διαχειρίζονται και να ανακτούν δεδομένα από μία ή περισσότερες βάσεις δεδομένων.

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

Χρησιμοποιούμε το Εννοιολογικό Μοντέλο και ειδικότερα το Διάγραμμα Οντοτήτων-Συσχετίσεων (Entity Relationship Diagram). Το διάγραμμα ER μας επιτρέπει να απεικονίσουμε με αφαιρετικό τρόπο αντικείμενα (οντότητες), τις ιδιότητες τους και να ορίσουμε συσχετίσεις μεταξύ των οντοτήτων.

Τα διαγράμματα ER είναι εύκολο να μεταφραστούν στην συνέχεια σε οποιαδήποτε βάση δεδομένων δημιουργώντας πίνακες, κλειδιά, ξένα κλειδιά κλπ.

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

Σημαίνει ότι η εγγραφή μιας οντότητας μπορεί να συνδέεται με μία ή περισσότερες εγγραφές μιας άλλης οντότητας αλλά οι εγγραφές της δεύτερης συνδέονται μόνο με μία εγγραφή της πρώτης.

Παράδειγμα: Η οντότητα ΠΕΛΑΤΗΣ με την οντότητα ΠΑΡΑΓΓΕΛΙΑ έχει σχέση ένα προς πολλά. Δηλαδή ένας πελάτης μπορεί να έχει πολλές παραγγελίες αλλά μία παραγγελία θα ανήκει μόνο σε ένα πελάτη.

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

Μια λειτουργική εξάρτηση είναι ένας περιορισμός που καθορίζει τη σχέση δύο ή περισσότερων γνωρισμάτων ενός πίνακα μεταξύ τους. Γενικότερα μια λειτουργική εξάρτηση $X \rightarrow Y$ μεταξύ δύο συνόλων γνωρισμάτων X και Y ενός πίνακα καθορίζει ότι αν το X είναι το ίδιο τότε και το Y θα είναι το ίδιο. Για παράδειγμα στον πίνακα ΦΟΙΤΗΤΗΣ η λειτουργική εξάρτηση $ID \rightarrow \text{Όνομα}$ καθορίζει ότι αν το ID είναι το ίδιο τότε και το όνομα θα είναι το ίδιο.

Μορφές λειτουργικών εξαρτήσεων:

- Πλήρης εξάρτηση
- Μερική εξάρτηση
- Μεταβατική εξάρτηση
- Τετριμμένη εξάρτηση

131. Ποια είναι τα στάδια ανάπτυξης μίας βάσης δεδομένων;

Ανάλυση απαιτήσεων: Περιγράφουμε τις απαιτήσεις του συστήματος που πρέπει να υλοποιηθούν.

Εννοιολογική σχεδίαση: Σχεδιάζουμε το μοντέλο Οντοτήτων Συσχετίσεων (διάγραμμα ER).

Λογική σχεδίαση: Με βάση το διάγραμμα ER σχεδιάζουμε τους πίνακες της βάσης δεδομένων και τις συσχετίσεις τους.

Φυσική σχεδίαση: Βελτιστοποιούμε την υλοποίηση για γρήγορη και αποδοτική διαχείριση των δεδομένων (πχ προσθέτουμε index όπου χρειάζεται για γρήγορη ανεύρεση).

Υλοποίηση της Εφαρμογής και των μηχανισμών προστασίας.

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

Το supertype είναι μια γενική οντότητα που περιέχει κοινά γνωρίσματα με άλλες οντότητες (subtype) που εξειδικεύουν περισσότερο τα χαρακτηριστικά της.

Παράδειγμα supertype:

Οντότητα ΕΡΓΑΖΟΜΕΝΟΣ. Έχει κοινά γνωρίσματα ΑΦΜ, Όνομα.

Παράδειγμα subtype:

Οντότητα ΜΟΝΙΜΟΣ. Έχει επιπλέον γνωρίσματα Μισθός.

Οντότητα ΣΥΜΒΑΣΙΟΥΧΟΣ. Έχει επιπλέον γνωρίσματα Διάρκεια_σύμβασης.

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

Κανονικοποίηση είναι μια τεχνική οργάνωσης των δεδομένων ώστε να εξλειφθούν τυχόν πλεονασμοί και ανωμαλίες στις εγγραφές. Αυτό επιτυγχάνεται με τη διαίρεση ενός πίνακα σε μικρότερους πίνακες αλληλοσυνδεδεμένους μεταξύ τους.

Η πρώτη κανονική μορφή (1NF) εξασφαλίζει ότι κάθε στήλη πίνακα περιέχει μία μοναδιαία τιμή και δεν υπάρχουν επαναλαμβανόμενες εγγραφές.

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

Το αρχείο δεικτών χρησιμοποιείται για τη γρήγορη αναζήτηση ή ταξινόμηση των δεδομένων ενός πίνακα. Είναι ένα ξεχωριστό αρχείο που περιέχει δείκτες προς τις εγγραφές του πίνακα. Index δημιουργούμε για τα πεδία του πίνακα που αναζητούμε συχνά. Προσφέρουν ταχύτητα στην ανεύρεση αλλά μεγαλώνουν το μέγεθος της βάσης δεδομένων.

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

Το πρωτεύον κλειδί είναι το πεδίο του πίνακα που περιέχει μοναδικές τιμές για κάθε εγγραφή του.

Το πρωτεύον κλειδί μπορεί να είναι και συνδυασμός πεδίων αρκεί αυτός να είναι μοναδικός.

Για παράδειγμα σε ένα πίνακα φοιτητών ο αριθμός μητρώου είναι πρωτεύον κλειδί καθώς είναι μοναδικός για κάθε φοιτητή.

Το ξένο κλειδί σε ένα πίνακα είναι το πρωτεύον κλειδί ενός άλλου πίνακα και συνδέει τους δύο πίνακες μεταξύ τους.

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

Στην SQL υποστηρίζονται αριθμητικοί, αλφαριθμητικοί, και ημερομηνίας/ώρας τύποι δεδομένων.

Αριθμητικοί:

INTEGER (INT): Ακέραιοι αριθμοί

FLOAT, DOUBLE, DECIMAL: Δεκαδικοί αριθμοί μονής ή διπλής ακρίβειας.

BOOLEAN: Λαμβάνουν τιμές true (1) ή false (0).

Αλφαριθμητικοί:

CHAR (n): Αλφαριθμητικοί χαρακτήρες σταθερού μήκους n.

VARCHAR (n): Αλφαριθμητικοί μεταβλητού μήκους με μέγιστο μήκος n.

Ημερομηνίας/Ωρας:

DATE, TIME, YEAR, DATETIME: Αποθηκεύουν ημερομηνία και ώρα

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 ανταποκρίνεται σε ερωτήματα που υποβάλλονται ταυτόχρονα από πολλούς διαφορετικούς χρήστες. Διαφυλάσσει την ασφάλεια των δεδομένων επιτρέποντας την πρόσβαση μόνο στις περιοχές που επιτρέπουν τα δικαιώματα του κάθε χρήστη ή ομάδας χρηστών.

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

Τα καταναμεμημένα συστήματα βάσεων δεδομένων αποτελούνται από βάσεις δεδομένων που αποθηκεύονται σε διαφορετικές τοποθεσίες και πολλαπλούς υπολογιστές που συνδέονται μεταξύ τους μέσω δικτύου. Για παράδειγμα σε έναν όμιλο εταιριών μπορεί οι συναλλαγές να αποθηκεύονται τοπικά σε κάθε θυγατρική αλλά συνδέονται λογικά μεταξύ τους σε μια κεντρική βάση.

Προσφέρουν ασφάλεια καθώς αν πέσει μία τοπική βάση το σύστημα θα εξακολουθήσει να λειτουργεί. Επίσης μεγαλύτερη ταχύτητα καθώς οι πληροφορίες αναζητούνται πρώτα τοπικά. Είναι όμως πολύπλοκα στην υλοποίηση και έχουν αυξημένο κόστος συντήρησης.

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

Τα ανοικτά συστήματα διασύνδεσης βάσεων δεδομένων επιτρέπουν την επικοινωνία εφαρμογών και βάσεων δεδομένων ανεξαρτήτως του τύπου ή κατασκευαστή τους. Στόχος είναι η πρόσβαση της εφαρμογής σε διαφορετικές βάσεις δεδομένων χωρίς εξειδικευμένο κώδικα για την κάθε μία.

Το ODBC είναι ένα πρότυπο που δημιουργήθηκε από τη Microsoft. Είναι ανεξάρτητο από λειτουργικά συστήματα, γλώσσες προγραμματισμού και τύπους βάσεων δεδομένων. Ανάμεσα στη βάση δεδομένων και την εφαρμογή παρεμβάλλεται ένας driver ODBC. Αυτός δέχεται τα αιτήματα από την εφαρμογή, τα μεταφράζει κατάλληλα προς τη βάση δεδομένων και επιστρέφει τις απαντήσεις πίσω στην εφαρμογή.

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

Με τον όρο ακεραιότητα αναφορών αναφερόμαστε στους κανόνες που προφυλάσσουν τις υφιστάμενες σχέσεις μεταξύ των πινάκων μιας σχεσιακής βάσης δεδομένων. Έτσι για παράδειγμα αν αλλάξουμε το πρωτεύον κλειδί μιας εγγραφής αυτό θα αλλάξει αυτόματα και σε όσους πίνακες το χρησιμοποιούν σαν ξένο κλειδί.

Παράδειγμα:

Έχουμε τον πίνακα ΠΕΛΑΤΕΣ και τον πίνακα ΠΑΡΑΓΓΕΛΙΕΣ. Οι ΠΑΡΑΓΓΕΛΙΕΣ έχουν ξένο κλειδί τον Κωδικό Πελάτη που είναι πρωτεύον κλειδί στον πίνακα ΠΕΛΑΤΕΣ.

Αν διαγράψουμε ένα πελάτη τότε οι παραγγελίες που έχει κάνει ο συγκεκριμένος πελάτης θα έχουν άκυρο Κωδικό πελάτη. Η ακεραιότητα αναφοράς λύνει το πρόβλημα είτε απαγορεύοντας τη διαγραφή του πελάτη είτε διαγράφοντας τον πελάτη μαζί με όλες τις παραγγελίες του. Η επιλογή ορίζεται κατά τη συσχέτιση των δύο πινάκων μεταξύ τους.

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

Τα βασικά μοντέλα βάσεων δεδομένων είναι:

Ιεραρχικό: Τα δεδομένα οργανώνονται σε μορφή δέντρου όπου κάθε κόμβος είναι μία εγγραφή και οι σχέσεις τους είναι ιεραρχικές (γονέας – παιδί).

Δικτυωτό: Μοιάζει με το ιεραρχικό μοντέλο αλλά κάθε εγγραφή μπορεί να έχει πολλούς γονείς ή παιδιά.

Σχεσιακό: Τα δεδομένα οργανώνονται σε πίνακες και κάθε πίνακας αποτελείται από στήλες και γραμμές.

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 για τη διαχείριση των δεδομένων. Αυτές οι εντολές επιτρέπουν την εισαγωγή, τροποποίηση, διαγραφή και αναζήτηση εγγραφών.

Γλώσσα ορισμού δεδομένων (DDL): Είναι το σύνολο των εντολών για τη διαχείριση του σχήματος της βάσης δεδομένων. Αυτές οι εντολές επιτρέπουν την δημιουργία, τροποποίηση ή διαγραφή πινάκων, τον καθορισμό των δικαιωμάτων χρηστών κλπ.

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

Έστω ότι έχουμε τον παρακάτω πίνακα στη βάση δεδομένων με πρωτεύον κλειδί το ΑΜ:

<u>ΑΜ</u>	<u>ΟΝΟΜΑ</u>	<u>ΓΕΝΟΣ</u>	<u>ΜΑΘΗΜΑ</u>	<u>ΒΑΘΜΟΣ</u>
1	ΓΙΩΡΓΟΣ	ΑΝΔΡΑΣ	ΧΗΜΕΙΑ, ΦΥΣΙΚΗ	8, 9
2	ΜΑΡΙΑ	ΓΥΝΑΙΚΑ	ΜΑΘΗΜΑΤΙΚΑ	8
3	ΝΙΚΟΣ	ΑΝΔΡΑΣ	ΦΥΣΙΚΗ	7

1η Κανονική Φόρμα (1NF): Κάθε πεδίο πρέπει να περιέχει μόνο μία τιμή.

Στον προηγούμενο πίνακα τα πεδία ΜΑΘΗΜΑ και ΒΑΘΜΟΣ περιέχουν περισσότερες από μία τιμές άρα θα πρέπει να διαχωριστούν:

ΑΜ	ΟΝΟΜΑ	ΓΕΝΟΣ	ΜΑΘΗΜΑ	ΒΑΘΜΟΣ
1	ΓΙΩΡΓΟΣ	ΑΝΔΡΑΣ	ΧΗΜΕΙΑ	8
1	ΓΙΩΡΓΟΣ	ΑΝΔΡΑΣ	ΦΥΣΙΚΗ	9
2	ΜΑΡΙΑ	ΓΥΝΑΙΚΑ	ΜΑΘΗΜΑΤΙΚΑ	8
3	ΝΙΚΟΣ	ΑΝΔΡΑΣ	ΦΥΣΙΚΗ	7

2η Κανονική Φόρμα (2NF): Να είμαστε ήδη σε 1NF. Κάθε πεδίο που δεν είναι κλειδί να εξαρτάται πλήρως από το πρωτεύον κλειδί.

Βλέπουμε ότι το ΜΑΘΗΜΑ δεν εξαρτάται από το ΑΜ (Αριθμός Μητρώου) του μαθητή. Επομένως χωρίζουμε τον πίνακα σε δύο πίνακες και τους συσχετίζουμε βάζοντας ξένο κλειδί στον πίνακα του ΜΑΘΗΤΗ το πρωτεύον κλειδί του πίνακα ΜΑΘΗΜΑ:

ΑΜ	ΟΝΟΜΑ	ΓΕΝΟΣ	ΚΩΔ_ΜΑΘΗΜΑΤΟΣ	ΒΑΘΜΟΣ
1	ΓΙΩΡΓΟΣ	ΑΝΔΡΑΣ	1	8
1	ΓΙΩΡΓΟΣ	ΑΝΔΡΑΣ	2	9
2	ΜΑΡΙΑ	ΓΥΝΑΙΚΑ	3	8
3	ΝΙΚΟΣ	ΑΝΔΡΑΣ	2	7

ΚΩΔ_ΜΑΘΗΜΑΤΟΣ	ΜΑΘΗΜΑ
1	ΧΗΜΕΙΑ
2	ΦΥΣΙΚΗ
3	ΜΑΘΗΜΑΤΙΚΑ

3η Κανονική Φόρμα (3NF): Να είμαστε ήδη σε 2NF. Να μην υπάρχουν μεταβατικές εξαρτήσεις, δηλαδή κανένα πεδίο να μην εξαρτάται από άλλα πεδία εκτός από αυτό του πρωτεύοντος κλειδιού. Στον παραπάνω πίνακα του μαθητή αν αλλάξει το ΟΝΟΜΑ είναι πιθανό να αλλάξει και το ΓΕΝΟΣ, άρα υπάρχει εξάρτηση ενώ κανένα από αυτά τα πεδία δεν είναι πρωτεύον κλειδί. Διαχωρίζουμε λοιπόν περαιτέρω τον πίνακα:

ΑΜ	ΟΝΟΜΑ	ΚΩΔ_ΓΕΝΟΥΣ	ΚΩΔ_ΜΑΘΗΜΑΤΟΣ	ΒΑΘΜΟΣ
1	ΓΙΩΡΓΟΣ	1	1	8
1	ΓΙΩΡΓΟΣ	1	2	9
2	ΜΑΡΙΑ	2	3	8
3	ΝΙΚΟΣ	1	2	7

ΚΩΔ_ΓΕΝΟΥΣ	ΓΕΝΟΣ
1	ΑΝΔΡΑΣ
2	ΓΥΝΑΙΚΑ

ΚΩΔ_ΜΑΘΗΜΑΤΟΣ	ΜΑΘΗΜΑ
1	ΧΗΜΕΙΑ
2	ΦΥΣΙΚΗ
3	ΜΑΘΗΜΑΤΙΚΑ

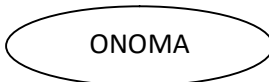
144. Τί είναι το Μοντέλο Οντοτήτων Συσχετίσεων(ΜΟΣ); Παρουσιάστε σχηματικά πώς απεικονίζονται στο ΜΟΣ οι οντότητες (entity), οι ιδιότητες των οντοτήτων (attributes), τα πρωτεύοντα κλειδιά (primary key) και οι σχέσεις (relation type και identifying relationship type).

Το μοντέλο Οντοτήτων Συσχετίσεων αποτελεί μια γενική περιγραφή της δομής μιας βάσης δεδομένων. Το διάγραμμα ER (Entity Relationship Diagram) χρησιμοποιεί γεωμετρικά σχήματα για να απεικονίσει τις οντότητες, τις ιδιότητες τους και τις σχέσεις μεταξύ των οντοτήτων. Οι οντότητες και οι ιδιότητες που συσχετίζονται ενώνονται με ευθύγραμμα τμήματα.

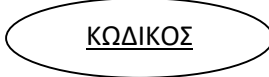
Οι οντότητες απεικονίζονται με ορθογώνιο παραλληλόγραμμο:



Οι ιδιότητες των οντοτήτων με έλλειψη:



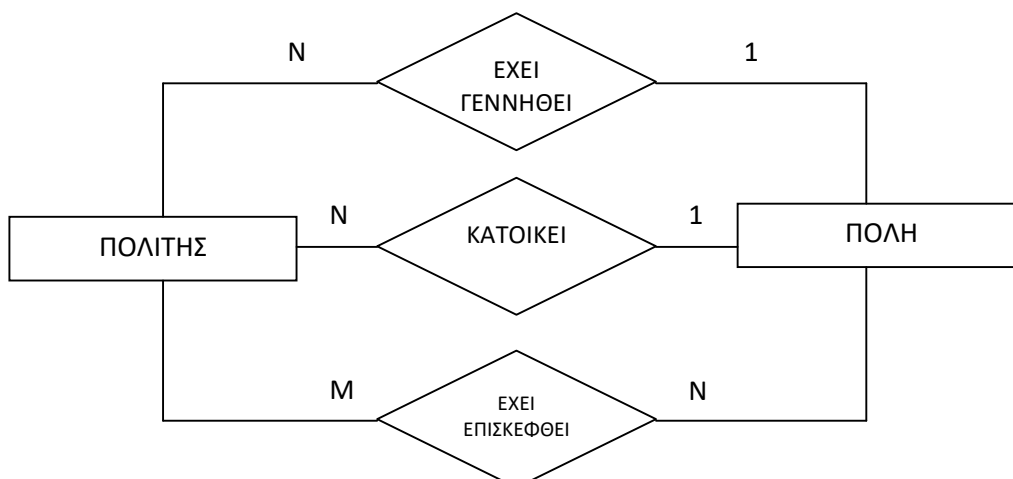
Τα πρωτεύοντα κλειδιά με έλλειψη και υπογράμμιση:



Οι σχέσεις απεικονίζονται με ρόμβο (στα ευθύγραμμα τμήματα που τις ενώνουν με τις οντότητες αναγράφουμε τον τύπο σχέσης):



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++.

Η βασική διαφορά είναι ότι η C++ υποστηρίζει αντικειμενοστραφή προγραμματισμό ενώ η C όχι. Πέραν τούτου η C++ προσθέτει και άλλα χαρακτηριστικά όπως:

- Πολυμορφισμό, ενθυλάκωση και κληρονομικότητα.
- Εικονικές συναρτήσεις.
- Υπερφόρτωση συναρτήσεων και τελεστών.
- Χώροι Ονομάτων (Namespaces) για την αποφυγή συγκρούσεων στα ονόματα μεταβλητών.
- Πρότυπα (Templates).
- Διαχείριση Εξαιρέσεων (Exception Handling).
- Ορισμός μεταβλητών ως αναφορά.
- Παρέχει έτοιμες δυναμικές δομές δεδομένων όπως vectors και lists.
- Παρέχει αλγόριθμους ταξινόμησης και αναζήτησης (sort, search).

Επίσης υπάρχουν και κάποιες διαφορές στις εντολές όπως:

- Στη C η δυναμική διαχείριση μνήμης γίνεται με συναρτήσεις όπως malloc(), free() ενώ στη C++ χρησιμοποιούμε τους τελεστές new και delete.

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

Η κλάση είναι το πρότυπο που περιγράφει τις ιδιότητες και μεθόδους της δομής δεδομένων. Το αντικείμενο είναι ένα στιγμιότυπο της κλάσης.

Για παράδειγμα μπορούμε να ορίσουμε μια κλάση:

```
public class Car {  
    private String brand, color;  
  
    Car(String brand, String color) {  
        this.brand = brand;  
        this.color = color;  
    }  
}
```

Αυτή η κλάση περιγράφει ότι τα αντικείμενα που θα δημιουργούνται από αυτήν θα έχουν γνωρίσματα brand και color.

Στη συνέχεια μπορούμε να ορίσουμε 2 αντικείμενα

```
Car myCar = new Car("Toyota", "red");  
Car yourCar = new Car("Audi", "blue");
```

Αυτό θα δεσμεύσει χώρο στη μνήμη για τα myCar και yourCar με τα συγκεκριμένα γνωρίσματα.

Μπορούμε να ορίσουμε κι άλλα αντικείμενα και για κάθε αντικείμενο θα δεσμεύεται χώρος στη μνήμη.

Αν θελήσουμε να τυπώσουμε τα myCar.color και yourCar.color θα πάρουμε απάντηση red και blue.

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

Στο δομημένο προγραμματισμό η ροή εκτέλεσης του προγράμματος καθορίζεται από τις διαδικασίες και συναρτήσεις που περιέχει. Αν χρειαστεί, θα ζητηθεί από το χρήστη να εισάγει κάποια στοιχεία όμως ο χρήστης παίζει δευτερεύοντα ρόλο. Δομικό στοιχείο στην εκτέλεση είναι οι διαδικασίες και οι συναρτήσεις του προγράμματος.

Στον προγραμματισμό οδηγούμενο από γεγονότα πρωταγωνιστικό ρόλο στη ροή εκτέλεσης του προγράμματος έχει ο χρήστης. Ανάλογα με τις επιλογές του εκτελείται και ο αντίστοιχος κώδικας για τη συγκεκριμένη επιλογή.

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

Η κλάση είναι το πρότυπο που καθορίζει τις ιδιότητες και μεθόδους μιας δομής δεδομένων. Η κλάση ορίζει τι ιδιότητες και συμπεριφορές θα έχουν τα αντικείμενα που θα ανήκουν σ' αυτήν. Δεν καταλαμβάνει χώρο στη μνήμη μέχρι να δημιουργηθεί κάποιο αντικείμενο.

Το στιγμιότυπο είναι η υλοποίηση μιας κλάσης δηλαδή ένα αντικείμενο που δεσμεύει χώρο στη μνήμη. Βασίζεται στις ιδιότητες και συμπεριφορές που έχουν ορισθεί στην κλάση αλλά μπορεί να έχει τις δικές του τιμές για κάθε μία.

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

Ο constructor είναι μία μέθοδος που καλείται όταν δημιουργείται ένα αντικείμενο μιας κλάσης. Μπορεί να πάρει ορίσματα και χρησιμοποιείται για να δώσουμε κατά τη δημιουργία τιμές στις μεταβλητές του αντικειμένου ή να καλέσουμε άλλες μεθόδους του.

Ο destructor είναι μια μέθοδος που καλείται όταν το αντικείμενο παύει να υπάρχει ώστε να αποδεσμεύσουμε το χώρο στη μνήμη που καταλάμβαναν οι δομές δεδομένων του.

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

Ενθυλάκωση σημαίνει ότι τα ιδιωτικά γνωρίσματα και οι μέθοδοι της κλάσης δεν φαίνονται έξω από την κλάση. Μπορούμε να επικοινωνήσουμε με τα αντικείμενα της κλάσης μόνο μέσω των δημόσιων μεθόδων ή γνωρισμάτων.

Κληρονομικότητα ονομάζουμε τη δυνατότητα μιας κλάσης να κληρονομεί τα χαρακτηριστικά της σε μια άλλη κλάση. Η υπάρχουσα κλάση λέγεται γονέας ή υπερκλάση και η κλάση που την κληρονομεί παιδί ή υποκλάση. Η κλάση παιδί κληρονομεί όλα τα γνωρίσματα και τις μεθόδους του γονέα αλλά μπορεί να προσθέσει και δικά της γνωρίσματα ή μεθόδους που θα είναι γνωστά μόνο σε αυτή ή σε άλλες κλάσεις παιδιά που θα την κληρονομούν.

Πολυμορφισμός είναι η δυνατότητα εκτέλεσης διαφορετικών λειτουργιών μιας μεθόδου που είναι κοινή σε διαφορετικές κλάσεις (συνήθως υπερκλάσεις και υποκλάσεις) ανάλογα με το αντικείμενο που την καλεί.

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

Διαδικαστικός προγραμματισμός

Χρησιμοποιεί διαδικασίες, δηλαδή συναρτήσεις (functions) που μπορούν να κληθούν από οποιοδήποτε σημείο του προγράμματος. Είναι εύκολος στην υλοποίηση μικρών προγραμμάτων. Όσο το μέγεθος του προγράμματος μεγαλώνει τόσο δυσκολότερη γίνεται η συντήρηση του κώδικα.

Δομημένος προγραμματισμός

Είναι μια βελτίωση του διαδικαστικού. Χρησιμοποιεί κι αυτός διαδικασίες που μπορούν όμως να οργανωθούν σε modules για επαναχρησιμοποίηση του κώδικα. Επίσης στο δομημένο προγραμματισμό αποφεύγεται η μεταπήδηση σε διαφορετικά σημεία του κώδικα με χρήση εντολών όπως goto. Συνιστάται αντίθετα η χρήση εντολών ελέγχου (if, switch) και επανάληψης (for, while) για τον έλεγχο ροής του προγράμματος. Είναι κατάλληλος για μεσαίου μεγέθους προγράμματα.

Αντικειμενοστραφής προγραμματισμός

Στηρίζεται σε αντικείμενα που αλληλεπιδρούν μεταξύ τους. Κάθε αντικείμενο έχει ιδιωτικά δεδομένα και μεθόδους που δεν είναι ορατά έξω από το αντικείμενο. Αυτό προσφέρει ασφάλεια και αποφυγή λαθών. Η επικοινωνία γίνεται μόνο με δημόσιες μεθόδους ή ιδιότητες. Οι κλάσεις των αντικειμένων έχουν δυνατότητες κληρονομικότητας και πολυμορφισμού που επιτρέπουν την επαναχρησιμοποίηση του κώδικα και σε άλλα προγράμματα. Χρειάζεται περισσότερο σχεδιασμό για την υλοποίηση αλλά χρησιμοποιείται σε μεγάλα ή σύνθετα προγράμματα καθώς είναι εύκολη η συντήρηση και ο εντοπισμός λαθών και προσφέρει περισσότερη ασφάλεια.

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

Με την οδηγία #include <iostream> συμπεριλαμβάνουμε την τυπική βιβλιοθήκη εισόδου/εξόδου της C++ στο πρόγραμμά μας. Αυτή η βιβλιοθήκη περιέχει εντολές για εισαγωγή δεδομένων από το χρήστη ή κάποιο αρχείο (cin) και την εξαγωγή δεδομένων σε αρχείο ή την εμφάνιση στην οθόνη (cout).

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>
using namespace std;

int main()
{
    cout << "Ένα απλό Πρόγραμμα" << endl;
    cout << "Είναι γραμμένο σε C++" << endl;
    cout << "Η C++ είναι αντικειμενοστραφής γλώσσα προγραμματισμού" << endl;
    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 σε χρόνο εκτέλεσης είναι σφάλματα που προκύπτουν κατά την εκτέλεση της εφαρμογής. Τέτοια λάθη είναι τα λογικά λάθη δηλαδή το αποτέλεσμα δεν είναι το αναμενόμενο. Πχ αντί να προσθέσουμε δύο αριθμούς τους αφαιρούμε.

Τα λάθη κατά τη μεταγλώττιση είναι λάθη που εντοπίζονται από τον compiler κι έτσι το πρόγραμμά μας δεν μπορεί να μεταφραστεί σε εκτελέσιμο πρόγραμμα. Μπορεί να είναι συντακτικά λάθη (πχ έχουμε ξεχάσει το ερωτηματικό ; στο τέλος μιας εντολής). Θα μπορούσε επίσης να είναι κάποια μη επιτρεπτή ενέργεια όπως η μετατροπή ενός double σε ακέραιο χωρίς explicit casting.

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

Η if εκτελείται μόνον όταν η συνθήκη ελέγχου είναι αληθής.

Αν ακολουθείται από else if τότε μπορούμε να ελέγξουμε και αν κάποια άλλη συνθήκη είναι αληθής ώστε να εκτελεστεί κώδικας. Μπορούμε να έχουμε πολλές else if συνεχόμενες.

Παράδειγμα if:

```
if (a > 10)
{
    cout << a << endl;
}
```

Παράδειγμα else if:

```
if (a <= 10)
{
    cout << "Small number" << endl;
}
else if (a > 100)
{
    cout << "Large number" << endl;
}
else
{
    cout << a << endl;
}
```

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

Η C++ για λόγους συμβατότητας με την C δέχεται τους πίνακες χαρακτήρων να υποδηλώνουν ένα string. Κάθε στοιχείο του πίνακα είναι ένας χαρακτήρας. Το τελευταίο στοιχείο είναι το null δηλαδή ο χαρακτήρας \0 που υποδηλώνει το τέλος του string. Για το λόγο αυτό αν θέλουμε να αποθηκεύσουμε ένα string 6 χαρακτήρων ο πίνακας θα πρέπει να έχει μέγεθος 7. Η διαχείριση των πινάκων χαρακτήρων (όπως αντιγραφή ή σύγκριση) γίνεται με χρήση ειδικών συναρτήσεων που περιέχονται στη βιβλιοθήκη <cstring>.

Η C++ όμως προσφέρει στην STL (Standard Template Library) την κλάση std::string που προσφέρει strings δυναμικού μεγέθους και απλοποιημένες λειτουργίες για προσθήκη, αντιγραφή, σύγκριση ή προσπέλαση.

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

Το for εκτελεί ένα συγκεκριμένο αριθμό επαναλήψεων. Παράδειγμα:

```
for (int i = 0; i < 10; i++)  
{  
    cout << i << endl;    // θα τυπώσει τους αριθμούς 0 έως 9  
}
```

Οι while και do while εκτελούνται όσο η συνθήκη ελέγχου είναι αληθής. Η διαφορά τους είναι ότι στη while ο έλεγχος γίνεται στην αρχή του βρόγχου ενώ στη do while στο τέλος. Αυτό σημαίνει ότι η do while εκτελείται πάντα τουλάχιστον μία φορά. Παράδειγμα:

```
int a = 0;  
while (a)  
{  
    cout << "Hello" << endl;    // Δεν θα τυπωθεί τίποτα  
}  
  
do  
{  
    cout << "Hello" << endl;    // θα τυπωθεί η λέξη Hello  
} while (a);
```

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

Η υπερφόρτωση επιτρέπει τη χρήση πολλών συναρτήσεων με το ίδιο όνομα που όμως δέχονται διαφορετικό αριθμό ή διαφορετικό τύπο ορισμάτων.

Το ίδιο ισχύει και για τους κατασκευαστές (constructors) που ουσιαστικά είναι συναρτήσεις που καλούνται κατά τη δημιουργία ενός αντικειμένου μιας κλάσης.

Ο μεταγλωττιστής ανάλογα με τα ορίσματα που θα χρησιμοποιήσουμε κατά την κλήση της συνάρτησης ή τη δημιουργία αντικειμένου θα εκτελέσει τη σωστή συνάρτηση ή κατασκευαστή.

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

Η C++ για λόγους συμβατότητας με την C δέχεται τους πίνακες χαρακτήρων να υποδηλώνουν ένα string. Κάθε στοιχείο του πίνακα είναι ένας χαρακτήρας. Το τελευταίο στοιχείο είναι το null δηλαδή ο χαρακτήρας \0 που υποδηλώνει το τέλος του string. Για το λόγο αυτό αν θέλουμε να αποθηκεύσουμε ένα string 6 χαρακτήρων ο πίνακας θα πρέπει να έχει μέγεθος 7. Η διαχείριση των πινάκων χαρακτήρων (όπως αντιγραφή ή σύγκριση) γίνεται με χρήση ειδικών συναρτήσεων που περιέχονται στη βιβλιοθήκη <cstring>.

Η C++ όμως προσφέρει στην STL (Standard Template Library) την κλάση std::string που προσφέρει strings δυναμικού μεγέθους και απλοποιημένες λειτουργίες για προσθήκη, αντιγραφή, σύγκριση ή προσπέλαση.

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

Ένας πίνακας χαρακτήρων υποδηλώνει ένα string. Το μέγεθός του πρέπει να είναι κατά ένα χαρακτήρα μεγαλύτερο από το μέγιστο μήκος string που θα αποθηκευτεί. Αυτό γιατί ο τελευταίος χαρακτήρας πρέπει να είναι null, δηλαδή ο χαρακτήρας \0 που υποδηλώνει το τέλος του string.

Οι συναρτήσεις για χειρισμό των πινάκων χαρακτήρων (αντιγραφή, σύγκριση κλπ) βρίσκονται στη βιβλιοθήκη <cstring>.

Παραδείγματα:

- `char str[4] = "Home";`
Θα εμφανίσει λάθος γιατί το "Home" καταλαμβάνει 5 χαρακτήρες ("Home\0").
- `char str[5];`
Ο πίνακας μπορεί να δεχθεί string μέγιστου μήκους 4 χαρακτήρων.
- `char str[5] = "Hi";`
Θα δεσμεύσει χώρο στη μνήμη για 5 χαρακτήρες. Το στοιχείο `str[0]` θα έχει τιμή 'H', το `str[1]` τιμή 'i' και το `str[2]` τιμή '\0' δηλαδή null που υποδηλώνει το τέλος του string. Τα στοιχεία `str[3]`, `str[4]` θα έχουν απροσδιόριστες τιμές.

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>
using namespace std;
```

```
unsigned long long factorial(int n)
{
    if (n <= 1) return 1;
    return n * factorial(n - 1);
}

int main()
{
    int num;
    cout << "Enter number: ";
    cin >> num;

    if (num < 0)
    {
        cout << "Number must be positive" << endl;
        return 1;
    }

    cout << num << "! = " << factorial(num) << endl;
    return 0;
}
```

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>
using namespace std;
```

```
int main()
{
    char message[] = "C++ is an object oriented Language and is a superset
of C";
    char word[] = "is";
    int found = 0;
    int len = sizeof(message) / sizeof(message[0]);

    cout << "Message = " << message << endl;
    cout << "word = " << word << endl;
    cout << "Searching for 'word' occurrences in 'message'..." << endl;
    for (int i = 0; i < len - 1; i++)
    {
        if (message[i] == word[0] && message[i + 1] == word[1])
        {
            found++;
            cout << "Word found at position " << i << endl;
        }
    }
    cout << "The word was found " << found << " times in message." << endl;
    return 0;
}
```

164. Γράψτε δυο συναρτήσεις που θα υπολογίζουν τον Μέγιστο Κοινό Διαιρέτη (ΜΚΔ) και το Ελάχιστο κοινό πολλαπλάσιο (ΕΚΠ) δυο ακεραίων 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 ακεραίος, εκτός από το 0, που το συναντάμε κοινό στα πολλαπλάσια των δύο αριθμών. Ο μικρότερος από τα κοινά πολλαπλάσια κάποιων ακεραίων αριθμών λέγεται Ελάχιστο Κοινό Πολλαπλάσιο (Ε.Κ.Π.).

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

// Εύρεση ΜΚΔ με τον αλγόριθμο του Ευκλείδη
// που βασίζεται στο ότι ο ΜΚΔ δεν αλλάζει αν αντικαταστήσουμε τον
// μεγαλύτερο αριθμό με τη διαφορά των 2 αρχικών αριθμών. Επαναλαμβάνουμε
// μέχρι το υπόλοιπο να γίνει 0. Τότε ο τελευταίος αριθμός είναι ο ΜΚΔ
int mkd(int a, int b)
{
    while (a != b)
    {
        if (a > b)
        {
            a = a - b;
        }
        else
        {
            b = b - a;
        }
    }
    return a;
}
```

```
// ΕΚΠ = (α * β) / ΜΚΔ
int ekp(int a, int b)
{
    return (a * b) / mkd(a, b);
}

int main()
{
    int a = 21;
    int b = 6;
    cout << "ΜΚΔ(" << a << ", " << b << ") = " << mkd(a, b) << endl;
    cout << "ΕΚΠ(" << a << ", " << b << ") = " << ekp(a, b) << endl;
    return 0;
}
```

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

Με τον όρο κληρονομικότητα εννοούμε τη δυνατότητα μιας κλάσης να κληρονομεί τα γνωρίσματα και τις μεθόδους της σε άλλες κλάσεις. Σ' αυτή την περίπτωση λέμε ότι η κλάση είναι γονέας ή υπερκλάση και η κλάση που την κληρονομεί λέγεται παιδί ή υποκλάση. Η υποκλάση μπορεί να ορίσει νέα γνωρίσματα ή μεθόδους που δεν θα είναι γνωστά στην υπερκλάση.

Μια υποκλάση μπορεί με τη σειρά της να γίνει υπερκλάση σε άλλες κλάσεις. Σ' αυτή την περίπτωση έχουμε πολυεπίπεδη κληρονομικότητα.

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

class Vehicle    // Base class
{
public:
    string brand;
    string model;
};

class Car : public Vehicle    // κληρονομεί την Vehicle
{
public:
    int doors;
};

class ElectricCar : public Car    // κληρονομεί την Car
{
public:
    int batteryCapacity;    // kWh
};

class IceCar : public Car    // κληρονομεί την Car
{
public:
    string fuelType;    // Βενζίνη, Πετρέλαιο, ..
};
```

```
int main()
{
    ElectricCar ecar;
    IceCar car;

    // Τα αντικείμενα ecar, car γνωρίζουν τις ιδιότητες της κλάσης
    // car λόγω της κληρονομικότητας και της κλάσης Vehicle
    // λόγω της πολυεπίπεδης κληρονομικότητας
    ecar.brand = "Tesla";
    ecar.model = "Model S";
    ecar.doors = 5;
    ecar.batteryCapacity = 100;

    car.brand = "Honda";
    car.model = "Civic";
    car.doors = 4;
    car.fuelType = "Petrol";

    return 0;
}
```

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

RGB (Red, Green, Blue)

Είναι προσθετικό μοντέλο. Χρησιμοποιεί σαν βασικά χρώματα το κόκκινο, πράσινο και μπλε.

Λειτουργεί προσθέτοντας διαβαθμίσεις 0-255 των 3 βασικών χρωμάτων για την απεικόνιση όλου του χρωματικού φάσματος. Το μαύρο είναι RGB(0,0,0) και το λευκό RGB(255,255,255). Χρησιμοποιείται σε συστήματα που βασίζονται στο φως όπως οθόνες υπολογιστών και τηλεοράσεις.

CMY και CMYK (Cyan, Magenta, Yellow, Black)

Είναι αφαιρετικό μοντέλο. Χρησιμοποιεί 3 βασικά χρώματα κυανό, ματζέντα, κίτρινο και μαύρο στην εκτεταμένη του μορφή (CMYK). Η ανάμειξη των χρωμάτων αφαιρεί φως από το λευκό σκουραίνοντας το χρώμα. Προσφέρει μεγαλύτερη ακρίβεια στις εκτυπώσεις.

HSV (Hue, Saturation, Value)

Αντιπροσωπεύει τα χρώματα σύμφωνα με την αντίληψη του ανθρώπινου ματιού. Το Hue (απόχρωση) είναι το βασικό χρώμα ανάμεσα σε ένα δίσκο χρωμάτων 0 - 360° και ρυθμίζουμε τον κορεσμό και τη λαμπρότητα. Χρησιμοποιείται στην επεξεργασία εικόνας.

HSL (Hue, Saturation, Luminance)

Παρόμοιο με το HSV αλλά χρησιμοποιεί τη φωτεινότητα αντί για τη λαμπρότητα.

LAB

Χρωματικό μοντέλο που περιγράφει το χρώμα ανεξάρτητα από τη συσκευή με βάση την ανθρώπινη όραση. L = Luminance δηλαδή η φωτεινότητα, A = χρωματική θέση ανάμεσα στο κόκκινο και το συμπληρωματικό του πράσινο, B = χρωματική θέση ανάμεσα στο μπλε και το συμπληρωματικό του κίτρινο. Χρησιμοποιείται για επεξεργασία χρωμάτων.

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

Σ' ένα χρωματικό μοντέλο πρωτεύοντα είναι τα χρώματα που δεν μπορούν να δημιουργηθούν από άλλα. Για παράδειγμα στο χρωματικό μοντέλο RGB πρωτεύονται είναι τα χρώματα κόκκινο (Red), πράσινο (Green), μπλε (Blue) ενώ στο χρωματικό μοντέλο RYB πρωτεύονται είναι τα κόκκινο (Red), κίτρινο (Yellow) και μπλε (Blue).

Δευτερεύοντα είναι τα χρώματα που προκύπτουν από τη μίξη δύο πρωτευόντων χρωμάτων. Για παράδειγμα στο χρωματικό μοντέλο RGB το κίτρινο είναι δευτερεύον καθώς προκύπτει από τη μίξη πράσινου και κόκκινου που είναι πρωτεύοντα.

Συμπληρωματικό είναι το χρώμα που αν αναμιχθεί με ένα πρωτεύον χρώμα δίνει λευκό στα προσθετικά μοντέλα (όπως το RGB) ή μαύρο στα αφαιρετικά μοντέλα (όπως το CMYK). Για παράδειγμα στο χρωματικό μοντέλο RGB το κίτρινο είναι συμπληρωματικό του μπλε γιατί αν αναμιχθούν δίνουν λευκό.

Τα συμπληρωματικά χρώματα χρησιμοποιούνται για να δημιουργούν έντονη αντίθεση και να τραβούν την προσοχή.

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

Ανάλυση (resolution)

Κάθε εικόνα bitmap αναπαρίσταται από ένα πλήθος εικονοστοιχείων (pixels). Κάθε pixel περιέχει πληροφορίες χρώματος. Η ανάλυση εκφράζει την πυκνότητα των pixels και μετράται σε κουκίδες ανά ίντσα (DPI) ή pixels ανά ίντσα (PPI).

Βάθος χρώματος

Ορίζει πόσα χρώματα υπάρχουν στην εικόνα. Η πληροφορία ενός χρώματος χρειάζεται 1bit για να αποθηκευθεί. Έτσι αν δεσμεύσουμε n bits για την πληροφορία χρώματος του κάθε pixel θα έχουμε 2^n χρώματα. Για παράδειγμα βάθος χρώματος 24bit σημαίνει 2^{24} ή 16.7 εκατομμύρια χρώματα.

Διαστάσεις

Το πλάτος και ύψος της εικόνας σε pixels (πχ. 1024 X 768).

Μέγεθος

Το μέγεθος που καταλαμβάνει η εικόνα. Αυτό εξαρτάται από τον αριθμό των pixels, το βάθος χρώματος και τον αλγόριθμο συμπίεσης.

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

Η πληροφορία ενός χρώματος χρειάζεται 1bit ανά pixel. Στις εικόνες true color το βάθος χρώματος είναι 24bit δηλαδή 2^{24} ή 16.7 εκατομμύρια χρώματος και στις grayscale 8bit. Για να μετατρέψουμε σε bytes θα πρέπει να διαιρέσουμε με 8 (1 byte = 8 bit). Επομένως:

α) True color: $100 \times 100 \times 24 / 8 = 30.000$ bytes.

β) Grayscale: $100 \times 100 \times 8 / 8 = 10.000$ bytes.

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

BMP: Εφευρέθηκε από τη Microsoft για χρήση στα Windows. Μπορεί να εμφανίσει την εικόνα με την ίδια πιστότητα σε διαφορετικές συσκευές. Δεν χρησιμοποιεί συμπίεση οπότε δεν υπάρχουν απώλειες στην ποιότητα εικόνας. Υποστηρίζει true color οπότε είναι ιδανικό για απεικόνιση φωτογραφιών όμως τα αρχεία που δημιουργούνται έχουν μεγάλο μέγεθος. Επειδή είναι πολύ παλιό format υποστηρίζεται από πολλά λειτουργικά συστήματα και προγράμματα σχεδίασης.

GIF: Υποστηρίζει έως 256 χρώματα κι έτσι δεν είναι κατάλληλο για φωτογραφίες. Χρησιμοποιεί μη απωλεστική συμπίεση οπότε οι εικόνες έχουν μικρό μέγεθος και δεν χάνουν σε ανάλυση. Υποστηρίζει απλή διαφάνεια και animation για αυτό χρησιμοποιείται για κινούμενες εικόνες και λογότυπα.

JPG: Υποστηρίζει 16.7 εκατομμύρια και είναι ιδανικό για αποθήκευση φωτογραφιών επειδή χρησιμοποιεί συμπίεση και τα αρχεία καταλαμβάνουν μικρό χώρο. Η συμπίεση είναι απωλεστική αλλά δεν επηρεάζει πολύ τις φωτογραφίες. Δεν είναι όμως κατάλληλο για λογότυπα γιατί σε γράμματα είναι πιο εμφανής η υποβάθμιση ποιότητας.

PNG: Υποστηρίζει 16.7 εκατομμύρια χρώματα και μη απωλεστική συμπίεση. Επομένως οι φωτογραφίες αποθηκεύονται χωρίς απώλειες αλλά η συμπίεση δεν είναι τόσο αποδοτική όσο του JPG και τα αρχεία καταλαμβάνουν περισσότερο χώρο. Χρησιμοποιείται κυρίως για κατασκευή λογότυπων και μακέτας γιατί υποστηρίζει διαφάνεια πολλών επιπέδων.

171. Ποια η χρησιμότητα μάσκας κατά την επεξεργασία εικόνας;

Η μάσκα είναι ένα εργαλείο που χρησιμοποιείται για να καθορίσει ή να περιορίσει τις περιοχές της εικόνας που θα εφαρμοστούν αλλαγές ή εφέ. Είναι πολύ σημαντική για την επεξεργασία εικόνας με ακρίβεια και ευελιξία.

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

Η συμπίεση JPEG είναι ένα απωλεστικό πρότυπο συμπίεσης. Αυτό σημαίνει ότι κατά την αποθήκευση της εικόνας χάνονται οριστικά κάποιες πληροφορίες. Το πλεονέκτημα είναι ότι μπορούμε να μειώσουμε πάρα πολύ το μέγεθος του αρχείου της εικόνας. Αυτή η μείωση μεγέθους μπορεί να φτάσει το 10:1 ή και 20:1 χωρίς να γίνεται αντιληπτό από το ανθρώπινο μάτι η μείωση στην πιστότητα της εικόνας. Σε κάποιες όμως περιπτώσεις η μείωση είναι εμφανής όπως σε κείμενα ή μεγάλες αντιθέσεις χρωμάτων. Για το λόγο αυτό οι εικόνες jpeg χρησιμοποιούνται κυρίως για απεικόνιση φωτογραφιών και όχι λογότυπων ή κειμένων.

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

Το color depth καθορίζει πόσα χρώματα μπορεί να απεικονίσει μια εικόνα και μετράται σε bits. Μια εικόνα αποτελείται από pixels. Η πληροφορία του κάθε χρώματος χρειάζεται 1bit ανά pixel. Αν διαθέσουμε n bits για την απεικόνιση χρωμάτων τότε θα έχουμε 2^n χρώματα. Έτσι για βάθος χρώματος 24bit θα έχουμε 2^{24} δηλαδή 16.7 εκατομμύρια χρώματα (true color). Στον αντίποδα για χρωματικό βάθος 1bit θα έχουμε $2^1 = 2$ χρώματα, δηλαδή ασπρόμαυρη εικόνα.

Το bit depth είναι το ίδιο ακριβώς. Χρησιμοποιείται σαν εναλλακτική ορολογία.

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

Πλεονεκτήματα:

- Επειδή οι εικόνες αποτελούνται από ένα matrix pixels είναι εύκολο να τις διαχειριστούμε σε απλές λειτουργίες επεξεργασίας.
- Μπορούμε να εφαρμόσουμε επάνω τους φίλτρα.
- Υπάρχουν πολλοί τύποι bitmap (GIF, BMP, JPG, PNG κλπ) για να καλύψουν πολλές διαφορετικές ανάγκες (φωτογραφίες, λογότυπα, animation κλπ).
- Υποστηρίζονται από σχεδόν όλα τα σχεδιαστικά προγράμματα και λειτουργικά.
- Πολλοί τύποι bitmap προσφέρουν φωτορεαλιστική απεικόνιση.
- Υποστηρίζονται πολλά χρωματικά μοντέλα (RGB, CMYK, RYB, LAB κλπ).

Μειονεκτήματα:

- Οι ασυμπίεστες εικόνες καταλαμβάνουν μεγάλο αποθηκευτικό χώρο.
- Οι συμπίεσμένες εικόνες έχουν μικρότερο μέγεθος αλλά χάνουν σε πιστότητα.
- Οι κουκίδες που τις αποτελούν αρχίζουν και φαίνονται αν τις μεγεθύνουμε.

175. Αναφέρετε τους τρόπους λήψης και ψηφιοποίησης ψηφιογραφικής εικόνας.

Η λήψη μπορεί να γίνει:

- Με ψηφιακές φωτογραφικές μηχανές.
- Με σαρωτές.
- Με βιντεοκάμερες και κατάλληλο λογισμικό.
- Με καταγραφή της οθόνης του υπολογιστή από κατάλληλο λογισμικό.

Μπορούμε επίσης να δημιουργήσουμε εικόνες με σχεδιαστικά προγράμματα.

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

Δημιουργία πίνακα: CREATE TABLE

```
CREATE TABLE Person (  
    id INT,  
    Name VARCHAR(50)  
);
```

Διαγραφή πίνακα: DROP TABLE

```
DROP TABLE Person;
```

Διαγραφή δεδομένων πίνακα: DELETE

```
DELETE FROM Person  
WHERE Name = 'John';
```

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

Η όψη είναι ένας εικονικός πίνακας που προκύπτει από ένα ή περισσότερα πεδία ενός ή περισσότερων πινάκων ή άλλων όψεων. Η όψη δημιουργείται με αποστολή ερωτήματος προς τη βάση δεδομένων.

Χρησιμοποιούμε τις όψεις είτε για να εμφανίσουμε μόνο τα πεδία που ενδιαφέρουν το χρήστη είτε για να εμφανίσουμε μόνο τα πεδία που επιτρέπεται η πρόσβαση στο χρήστη. Το πλεονέκτημα της όψης είναι ότι αφού τη δημιουργήσουμε μπορούμε να την καλέσουμε ξανά χωρίς να χρειάζεται να ξαναγράψουμε περίπλοκα ερωτήματα.

Δημιουργία όψης: CREATE VIEW ViewName
CREATE VIEW FriendList AS
SELECT Name
FROM AddressBook;

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

Το ευρετήριο είναι ένα αρχείο δεικτών προς τις εγγραφές ενός πίνακα. Χρησιμεύει για τη γρήγορη αναζήτηση ή ταξινόμηση των εγγραφών του πίνακα. Επειδή το ευρετήριο καταλαμβάνει χώρο στο δίσκο, χρησιμοποιούμε index για τα πεδία που αναζητούμε συχνά. Παράδειγμα:

```
CREATE INDEX idxName  
ON Person (Name);
```

Θα δημιουργήσει index με όνομα idxName για το πεδίο Name του πίνακα Person.

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

Τα stored procedures είναι κώδικας SQL που μπορούμε να αποθηκεύσουμε και να χρησιμοποιήσουμε ξανά και ξανά. Έτσι αν έχουμε ένα περίπλοκο ερώτημα SQL μπορούμε να το αποθηκεύσουμε σε stored procedure και να το εκτελέσουμε όσες φορές θέλουμε καλώντας απλά το stored procedure.

Δημιουργία stored procedure χωρίς παραμέτρους:

```
DELIMITER //  
CREATE PROCEDURE ShowNames()  
BEGIN  
    SELECT Name FROM users;  
END //  
DELIMITER ;
```

Το DELIMITER δεν αποτελεί μέρος του stored procedure. Αλλάζει μόνο το χαρακτήρα τέλους εντολής που είναι το ελληνικό ερωτηματικό (;) σε //. Αυτό γίνεται για να μην υπάρχει σύγκρουση με το ερωτηματικό των εντολών SQL. Αφού ολοκληρωθεί η δήλωση του stored procedure το αλλάζουμε ξανά σε ερωτηματικό με το DELIMITER.

Για να εκτελέσουμε το stored procedure γράφουμε CALL ShowNames();

Δημιουργία stored procedure με παραμέτρους:

```
DELIMITER //  
CREATE PROCEDURE SearchName(IN criteria VARCHAR(50))  
BEGIN  
    SELECT * FROM users  
    WHERE name LIKE criteria;  
END //  
DELIMITER ;
```

Εδώ το stored procedure SearchName δέχεται σαν είσοδο (IN) την παράμετρο criteria που είναι αλφαριθμητική μεταβλητή (VARCHAR). Το εκτελούμε με πχ CALL SearchName("John");

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 phones (  
    Κωδικός INT,  
    Όνομα VARCHAR(50),  
    Τηλέφωνο VARCHAR(10)  
);  
  
INSERT INTO phones  
    (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 είναι εναλλακτικές ονομασίες στηλών ή πινάκων σε ένα ερώτημα SQL.

Χρησιμοποιούνται για να συντομεύσουν μεγάλα ονόματα και να απλοποιήσουν την πληκτρολόγηση ή να κάνουν πιο ευανάγνωστες τις συσχετίσεις ή αυτοσυσχετίσεις.

Στον κώδικα της εκφώνησης δίνεται το ψευδώνυμο E στον πίνακα EMPLOYEE και το S πάλι στον ίδιο πίνακα. Αυτό κάνει πιο ευανάγνωστη την αυτοσυσχέτιση των πεδίων του πίνακα.

Αν το SN (Social Security Number) είναι το ΑΜΚΑ του εργαζόμενου και SUPERSN το ΑΜΚΑ του προϊσταμένου θα πάρουμε μια λίστα με δύο στήλες. Στη μία θα είναι το όνομα του εργαζόμενου και στην άλλη το όνομα του προϊσταμένου του.

Παράδειγμα δεδομένων του πίνακα EMPLOYEE:

SN	NAME	SUPERSN
101	GEORGE	102
102	MARIA	NULL
103	NIKOS	101
104	PETROS	102

Αποτελέσματα ερωτήματος:

E.NAME	S.NAME
GEORGE	MARIA
NIKOS	GEORGE
PETROS	MARIA

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

Με την εντολή DISTINCT.

Παράδειγμα:

```
SELECT DISTINCT City FROM Customer;
```

Μπορεί να έχουμε πολλούς πελάτες από την ίδια πόλη αλλά το παραπάνω ερώτημα θα εμφανίσει μόνο μία φορά την ίδια πόλη.

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.

Χρησιμοποιούμε την εντολή ORDER BY Όνομα στήλης. Σαν προεπιλογή είναι η αύξουσα ταξινόμηση (ASC). Αν θέλουμε φθίνουσα ταξινόμηση γράφουμε DESC μετά το όνομα της στήλης.

```
SELECT FirstName, LastName  
FROM Students  
ORDER BY FirstName, LastName 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);
```

Θα εμφανίσει τα ονόματα από τον πίνακα WORKER για όσους εργάτες τα πεδία ID, Name του πίνακα WORKER υπάρχουν και ταυτίζονται με τα πεδία CONNECT_ID, CONNECT_Name του πίνακα CONNECT.

Παράδειγμα δεδομένων WORKER	
ID	NAME
101	MARIA
102	GEORGE
103	PETROS

Παράδειγμα δεδομένων CONNECT	
CONNECT_ID	CONNECT_NAME
101	MARIA
110	NIKOS
103	PETROS

Αποτέλεσμα ερωτήματος	
NAME	
MARIA	
PETROS	

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

LEFT JOIN: Επιστρέφει όλες τις εγγραφές του αριστερού πίνακα (Student) και μόνο τις εγγραφές του δεξιού πίνακα (Assignment) για τις οποίες το student_id του Assignment είναι ίδιο με το πεδίο id του Student. Θα εμφανιστούν λοιπόν τα ονόματα όλων των μαθητών άσχετα αν έχουν αναλάβει εργασίες. Όπου δεν υπάρχει αντιστοιχία θα εμφανίζεται NULL στη στήλη Assignment.Title.

```
SELECT Student.Name, Assignment.Title
FROM Student
LEFT JOIN Assignment ON Student.id = Assignment.student_id;
```

RIGHT JOIN: Επιστρέφει όλες τις εγγραφές του δεξιού πίνακα (Assignment) και μόνο τις εγγραφές του αριστερού πίνακα (Student) για τις οποίες το id του Student είναι ίδιο με το πεδίο student_id του Assignment. Θα εμφανιστούν λοιπόν οι τίτλοι όλων των εργασιών άσχετα αν έχουν ανατεθεί σε μαθητές. Αν δεν υπάρχει αντιστοιχία θα εμφανίζεται NULL στη στήλη Student.Name.

```
SELECT Student.Name, Assignment.Title
FROM Student
RIGHT JOIN Assignment ON Student.id = Assignment.student_id;
```

FULL JOIN: Επιστρέφει όλες τις εγγραφές του αριστερού πίνακα (Student) και όλες τις εγγραφές του δεξιού πίνακα (Assignment) ακόμα κι αν δεν υπάρχουν αντιστοιχίσεις για τις οποίες το id του Student είναι ίδιο με το πεδίο student_id του Assignment. Θα εμφανιστούν λοιπόν οι τίτλοι όλων των εργασιών άσχετα αν έχουν ανατεθεί σε μαθητές και τα ονόματα όλων των μαθητών άσχετα αν έχουν αναλάβει εργασίες. Στις στήλες που δεν υπάρχει αντιστοιχία θα εμφανίζεται NULL.

```
SELECT Student.Name, Assignment.Title
FROM Student
FULL JOIN Assignment ON Student.id = Assignment.student_id;
```

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

Οι συναρτήσεις είναι μια ομάδα εντολών που παράγουν κάποιο αποτέλεσμα. Μια συνάρτηση μπορεί να δέχεται δεδομένα στην είσοδο που καλούνται ορίσματα και να επιστρέφει δεδομένα. Οι συναρτήσεις που βρίσκονται μέσα σε κλάσεις καλούνται μέθοδοι. Μέσω των μεθόδων οι κλάσεις μπορούν να ανταλλάσσουν δεδομένα

```
// παράδειγμα χρήσης μεθόδων για ανταλλαγή μηνυμάτων μεταξύ κλάσεων
public class Example {
    public static void main(String[] args) {
        Address address = new Address("Tsimiski 80", "Thessaloniki");
        Person person = new Person("George", address);

        // χρήση μεθόδου της Person για την εμφάνιση των στοιχείων
        // (Η κλάση Person χρησιμοποιεί μέθοδο της Address για
        // εμφάνιση της διεύθυνσης)
        person.displayPerson();
    }
}
```

```
class Address {
    private String street;
    private String city;

    Address(String street, String city) {
        this.street = street;
        this.city = city;
    }

    public String toString() {
        return street + ", " + city;
    }
}

class Person {
    private String name;
    private Address address;

    Person(String name, Address address) {
        this.name = name;
        this.address = address;
    }

    public void displayPerson() {
        System.out.println("Name: " + name);
        // Χρήση μεθόδου της Address
        System.out.println("Address: " + address.toString());
    }
}
```

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();

Αν υποθέσουμε ότι:

Person: υπερκλάση.

Student: υποκλάση της Person.

PhDStudent: υποκλάση της Student και άρα της Person.

Prof: υποκλάση της PhDStudent και άρα της Student και άρα της Person

Σύμφωνα με την αρχή της πολυμορφικότητας μπορούμε να ορίσουμε ένα αντικείμενο υποκλάσης με αναφορά στην υπερκλάση. Επομένως:

Person p1 = new Student(); Σωστό γιατί η Student είναι υποκλάση της Person.

Person p2 = new PhDStudent(); Σωστό γιατί η PhDStudent είναι υποκλάση της Person.

PhDStudent phd1 = new Student(); Λάθος γιατί η Student δεν είναι υποκλάση της PhDStudent.

Prof t1 = new Person(); Λάθος γιατί η Person δεν είναι υποκλάση της Prof.

Student s1 = new PhDStudent(); Σωστό γιατί η PhDStudent είναι υποκλάση της Student.

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

Από τον ορισμό της μια υποκλάση μπορεί να επεκτείνει μία κλάση. Δηλαδή κληρονομεί όλα τα πεδία της κλάσης γονέα και μπορεί να τα επεκτείνει δημιουργώντας δικά της. Δεν μπορεί να διαγράψει τα πεδία της υπερκλάσης γιατί δεν της ανήκουν. Ανήκουν στην κλάση γονέα.

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

Τα JFrame και JDialog είναι και τα δύο παράθυρα που κληρονομούν από την κλάση Window. Όμως όταν ανοίγει ένα παράθυρο dialog ο χρήστης μπορεί να αλληλεπιδράσει μόνο με αυτό. Δεν μπορεί να επιλέξει κάποιο άλλο ανοικτό παράθυρο της εφαρμογής όσο το dialog είναι ανοικτό. Σε αντίθεση το frame επιτρέπει στο χρήστη να αλληλεπιδράσει και με άλλα ανοικτά παράθυρα. Επίσης το dialog δεν διαθέτει minimize/maximize buttons.

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

Τα Menu και MenuBar είναι κλάσεις που χρησιμοποιούνται στην κατασκευή γραφικού περιβάλλοντος.

Το MenuBar είναι η οριζόντια μπάρα που εμφανίζεται στο επάνω μέρος του παραθύρου και περιέχει τα Menu.

Τα Menu είναι πτυσσόμενες λίστες που εμφανίζονται όταν τις επιλέξουμε από το MenuBar και περιέχουν MenuItem που είναι τα στοιχεία που επιλέγει ο χρήστης.

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

Τα Menu είναι πτυσσόμενες λίστες που εμφανίζονται όταν τις επιλέξουμε από το MenuBar και περιέχουν MenuItem που είναι τα στοιχεία που επιλέγει ο χρήστης.

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

Η επανεκκίνηση (Restart) ενός applet σημαίνει την πλήρη διακοπή του και την έναρξη από την αρχή. Όλα τα δεδομένα και οι πόροι που χρησιμοποιούσε το applet καθαρίζονται και αρχικοποιούνται ξανά. Συμβαίνει όταν πατάμε F5 στον browser και ξαναφορτώνουμε τη σελίδα.

Η επαναφόρτωση (Reload) διακόπτει προσωρινά την εκτέλεση του applet. Τα δεδομένα του applet δεν διαγράφονται. Μετά την επαναφόρτωση, το applet συνεχίζει την εκτέλεση από το σημείο που βρισκόταν. Συμβαίνει όταν αλλάζουμε καρτέλα στον browser και μετά γυρνάμε πάλι στην προηγούμενη.

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

Ο constructor είναι ειδική μέθοδος κλάσης που εκτελείται αυτόματα όταν δημιουργείται ένα αντικείμενο της κλάσης. Έχει το ίδιο όνομα με την κλάση. Δέχεται ορίσματα αλλά δεν επιστρέφει κανένα δεδομένο. Χρησιμοποιείται για την αρχικοποίηση των ιδιοτήτων της κλάσης και την εκτέλεση εντολών κατά τη δημιουργία του αντικειμένου

```
public class Example {  
    public static void main(String[] args) {  
        Person person = new Person("Γιώργος", "Δημητρίου");  
    }  
}
```

```
        person.displayInfo();
    }
}

class Person {
    private String firstName, lastName;

    // Constructor
    Person (String name, String surName) {
        firstName = name;
        lastName = surName;
    }

    public void displayInfo() {
        System.out.println("Όνομα: " + firstName);
        System.out.println("Επίθετο: " + lastName);
    }
}
```

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

Τα κύρια μέρη της JVM είναι:

Class Loader: Φορτώνει και αρχικοποιεί τις κλάσεις της Java.

Runtime Data Area: Η μνήμη που χρησιμοποιεί η Java για την εκτέλεση προγραμμάτων.

Execution Engine: Εκτελεί το bytecode ενός προγράμματος.

Native Interface: Διασύνδεση με μεθόδους που είναι γραμμένες σε άλλες γλώσσες όπως C++.

Native libraries: Βιβλιοθήκες που απαιτούνται για την εκτέλεση των native methods.

Μέθοδοι που τρέχει η JVM:

Main method: Η κύρια μέθοδος από όπου ξεκινάει η εκτέλεση μιας εφαρμογής.

Static initializers: Εκτελούνται όταν φορτώνεται μια κλάση στη μνήμη.

Instance methods: Μη στατικές μέθοδοι που καλούνται μέσω αντικειμένων.

Native methods: Μέθοδοι που καλούνται από τη Java και εκτελούνται με native κώδικα.

Garbage collector: Διαχειρίζεται τα αντικείμενα που δεν χρησιμοποιούνται.

195. Ποιοι είναι οι τύποι μεταβλητών στην Java;

Local: Ορατές μόνο στο μπλοκ που δηλώθηκαν.

Class/Static: Είναι κοινές για όλα τα αντικείμενα της κλάσης

Instance: Ανήκουν στα αντικείμενα της κλάσης και μπορεί να διαφέρουν σε κάθε αντικείμενο.

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

Το ? είναι ternary operator. Αν το $k > j$ τότε το $\text{max} = k$ αλλιώς $\text{max} = j$. Επομένως αφού $10 > 5$ το max θα γίνει 10.

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

Με τις εξαιρέσεις μπορούμε να διαχειριστούμε σφάλματα που θα προκαλούσαν το κρέμασμα της εφαρμογής. Για παράδειγμα να διαβάσουμε τα δεδομένα ενός αρχείου που δεν υπάρχει.

Τις εξαιρέσεις διαχειριζόμαστε με την εντολή try catch όπου στο block try τοποθετούμε τις εντολές που μπορεί να προκαλέσουν εξαίρεση και στο block catch τις διαχειριζόμαστε.

Παράδειγμα 1: Προσπάθεια ανάγνωσης αρχείου που πιθανόν δεν υπάρχει

```
import java.io.*;
```

```
public class ReadFile {
    public static void main(String[] args) {
        try {
            // Προσπάθεια ανάγνωσης αρχείου
            FileReader file = new FileReader("FileNotExists.txt");
            BufferedReader reader = new BufferedReader(file);
            System.out.println(reader.readLine());
            reader.close();
        } catch(IOException e) {
            // Διαχείριση σφάλματος αν δεν μπορούμε να διαβάσουμε το αρχείο
            System.out.println("Error reading file");
        }
    }
}
```

Παράδειγμα 2: Προσπάθεια εγγραφής σε πιθανόν μη έγκυρη διαδρομή καταλόγου

```
import java.io.*;
```

```
public class writeFile {
    public static void main(String[] args) {
        // Προσπάθεια εγγραφής σε πιθανόν μη έγκυρο κατάλογο
        try {
            FileWriter file = new FileWriter("c:\\NoAccess\\file.txt");
            BufferedWriter writer = new BufferedWriter(file);
            writer.write("New data");
            writer.close();
        } catch(IOException e) {
            // Διαχείριση αν υπάρξει σφάλμα εγγραφής
            System.out.println("Error writing file");
        }
    }
}
```

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.

```
import java.util.Scanner;

public class Hangman {
    public static void main(String[] args) {
        String[] words = {"printer", "disk", "monitor", "keyboard", "mouse"};
        // Επιλέγουμε μια τυχαία λέξη του πίνακα words
        int randomIndex = (int) (Math.random() * (words.length));
        String word = words[randomIndex];

        // Στην αρχή εμφανίζουμε _ για κάθε γράμμα της λέξης και στη
        // συνέχεια τα αντικαθιστούμε με τα σωστά γράμματα που δίνει ο παίκτης
        // κρατάμε τη λέξη που θα εμφανίζουμε σαν πίνακα χαρακτήρων
        char[] guess = "_".repeat(word.length()).toCharArray();

        int round = 8; // Αριθμός προσπαθειών
        Scanner scanner = new Scanner(System.in);
        System.out.println("H A N G M A N");

        while (round > 0) {
            System.out.println("word to find: " + new String(guess));
            System.out.println("You have " + round + " tries.");
            System.out.print("Enter letter: ");
            String input = scanner.next().toLowerCase();
            char letter = input.charAt(0); //Κρατάμε τον 1ο χαρακτήρα που δόθηκε
            boolean found = false;
            for (int i = 0; i < word.length(); i++) {
                if (word.charAt(i) == letter) {
                    guess[i] = letter;
                    found = true;
                }
            }
            if (!found) { // Αν το γράμμα δεν βρέθηκε μείωσε τις προσπάθειες
                round--;
            }
            if (word.equals(new String(guess))) {
                break; // Η λέξη βρέθηκε, έξοδος από το loop
            }
        }
        scanner.close();

        if (round > 0) {
            System.out.println("You win!");
        } else {
            System.out.println("Sorry, you lose!");
        }
        System.out.println("The word was: " + word);
    }
}
```

199. Να ορίσετε σε JAVA την κλάση Tires (Λάστιχα) με τις ακόλουθες ιδιότητες: width Πλάτος της επιφάνειας του ελαστικού (inches) radius Ακτίνα του ελαστικού (inches) rim Ζάντα (inches) type Τύπος ελαστικού 0 – Παντός Καιρού 1 – Ξηρό 2 – Βροχή 3 – Χιόνι brand Κατασκευαστής model Μοντέλο year έτος κατασκευής Να δημιουργήσετε constructor για την κλάση αυτή, ο οποίος να παίρνει παραμέτρους για όλα τα στοιχεία της κλάσης.

```
public class Tires {
    // width, radius, rim παρότι σε ίντσες σύμφωνα με την εκφώνηση
    // θα μπορούσαν να είναι int γιατί στα ελαστικά χρησιμοποιούμε
    // ακέραιες τιμές
    private double width, radius, rim;
    private int type, year;
    private String brand, model;

    Tires(double width, double radius, double rim, int type,
          String brand, String model, int year) {
        this.width = width;
        this.radius = radius;
        this.rim = rim;
        this.type = type;
        this.brand = brand;
        this.model = model;
        this.year = year;
    }
}
```

Στη συνέχεια μπορούμε να δημιουργήσουμε ένα αντικείμενο της κλάσης ως εξής:
 Tires tyre = new Tires(225, 55, 16, 1, "Michelin", "Primacy4", 2023);

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

Abstract class	Interface
Δεν υποστηρίζει πολλαπλή κληρονομικότητα	Υποστηρίζει πολλαπλή κληρονομικότητα
Υποστηρίζει Constructor	Δεν υποστηρίζει Constructor
Κανονικές μεταβλητές	Μόνο static final σταθερές
Κανονικές και αφηρημένες (abstract) μέθοδοι	Μόνο αφηρημένες (abstract) μέθοδοι
Προσδιοριστές πρόσβασης private, protected, public	Προσδιοριστές πρόσβασης μόνο public
Μπορεί να επεκτείνει άλλη Class ή Abstract Class	Μπορεί να επεκτείνει μόνο άλλο Interface

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

Τα interfaces μοιάζουν με πλήρως αφηρημένες κλάσεις (abstract class). Περιέχουν μόνο αφηρημένες μεθόδους, δηλαδή μεθόδους χωρίς σώμα. Χρησιμοποιούνται σαν πρότυπο και οι μέθοδοι που περιέχουν θα πρέπει να υλοποιηθούν από κάποια κλάση που θα υλοποιήσει το interface.

Δημιουργία interface:

```
interface Animal {
    void sound();
}
```

Στη συνέχεια το υλοποιούμε με κάποια κλάση:

```
class Pig implements Animal {
    public void sound() {
        System.out.println("Pig makes hog hog");
    }

    public static void main(String[] args) {
        Animal pig = new Pig();
        pig.sound();
    }
}
```

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

```
import java.util.Scanner;

class Temperature {
    public static void main(String[] args) {
        float f, c;
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter temperature in Fahrenheit: ");
        f = scanner.nextFloat();
        c = 5 * (f - 32) / 9;
        System.out.println("Temperature in Celsius: " + c);
        scanner.close();
    }
}
```

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.*;

class NameList {
    public static void main(String[] args) {
        final int n = 6;
        List<String> names = new ArrayList<>();
        Scanner scanner = new Scanner(System.in);

        for (int i = 0; i < n; i++) {
            System.out.print("Enter name: ");
            names.add(scanner.next());
        }
        scanner.close();

        System.out.println("Original order: " + names);
        Collections.sort(names);
        System.out.println("Sorted order: " + names);
    }
}
```

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

Το Method Overloading μας επιτρέπει να έχουμε δύο ή περισσότερες μεθόδους σε μια κλάση με το ίδιο όνομα. Οι μέθοδοι θα πρέπει να δέχονται διαφορετικό αριθμό ή τύπο ορισμάτων. Η Java ελέγχει τα ορίσματα όταν καλούμε τη συνάρτηση και εκτελεί την κατάλληλη.

```
class Overload {
    public static void main(String[] args) {
        showInfo("George");    // Τύπωσε το όνομα
        showInfo(30);          // Τύπωσε την ηλικία
    }

    // Δύο μέθοδοι showInfo (overloading)
    public static void showInfo(String name) {
        System.out.println("Your name is: " + name);
    }

    public static void showInfo(int age) {
        System.out.println("Your age is: " + age);
    }
}
```

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

Με το Method Overriding μια υποκλάση μπορεί να υλοποιήσει με διαφορετικό τρόπο μια μέθοδο που έχει οριστεί ήδη στην κλάση γονέα. Όταν στην κλάση παιδί ορίζεται μια μέθοδος που έχει το ίδιο όνομα, δέχεται τον ίδιο αριθμό και τύπο ορισμάτων και επιστρέφει τον ίδιο τύπο δεδομένων με μια μέθοδο της κλάσης γονέα τότε υπερισχύει η μέθοδος της κλάσης παιδί. Το ποια μέθοδος θα εκτελεστεί εξαρτάται από το αντικείμενο. Αν ανήκει στην κλάση γονέα θα εκτελεστεί η μέθοδος του γονέα. Αν ανήκει στην κλάση παιδί θα εκτελεστεί η μέθοδος της κλάσης παιδί.

```
public class Override {
    public static void main(String[] args) {
        Animal animal = new Animal();
        Animal pig = new Pig();
        animal.sound();    // Θα εμφανίσει Animals make sounds
        pig.sound();        // Θα εμφανίσει Pig makes hog hog
    }
}

class Animal {
    public void sound() {
        System.out.println("Animals make sounds");
    }
}

class Pig extends Animal {
    @Override
    public void sound() {
        System.out.println("Pig makes hog hog");
    }
}
```


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();}}
```

Το MyInterface δηλώνει 2 μεθόδους: method1(), method2();

Ένα interface είναι απλά ένα πρότυπο που καθορίζει ποιες μεθόδους πρέπει να εφαρμόσουν οι κλάσεις που το υλοποιούν. Οι μέθοδοι στο interface δεν έχουν σώμα. Το σώμα δημιουργείται και υλοποιείται στην κλάση.

Έτσι η κλάση XYZ που δηλώνεται στη συνέχεια υλοποιεί τις δύο μεθόδους του MyInterface.

Μετά δημιουργείται το αντικείμενο obj που ανήκει στην κλάση XYZ και καλείται η method1 με την obj.method1().

Επομένως θα τυπωθεί στην οθόνη "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.

```
import java.util.Vector;  
  
public class Inventory {  
    private Vector<String[]> items = new Vector<String[]>();  
  
    public void addItem(String name, String category) {  
        items.add(new String[]{name, category});  
    }  
  
    public void showItems() {  
        System.out.println("Inventory items\tCategory");  
        for (int i = 0; i < items.size(); i++) {  
            System.out.println(items.get(i)[0] + "\t" + items.get(i)[1]);  
        }  
    }  
  
    public static void main(String[] args) {  
        Inventory inventory = new Inventory();  
        inventory.addItem("Mithril Sword", "Attack");  
        inventory.addItem("Plate Armour", "Defense");  
        inventory.showItems();  
    }  
}
```

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.

```
public class Main {
    public static void main(String[] args) {
        int x = 10;
        int y = 20;
        Calculator sum = new Sum();
        Calculator sub = new Sub();
        Calculator multiply = new Multiply();
        sum.print(x, y);          // Τυπώνει Result: 30
        sub.print(x, y);          // Τυπώνει Result -10
        multiply.print(x, y);      // Τυπώνει Result 200
    }
}

// Η υπερκλάση Calculator τυπώνει τα αποτελέσματα των υποκλάσεων
// Sum, Sub, Multiply οι οποίες υλοποιούν την αφαιρετική μέθοδο getResult()
abstract class Calculator {
    public void print(int a, int b) {
        System.out.println("Result: " + getResult(a, b));
    }

    abstract protected int getResult(int a, int b);
}

class Sum extends Calculator {
    @Override
    protected int getResult(int x, int y) {
        return x + y;
    }
}

class Sub extends Calculator {
    @Override
    protected int getResult(int x, int y) {
        return x - y;
    }
}

class Multiply extends Calculator {
    @Override
    protected int getResult(int x, int y) {
        return x * 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 = $Width \cdot Width$).

```
public class Area {
    public static void main(String[] args) {
        double x = 10, y = 20;
        Rectangle rectangle = new Rectangle(x, y);
        Square square = new Square(x);
        Circle circle = new Circle(x);
        rectangle.printArea(); // Τυπώνει Rectangle area is: 200.0
        square.printArea();    // Τυπώνει Square area is: 100.0
        circle.printArea();    // Τυπώνει Circle area is: 628.0
    }
}

class Rectangle {
    protected double width, height;
    protected String shapeName;

    public Rectangle(double width, double height) {
        this.width = width;
        this.height = height;
        shapeName = "Rectangle";
    }

    public double getArea() {
        return width * height;
    }
    public void printArea() {
        System.out.println(shapeName + " area is: " + getArea());
    }
}

class Square extends Rectangle {
    public Square(double width) {
        super(width, width);
        shapeName = "Square";
    }
}

class Circle extends Rectangle {
    private double radius;
    private final double pi = 3.14;
    public Circle(double radius) {
        super(0, 0); // Δεν χρειαζόμαστε width, height στον κύκλο
        this.radius = radius;
        shapeName = "Circle";
    }
    @Override
    public double getArea() {
        return 2 * pi * radius * radius;
    }
}
```

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

Το επίπεδο είναι ένα αντικείμενο δύο διαστάσεων που έχει μόνο μήκος και πλάτος. Το πάχος του είναι αμελητέο. Το επίπεδο μπορεί να έχει πολλά σχήματα (τρίγωνο, τετράγωνο, κύκλος, πολύγωνο). Στην OpenGL σχεδιάζουμε ένα επίπεδο δηλώνοντας τις κορυφές του με συντεταγμένες X, Y. Η αρχή των αξόνων είναι το σημείο 0, 0. Το X παίρνει αρνητικές τιμές αριστερά του και θετικές δεξιά του. Το Y είναι θετικό προς τα πάνω και αρνητικό προς τα κάτω. Για το Z η τιμή είναι αρνητική όταν κατευθύνεται προς το εσωτερικό της οθόνης και θετική όταν κατευθύνεται προς τον θεατή.

Y \ X	-1	0	1
1	-1, 1	0, 1	1, 1
0	-1, 0	0, 0	1, 0
-1	-1, -1	0, -1	1, -1

Η παρακάτω συνάρτηση σχεδιάζει ένα ορθογώνιο επίπεδο στην OpenGL:

```
void drawPlane()
{
    glBegin(GL_QUADS);
    glVertex3f(-1.0, 1.0, 0.0);
    glVertex3f(-1.0, -1.0, 0.0);
    glVertex3f( 1.0, -1.0, 0.0);
    glVertex3f( 1.0, 1.0, 0.0);
    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 χρησιμοποιεί normalized values για την απόδοση των χρωμάτων, δηλαδή τιμές που δεν εξαρτώνται από το υλικό. Η ελάχιστη τιμή για κάθε χρώμα είναι 0 και η μέγιστη 1.

Για παράδειγμα στο χρωματικό μοντέλο RGB το κόκκινο χρώμα δηλώνεται ως RGB(1f, 0f, 0f). Αν το υλικό υποστηρίζει 8bit ανά κανάλι τότε η τιμή 1 αντιστοιχεί σε 256 (2^8). Επειδή έχουμε 3 κανάλια χρώματος (Red, Green, Blue) μπορούμε να απεικονίσουμε 256^3 δηλαδή 16.7 εκατομμύρια χρώματα. Αν όμως το υλικό υποστηρίζει 10bit ανά κανάλι τότε το 1f αντιστοιχεί σε 1024 (2^{10}) και μπορούμε να έχουμε 1024^3 ή περίπου 1 δις χρώματα.

212. Η OpenGL χρησιμοποιεί μια απλή, βασική, μορφή ονοματολογίας για τις εντολές της. Αναφέρατε ποια είναι αυτή και παραθέστε παράδειγμα.

Τα δύο πρώτα γράμματα της εντολής ορίζουν τη βιβλιοθήκη (gl). Ακολουθεί η εντολή με κεφαλαίο το 1^ο γράμμα κάθε λέξης. Τέλος δηλώνονται οι παράμετροι που δέχεται και ο τύπος παραμέτρων.

Παράδειγμα glVertex3f().

- gl: Όνομα βιβλιοθήκης.
- Vertex: Όνομα εντολής.
- 3: Δέχεται 3 παραμέτρους.
- f: Οι παράμετροι είναι τύπου float.

213. Αναφέρατε τι ονομάζουμε GLUT στην OpenGL. Για ποιο λόγο δημιουργήθηκε και ποιες βασικές λειτουργίες προσφέρει;

Το GLUT είναι μια βιβλιοθήκη της OpenGL που δημιουργήθηκε για να διαχειρίζεται το χειρισμό γεγονότων, την είσοδο δεδομένων και την απεικόνιση παραθύρων, μενού κλπ. Είναι ανεξάρτητη από λειτουργικό σύστημα (Linux, Mac OS, Windows) κι έτσι ο προγραμματιστής δεν χρειάζεται να γράφει ξανά τον κώδικα για κάθε λειτουργικό. Οι εντολές της βιβλιοθήκης ξεκινούν με το πρόθεμα `glut`.

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);

Με την εντολή `glOrtho`. Η γενική σύνταξή είναι `glOrtho(αριστερά, δεξιά, κάτω, πάνω, κοντά, μακριά)`, επομένως:
`glOrtho(-3f, 3f, -3f, 3f, 1f, 100f);`

215. Εξετάστε την ισχύ των παρακάτω προτάσεων δηλώνοντας ΣΩΣΤΟ ή ΛΑΘΟΣ και εξηγώντας την επιλογή σας (Όπου κρίνετε απαραίτητο, παραθέστε παράδειγμα). α) 1. Τα επικρατέστερα Λειτουργικά Συστήματα (Windows, Unix, Linux, Mac OS) υποστηρίζουν την OpenGL. β) Η OpenGL μπορεί αποκλειστικά να κληθεί (is callable) από τις γλώσσες προγραμματισμού C / C++ (δηλαδή υπάρχει μοναδικό language binding). γ) Το μοναδικό περιβάλλον ανάπτυξης προγραμμάτων OpenGL είναι το DEV C++. δ) Η OpenGL περιέχει εντολές επιλογής (τύπου If ... else). ε) Οι εντολές της OpenGL ξεκινούν με το πρόθεμα `gl`.

α) ΣΩΣΤΟ: Η OpenGL είναι μια ανοιχτή πλατφόρμα-πρότυπο γραφικών που υποστηρίζεται από τα δημοφιλέστερα λειτουργικά.

β) ΛΑΘΟΣ: Η OpenGL σχεδιάστηκε για C/C++ αλλά έχει bindings και άλλες γλώσσες προγραμματισμού μέσω βιβλιοθηκών όπως Python, Java, C# κλπ.

γ) ΛΑΘΟΣ: Προγράμματα OpenGL μπορούν να αναπτυχθούν σε οποιοδήποτε περιβάλλον προγραμματισμού C/C++ ή γλωσσών που έχουν bindings για OpenGL. Για παράδειγμα Visual Studio, Visual Studio Code, Eclipse κλπ.

δ) ΛΑΘΟΣ: Η OpenGL δεν είναι γλώσσα προγραμματισμού αλλά ανοικτό περιβάλλον διαχείρισης γραφικών.

ε) ΣΩΣΤΟ: Όλες οι εντολές ξεκινούν με `gl`. Παράδειγμα: `glVertex3f`.

216. Ποια η λειτουργία των παρακάτω εντολών στην OpenGL και τι παραμέτρους δέχονται; Να δοθεί παράδειγμα. α) `glutInitWindowPosition` β) `glutInitWindowSize` γ) `glutCreateWindow`

Αυτές οι εντολές αρχικοποιούν και εμφανίζουν ένα παράθυρο στην οθόνη.

α) `glutInitWindowPosition(int x, int y)`: Καθορίζει τις συντεταγμένες X, Y που θα τοποθετηθεί η επάνω αριστερή γωνία του παραθύρου. Η αρχή των συντεταγμένων είναι η επάνω αριστερή γωνία της οθόνης.

// Η επάνω αριστερή γωνία του παραθύρου θα είναι σε θέση `x=100, y=50`
`glutInitWindowPosition(100, 50);`

β) `glutInitWindowSize(int width, int height)`: Το μήκος και πλάτος του παραθύρου σε pixels.
// Το παράθυρο θα έχει πλάτος 500 και ύψος 400 pixels

```
glutInitWindowSize(500, 400);
```

γ) `glutCreateWindow(char *name)`: Δημιουργεί το παράθυρο με όνομα `name` στη μπάρα του.
// θα δημιουργήσει παράθυρο με το όνομα "Hello"
`glutCreateWindow("Hello");`

217. Περιγράψτε τρεις βασικές βιβλιοθήκες εντολών της OpenGL που περιέχουν εντολές σχεδίασης, γραφικών αι απόδοσης. Σε ποιο σημείο του προγράμματος δηλώνονται;

Οι βιβλιοθήκες πρέπει να δηλώνονται στην αρχή του προγράμματος πριν τη `main` ώστε να είναι διαθέσιμες σε όλο το πρόγραμμα. Για να τις δηλώσουμε χρησιμοποιούμε την εντολή `#include`:

```
#include <GL/gl.h>      // Core Library
#include <GL/glu.h>     // Utility Library
#include <GL/glut.h>    // Utility Toolbox Library
```

α) Core Library: Περιέχει τις βασικές εντολές σχεδιασμού όπως `glBegin()`, `glEnd()` για την αρχή και λήξη σχεδιασμού ενός πρωτογενούς σχήματος, `glVertex3f()` για τον καθορισμό των συντεταγμένων ενός σημείου κλπ. Οι εντολές ξεκινούν με το πρόθεμα `gl`.

β) Utility Library: Είναι μια βοηθητική βιβλιοθήκη που δημιουργεί σύνθετες δομές και εργαλεία, όπως για παράδειγμα `gluSphere` (δημιουργεί μια σφαίρα). Οι εντολές ξεκινούν με το πρόθεμα `glu`.

γ) Utility Toolbox Library: Περιέχει εντολές για απόκριση σε γεγονότα, εντολές εισόδου δεδομένων και δημιουργία παραθύρων, μενού κλπ. Για παράδειγμα η `glutCreateWindow()` δημιουργεί ένα παράθυρο. Οι εντολές ξεκινούν με το πρόθεμα `glut`.

218. Αναφέρετε την λειτουργία της `glutMainLoop()` στην OpenGL. Σε ποιο σημείο πρέπει να καλείται;

Η `glutMainLoop()` καλείται στο τέλος του κύριου προγράμματος αφού έχουν πρώτα δηλωθεί όλα τα event handlers και έχει δημιουργηθεί το παράθυρο της εφαρμογής.

Η `glutMainLoop()` ελέγχει επ' άπειρο και ανταποκρίνεται σε γεγονότα που δημιουργούνται όπως κίνηση ποντικιού, πατήματα πλήκτρων, αλλαγή σκηνικού κλπ.

219. Ποια είναι τα πλεονεκτήματα χρήσης μιας πλατφόρμας CMS (Content Management System), σε σχέση με την δημιουργία εκ νέου ενός web site;

Γρήγορη ανάπτυξη: Παρέχει εργαλεία και έτοιμα θέματα εμφάνισης. Έτσι ένας ιστότοπος μπορεί να δημιουργηθεί γρήγορα έχοντας ταυτόχρονα επαγγελματική εμφάνιση.

Ευκολία διαχείρισης: Η διαχείριση και ενημέρωση του ιστότοπου μπορεί να γίνει από άτομα που δεν είναι προγραμματιστές χρησιμοποιώντας απλά ένα browser.

Ασφάλεια: Δεν μπορεί ένας χρήστης να δει ή να τροποποιήσει περιεχόμενο που δεν έχει δικαίωμα πρόσβασης. Υπάρχουν έτοιμα plugins για προστασία από κακόβουλες επιθέσεις.

Επεκτασιμότητα: Υπάρχουν πολλά έτοιμα modules που μπορεί να επεκτείνουν τη λειτουργικότητα, όπως για παράδειγμα υποστήριξη πληρωμής με πιστωτικές κάρτες, αποστολή ομαδικών e-mails κλπ.

Φιλικότητα προς SEO: Οι ιστοσελίδες ακολουθούν βέλτιστες πρακτικές SEO και υπάρχουν modules που προσφέρουν βελτίωση της θέσης κατάταξης στις μηχανές αναζήτησης.

Χαμηλό κόστος: Έχουν χαμηλό κόστος ανάπτυξης και συντήρησης.

220. Το σύστημα διαχείρισης περιεχομένου (CMS) υποστηρίζει την εγκατάσταση διαφόρων modules. Τι είναι τα modules και γιατί είναι χρήσιμα για ένα CMS;

Τα modules είναι έτοιμα κομμάτια κώδικα που προσφέρουν επιπλέον λειτουργικότητα και επεκτασιμότητα σε ένα CMS. Για παράδειγμα ένα module μπορεί να προσφέρει εύκολη δημιουργία και διαχείριση φορμών, κάποιο άλλο προστασία από κακόβουλες επιθέσεις κλπ.

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;

Το alt tag περιγράφει το περιεχόμενο μιας εικόνας όταν δεν είναι δυνατή η εμφάνιση της στο φυλλομετρητή. Επίσης χρησιμοποιείται από τους screen readers για άτομα με προβλήματα όρασης. Το title tag όταν υπάρχει στο head της ιστοσελίδας δηλώνει τον τίτλο που θα εμφανίζεται για τη σελίδα μας στην καρτέλα του φυλλομετρητή.

Και τα δύο tags δεν είναι υποχρεωτικά για να εμφανιστεί η ιστοσελίδα. Συνιστάται όμως έντονα η χρήση τους για να είναι η σελίδα φιλική προς SEO. Οι μηχανές αναζήτησης δίνουν μεγάλη σημασία στο title καθώς περιγράφει το περιεχόμενο της σελίδας αλλά και στο alt γιατί μπορούν να καταλάβουν το περιεχόμενο των φωτογραφιών. Όσο πιο φιλική προς SEO είναι μια ιστοσελίδα τόσο μεγαλύτερες είναι οι πιθανότητες να καταλάβει υψηλή θέση κατάταξης στα αποτελέσματα αναζήτησης.

Σημείωση: Το title attribute μπορεί να χρησιμοποιηθεί επίσης σε οποιοδήποτε στοιχείο μιας HTML σελίδας για να εμφανίσει ένα tooltip με επεξηγηματικό κείμενο όταν ο χρήστης περνάει το ποντίκι πάνω από το στοιχείο.

222. Η επιστήμη του SEO (Search 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: Είναι ο τίτλος της σελίδας που εμφανίζεται στην καρτέλα του φυλλομετρητή και ταυτόχρονα προΐδεάζει για το περιεχόμενο της σελίδας.

Description tag: Είναι μια σύντομη περιγραφή των περιεχομένων της σελίδας, πιο αναλυτική από το title.

Και τα δύο αυτά tags είναι πολύ σημαντικά για το SEO καθώς η μηχανή αναζήτησης καταλαβαίνει το περιεχόμενο της σελίδας και τι αφορά. Είναι σημαντικό να συμπεριλαμβάνουμε στο title και description λέξεις κλειδιά που πιθανόν αναζητούν οι χρήστες. Στα αποτελέσματα αναζήτησης το Google εμφανίζει τον τίτλο και την περιγραφή της σελίδας ώστε να καταλαβαίνει εύκολα ο χρήστης αν η σελίδα είναι σχετική με την αναζήτηση του.

Keyword tag: Σ' αυτό τοποθετούμε λέξεις που πιθανόν χρησιμοποιούν οι χρήστες για να αναζητήσουν σελίδες με περιεχόμενο ανάλογο της δικής μας. Δεν χρησιμοποιείται πλέον από τις μηχανές αναζήτησης.

223. Μετά την υλοποίηση της ιστοσελίδας με χρήση CMS τοπικά, ποια βήματα πρέπει να γίνουν ώστε να "ανέβει" (upload) στο Internet;

Κατοχύρωση domain name: Πρέπει να διαλέξουμε ένα όνομα για τον ιστότοπο, πχ mysite.gr.

Web Hosting: Θα πρέπει να νοικιάσουμε χώρο σε κάποιο πάροχο που υποστηρίζει την εγκατάσταση του CMS που χρησιμοποιήσαμε.

Εγκατάσταση CMS: Θα πρέπει να εγκαταστήσουμε το CMS στο χώρο που μας διέθεσε ο provider.

Μεταφορά αρχείων: Εξάγουμε την τοπική βάση δεδομένων και τα αρχεία των ιστοσελίδων και τα ανεβάζουμε στον κατάλληλο φάκελο του web server που νοικιάσαμε. Το ανέβασμα γίνεται μέσα από το control panel που μας διαθέτει ο service provider.

Αλλαγή DNS records: Τροποποιούμε τις εγγραφές DNS ώστε να δείχνουν στο νέο server που δημιουργήσαμε.

224. Τι είναι τα σχεδιαστικά χνάρια και ποια η χρησιμότητά τους;

Τα σχεδιαστικά χνάρια (design patterns) αποτυπώνουν τον πυρήνα της λύσης ενός προβλήματος ώστε να είναι δυνατή η λύση άπειρες φορές χωρίς να επαναλαμβάνεται ο ίδιος τρόπος δύο φορές. Στο σχεδιασμό ιστοσελίδων τα σχεδιαστικά χνάρια χρησιμοποιούνται κατά το αρχικό στάδιο υλοποίησης για να περιγράψουν τη δομή του ιστότοπου, τις εξαρτήσεις μεταξύ των σελίδων, τα μενού, τη λειτουργικότητα κλπ.

ΟΜΑΔΑ Β. ΕΙΔΙΚΕΣ ΕΡΩΤΗΣΕΙΣ

1. Περιγράψτε τον τρόπο λειτουργίας των οπτικών δίσκων (CD-DVD/ROM).

Οι οπτικοί δίσκοι έχουν μια ανακλαστική επιφάνεια. Τα σημεία της επιφάνειας που ανακλούν το φως αναπαριστούν το ψηφιακό 1 ενώ όσα δεν το ανακλούν το ψηφιακό 0.

Η μονάδα CD/DVD στέλνει μια ακτίνα λέιζερ προς την επιφάνεια του δίσκου και από την ανάκλαση της ακτίνας καταλαβαίνει ποια bit είναι 0 ή 1. Για να γράψει στοιχεία χρησιμοποιεί μια ισχυρότερη ακτίνα laser που μπορεί να κάψει σημεία της επιφάνειας ώστε να μην ανακλούν το φως.

2. Τι ονομάζουμε κύρια και τι δευτερεύουσα μνήμη; Ποια είναι η βασική διαφορά τους;

Η κύρια μνήμη είναι η μνήμη RAM του υπολογιστή. Η προσπέλασή της είναι πολύ γρήγορη αλλά τα περιεχόμενα της χάνονται όταν κλείσει ο υπολογιστής. Στη μνήμη RAM φορτώνεται το λειτουργικό σύστημα και οι εφαρμογές για να εκτελεστούν. Η RAM έχει άμεση επικοινωνία με την ΚΜΕ.

Η δευτερεύουσα μνήμη είναι η μονάδα αποθήκευσης. Είναι πιο αργή από τη RAM αλλά τα περιεχόμενα παραμένουν και μετά το κλείσιμο του υπολογιστή. Χρησιμοποιείται για τη μόνιμη αποθήκευση εφαρμογών και δεδομένων. Δευτερεύουσα μνήμη οι σκληροί δίσκοι, δίσκοι SSD, flash disks, CD/DVD κλπ.

3. Πόσους αγωγούς χρησιμοποιεί ο δίαυλος USB και ποια η χρήση του καθενός απ' αυτούς;

Χρησιμοποιεί 4 αγωγούς.

- VCC: Παρέχει τάση 5V.
- GND: Γείωση, μηδενική τάση αναφοράς 0V.
- D+: Μεταφορά δεδομένων.
- D- : Μεταφορά δεδομένων.

Επιπλέον τα μεταλλικά βύσματα χρησιμοποιούνται ως θωράκιση για ηλεκτρικές παρεμβολές.

4. Τι είναι το υλικό (Hardware) και τι το λογισμικό (Software) στοιχείο ενός υπολογιστικού συστήματος; Να αναφέρετε τέσσερα παραδείγματα από το καθένα.

Το hardware είναι τα φυσικά εξαρτήματα που απαρτίζουν έναν υπολογιστή, δηλαδή η κεντρική μονάδα, οι μονάδες εισόδου/εξόδου και τα αποθηκευτικά μέσα. Παραδείγματα: η μητρική πλακέτα, ο επεξεργαστής, η οθόνη, το πληκτρολόγιο, ο σκληρός δίσκος.

Το software είναι όλα τα προγράμματα που περιέχουν εντολές και υπαγορεύουν στην ΚΜΕ τον τρόπο που θα χειριστεί τα δεδομένα για να παράξει πληροφορίες. Παραδείγματα: το λειτουργικό σύστημα, επεξεργαστές κειμένου, σχεδιαστικά προγράμματα, παιχνίδια.

5. Ποιες είναι οι βασικές κατηγορίες των λειτουργικών συστημάτων; Ποια είναι τα ιδιαίτερα χαρακτηριστικά καθενός;

Θα μπορούσαμε να χαρακτηρίσουμε τα λειτουργικά συστήματα ανάλογα:

- Με το αριθμό ταυτόχρονων χρηστών που υποστηρίζουν.
- Με την άδεια χρήσης λογισμικού.
- Με τον τρόπο επεξεργασίας δεδομένων που υποστηρίζουν.

Κατάταξη με πλήθος χρηστών

Διακρίνονται σε δύο κατηγορίες:

- Ενός χρήστη: Μόνο ένας χρήστης μπορεί να εξυπηρετηθεί σε κάθε χρονική στιγμή. Παράδειγμα τέτοιου λειτουργικού είναι το MS-DOS.
- Πολλών χρηστών: Το λειτουργικό σύστημα μπορεί να εξυπηρετήσει ταυτόχρονα πολλούς χρήστες. Παράδειγμα: Windows Server, Linux.

Κατάταξη με την άδεια χρήσης λογισμικού

Διακρίνονται σε δύο κατηγορίες:

- Ανοικτού κώδικα: Τα λειτουργικά αυτά αναπτύσσονται από πολλούς προγραμματιστές που εργάζονται εθελοντικά. Η χρήση τους είναι δωρεάν και κάθε χρήστης έχει πρόσβαση στον πηγαίο κώδικα του λειτουργικού. Τέτοιο λειτουργικό είναι για παράδειγμα το Linux.
- Κλειστού κώδικα: Αυτά τα λειτουργικά αναπτύσσονται από κάποια εταιρία λογισμικού και απαιτείται πληρωμή για τη χρήση τους. Είναι κλειστού κώδικα γιατί ο χρήστης δεν έχει πρόσβαση στον πηγαίο κώδικα. Παράδειγμα τέτοιου λειτουργικού είναι τα Windows.

Κατάταξη κατά τρόπο επεξεργασίας δεδομένων

Διακρίνονται στις εξής κατηγορίες:

- Κατά δέσμες (batch): Συναντάται σε παλαιότερα λειτουργικά. Τα προγράμματα των χρηστών επεξεργάζονται όταν βρεθεί χρόνος και συνήθως αυτή η διαχείριση γίνεται κεντρικά.
- Μερισμού χρόνου (time sharing): Στη μνήμη του υπολογιστή βρίσκονται πολλά προγράμματα. Το λειτουργικό αναλαμβάνει να μοιράσει το χρόνο της ΚΜΕ σε όσα προγράμματα είναι έτοιμα προς εκτέλεση αξιοποιώντας έτσι πλήρως την επεξεργαστική ισχύ της ΚΜΕ.
- Πραγματικού χρόνου (real time): Το λειτουργικό εκτελεί τις διαδικασίες σε πραγματικό χρόνο χωρίς διακοπή από άλλες διεργασίες. Χρησιμοποιούνται σε κρίσιμες εφαρμογές όπως τραπεζικά συστήματα, έλεγχο διαστημοπλοίων κλπ.

6. Ποιους τύπους οθονών υπολογιστών γνωρίζετε; Περιγράψτε σύντομα τις αρχές λειτουργίας τους.

Καθοδικού σωλήνα (CRT)

Δημιουργούνται δέσμες ηλεκτρονίων από ηλεκτρονικούς εκτοξευτήρες (πυροβόλα). Υπάρχει ένα πυροβόλο στις μονόχρωμες οθόνες και τρία στις έγχρωμες. Ο καθοδικός σωλήνας περιβάλλεται από πηνία που δημιουργούν μαγνητικά πεδία κατευθύνοντας έτσι τις δέσμες στα κατάλληλα σημεία της οθόνης. Κάθε σημείο της οθόνης έχει ειδική επίστρωση που διεγείρεται από τα ηλεκτρόνια και εκπέμπει το κατάλληλο φως. Όλα τα σημεία της οθόνης σαρώνονται πολλές φορές το δευτερόλεπτο κι έτσι δημιουργείται η εικόνα.

Υγρών κρυστάλλων (LCD)

Αποτελείται από ένα πλέγμα κυττάρων υγρών κρυστάλλων. Κάθε κύτταρο ελέγχεται από τρανζίστορ. Ανάλογα με την ένταση του ρεύματος οι κρύσταλλοι περιστρέφονται αφήνοντας να περάσει πολύ, λίγο ή καθόλου φως. Το πλέγμα περιβάλλεται από φίλτρα οριζόντιας και κάθετης πόλωσης. Το πλέγμα υγρών κρυστάλλων δεν παράγει φως απλά επιτρέπει ή όχι τη διέλευσή του. Για αυτό

υπάρχουν από πίσω λάμπες φθορισμού που εκπέμπουν φως. Το φως πολώνεται από τα φίλτρα περνάει μέσα από το πλέγμα και δημιουργεί τα ίχνη (pixels) στην οθόνη.

Φωτοδιόδων (LED)

Είναι μια εξέλιξη των LCD. Οι λάμπες φθορισμού αντικαθίστανται με φωτοδιόδους LED. Τα LED παράγουν πιο έντονο φως και έχουν χαμηλότερη κατανάλωση. Ο τρόπος απεικόνισης παραμένει ίδιος.

7. Με ποιο τρόπο το Λειτουργικό Σύστημα συνεργάζεται με την Κεντρική Μονάδα Επεξεργασίας;

Η ΚΜΕ μπορεί να εκτελέσει ορισμένο αριθμό εντολών ανά μονάδα χρόνου (clock cycle). Ο πυρήνας ενός λειτουργικού συστήματος (kernel) φροντίζει να αξιοποιείται στο έπακρο αυτός ο χρόνος και να μην υπάρχουν κύκλοι ρολογιού χωρίς εντολές προς εκτέλεση. Έτσι προωθεί στην ΚΜΕ διεργασίες από πολλά διαφορετικά προγράμματα που είναι έτοιμα να στείλουν δεδομένα. Με τον τρόπο αυτό αξιοποιείται κατά το δυνατό περισσότερο η επεξεργαστική ισχύς της ΚΜΕ και ταυτόχρονα οι εφαρμογές φαίνεται ότι λειτουργούν παράλληλα χωρίς να διακόπτει η μία την εκτέλεση της άλλων.

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;

Στο Microsoft Word επιλέγουμε την εικόνα, κάνουμε δεξί κλικ και επιλέγουμε:

Αναδίπλωση κειμένου -> Πίσω από το κείμενο

Τώρα το κείμενο που θα γράψουμε θα εμφανίζεται μπροστά από την εικόνα.

9. Να περιγράψετε τον κύκλο ανάπτυξης ενός Πληροφοριακού Συστήματος. Δώστε ένα παράδειγμα.

Διερευνητική μελέτη: Προσδιορίζει το πρόβλημα και καταγράφει όλες τις πιθανές λύσεις του.

Μελέτη σκοπιμότητας: Επιλέγεται η βέλτιστη λύση με βάση το κόστος, το χρόνο υλοποίησης, την τεχνολογία, την τεχνογνωσία κλπ.

Ανάλυση απαιτήσεων: Καθορίζονται οι λειτουργίες του συστήματος, οι επιδόσεις του (πχ ο αριθμός χρηστών που θα εξυπηρετείται), τα δεδομένα εισόδου κλπ.

Σχεδιασμός: Προσδιορίζει τα μέρη του συστήματος και πως θα συνεργάζονται μεταξύ τους, τις οντότητες και τις σχέσεις τους, την ασφάλεια κλπ.

Υλοποίηση: Επιλέγονται τα κατάλληλα εργαλεία, βάση δεδομένων και γλώσσα προγραμματισμού για την υλοποίηση του σχεδιασμού.

Έλεγχος: Ελέγχεται το σύστημα για λάθη και παραλείψεις.

Εγκατάσταση: Εγκαθίσταται το σύστημα στον πελάτη και εκπαιδεύονται οι χρήστες στη λειτουργία του.

Συντήρηση: Με την πάροδο του χρόνου γίνονται αλλαγές για να συνάδει με τις τεχνολογικές απαιτήσεις ή την προσθήκη επιπλέον λειτουργιών.

Παράδειγμα: Διαχείριση εστιατορίου.

Στη διερευνητική μελέτη εξετάζουμε ποιες είναι οι λύσεις που ικανοποιούν τη λειτουργικότητα που επιθυμεί ο πελάτης.

Στη μελέτη σκοπιμότητας αποφασίζουμε ότι δεν υπάρχει έτοιμη λύση και πρέπει να σχεδιαστεί από την αρχή.

Αναλύουμε τις απαιτήσεις. Ο πελάτης εξυπηρετεί online αλλά και με φυσική παρουσία. Χρειάζεται GPS στους διανομείς για να υπολογίζει χρόνο παράδοσης κλπ.

Σχεδιάζουμε το σύστημα και πως θα αλληλεπιδρούν τα τμήματα μεταξύ τους.

Η υλοποίηση γίνεται με βάση δεδομένων SQL, PHP και JavaScript.

Ελέγχουμε την καλή λειτουργία και εξαλείφουμε τυχόν σφάλματα.

Γίνεται η εγκατάσταση στον πελάτη και εκπαιδεύουμε το προσωπικό.

Συντηρούμε το πρόγραμμα όταν χρειάζεται. Για παράδειγμα μετά από μήνες πρέπει να αλλάξουμε το πρωτόκολλο επικοινωνίας με την τράπεζα για πληρωμή μέσω πιστωτικών καρτών.

10. Με τη χρήση κατάλληλου προγράμματος να παρουσιάσετε την διαφορά μεταξύ των εντολών break και continue.

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

int main()
{
    for (int i = 0; i < 10; i++)
    {
        if (i == 2) continue;
        if (i == 5) break;
        cout << i << " ";
    }
    return 0;
}
```

Το παραπάνω πρόγραμμα θα εμφανίσει 0 1 3 4.

Η εντολή continue ξεκινάει την επόμενη επανάληψη όταν i=2 άρα δεν θα τυπωθεί το 2.

Η εντολή break θα διακόψει τον βρόγχο όταν i=5 άρα δεν θα τυπωθεί το 5 και οι υπόλοιποι αριθμοί.

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 <stdlib.h>
#include <stdio.h>
int main()
{
    int num = 0, sum = 0, count = 0;
    double avg;
    while(1)
    {
        printf("Enter number (0 to exit): ");
        scanf("%d", &num);
        if (num == 0) break;
        sum += num;
        count++;
        avg = (double)sum / count;
        printf("Average value is: %.2lf\n", avg);
    }
    return 0;
}
```

12. Να γραφεί πρόγραμμα σε C, που θα διαβάζει 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 num1, num2, sum;
    printf("Enter number 1: ");
    scanf("%d", &num1);
    printf("Enter number 2: ");
    scanf("%d", &num2);
    sum = num1 + num2;

    printf("Decimal sum: %d, Hex sum: %X, Octal sum: %o\n", sum, sum, 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 getAverage(double a, double b)
{
    return (a + b) / 2;
}

int main()
{
    double num1, num2;
    printf("Calculate mean value of 2 real numbers\n");
    printf("Enter number 1: ");
    scanf("%lf", &num1);
    printf("Enter number 2: ");
    scanf("%lf", &num2);
    printf("Mean value is: %.2lf", getAverage(num1, num2));
    return 0;
}
```

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 name[50];
```

```
printf("Enter your name: ");
scanf("%s", name);
printf("Hello %s\n", name);
}
```

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()
{
    printf("Fahrenheit\tCelsius\n");
    for (int f = -100; f <= 300; f+=20)
    {
        double c = 5 * (f - 32) / 9.0;
        printf("%d\t\t%.2lf\n", f, c);
    }
    return 0;
}
```

16. Να σχεδιαστεί ο αλγόριθμος εξαγωγής στοιχείου από λίστα.

Εξαγωγή στοιχείου από μονή συνδεδεμένη λίστα:

```
! L          Λίστα
! L.data     Τιμή στοιχείου λίστας
! L.next     Δείκτης προς το επόμενο στοιχείο λίστας
! x          Τιμή στοιχείου για εξαγωγή
Αλγόριθμος deleteNode
Δεδομένα    // L //
Διάβασε x
previous <- NULL
current <- L.αρχή
Όσο current <> NULL Επανάλαβε
    Αν L.data = x τότε
        Αν previous = NULL τότε
            ! Το στοιχείο είναι στην αρχή της λίστας
            ! Αρχή της λίστας θα γίνει ο επόμενος κόμβος
            L.αρχή <- current.next
        Αλλιώς
            ! Παράκαμψη του κόμβου που περιέχει το x
            previous.next <- current.next
        Τέλος_Αν
        current <- NULL ! Αποδέσμευση από τη μνήμη
        Εκτύπωσε "Το στοιχείο διαγράφηκε"
        Τέλος deleteNode ! Έξοδος
    Τέλος_Αν
    previous <- current
    current <- current.next
Τέλος_Επανάληψης
Εκτύπωσε "Το στοιχείο δεν βρέθηκε"
Τέλος deleteNode
```


17. Να σχεδιαστεί ο αλγόριθμος εισαγωγής στοιχείου σε λίστα.

Εισαγωγή στοιχείου στο τέλος μονής συνδεδεμένης λίστας:

```
! L          Λίστα
! L.data     Τιμή στοιχείου λίστας
! L.next     Δείκτης προς τον επόμενο κόμβο
! x          Τιμή του στοιχείου που θα προστεθεί
Αλγόριθμος addNode
Δεδομένα // L //
Διάβασε x
! Δημιουργία κόμβου με τιμή x και όνομα newNode
newNode.data <- x
newNode.next <- NULL

current <- L.αρχή
! Αν η λίστα είναι άδεια, αρχή της λίστας θα γίνει ο νέος κόμβος
Αν current = NULL τότε
    L.αρχή <- newNode
Αλλιώς
    Όσο current <> NULL επανάλαβε
        Αν current.next = NULL τότε ! Εισαγωγή στο τέλος
            current.next <- newNode
        Τέλος_Αν
        current <- current.next
    Τέλος_επανάληψης
Τέλος_Αν
Εκτύπωσε "Το στοιχείο προστέθηκε"
Τέλος addNode
```

18. Να σχεδιαστεί ο αλγόριθμος εισαγωγής στοιχείου σε ουρά.

Υλοποίηση ουράς με χρήση πίνακα:

```
! Q          = Η ουρά
! size       = Το μέγιστο μέγεθος της ουράς
! front      = Δείκτης στο πρώτο στοιχείο της ουράς
! rear       = Δείκτης στο τελευταίο στοιχείο της ουράς
! x          = Τιμή του στοιχείου που θα προστεθεί
Αλγόριθμος enqueue
Δεδομένα // Q, front, rear, size //
Διάβασε x
Αν rear = size - 1 τότε
    Γράψε "Η ουρά είναι γεμάτη"
Αλλιώς
    rear <- rear + 1
    Q[rear] <- x
    Γράψε "Το στοιχείο προστέθηκε στην ουρά"
    ! Αν η ουρά ήταν άδεια, ενημέρωσε και τον δείκτη αρχής
    Αν front < 0 τότε
        front <- rear
    Τέλος_Αν
Τέλος_Αν
Τέλος enqueue
```

19. Να σχεδιαστεί ο αλγόριθμος εξαγωγής στοιχείου από ουρά.

Υλοποίηση ουράς με χρήση πίνακα:
! Q = Η ουρά
! front = Δείκτης στο 1ο στοιχείο της ουράς
! rear = Δείκτης στο τελευταίο στοιχείο της ουράς
Αλγόριθμος dequeue
Δεδομένα // Q, front, rear //
Αν front < 0 τότε
 Εκτύπωσε "Η ουρά είναι άδεια"
Αλλιώς
 value <- Q[front]
 front <- front + 1
 ! Έλεγχος αν η ουρά έγινε κενή
 Αν front > rear τότε
 front <- rear <- -1
 Τέλος_Αν
 Εκτύπωσε "Το στοιχείο ", value, " αφαιρέθηκε από την ουρά"
Τέλος_Αν
Τέλος dequeue

20. Να σχεδιαστεί ο αλγόριθμος εισαγωγής στοιχείου σε στοίβα.

Στοίβα με χρήση πίνακα:
! s = Η στοίβα
! size = Μέγιστο πλήθος στοιχείων στοίβας
! top = Δείκτης στο τελευταίο στοιχείο της στοίβας
! x = Τιμή στοιχείου που θα προστεθεί
Αλγόριθμος push
Δεδομένα // s, size, top //
Διάβασε x
Αν top = size - 1 τότε
 Εκτύπωσε "Η στοίβα είναι γεμάτη"
Αλλιώς
 top <- top + 1
 s[top] <- x
 Εκτύπωσε "Το στοιχείο προστέθηκε"
Τέλος_Αν
Τέλος push

21. Να σχεδιαστεί ο αλγόριθμος εξαγωγής στοιχείου από στοίβα.

Στοίβα με χρήση πίνακα:
! s = Η στοίβα
! top = Δείκτης στο τελευταίο στοιχείο της στοίβας
Αλγόριθμος pop
Δεδομένα // s, top //
Αν top < 0 τότε
 Εκτύπωσε "Η στοίβα είναι άδεια"
Αλλιώς
 value <- s[top]
 top <- top - 1
 Εκτύπωσε value
Τέλος_Αν
Τέλος pop

22. Να σχεδιαστεί ο αλγόριθμος της σειριακής αναζήτησης.

Σειριακή αναζήτηση σε πίνακα:

```
! array    Ο πίνακας στον οποίο αναζητούμε το στοιχείο
! size     Το μέγεθος του πίνακα
! key      Το στοιχείο που αναζητούμε
Αλγόριθμος serialSearch
Δεδομένα // array, size //
Διάβασε key
Για i Από 0 Μέχρι size - 1
    Αν array[i] = key τότε
        Εκτύπωση "Το στοιχείο βρέθηκε στη θέση ", i
        Τέλος serialSearch
    Τέλος_Αν
Τέλος_επανάληψης
Εκτύπωσε "Το στοιχείο δεν βρέθηκε"
Τέλος serialSearch
```

23. Να σχεδιαστεί ο αλγόριθμος της δυαδικής αναζήτησης.

Δυαδική αναζήτηση σε ταξινομημένο πίνακα

```
! array    Ο πίνακας στον οποίο αναζητούμε το στοιχείο
! size     Το μέγεθος του πίνακα
! key      Το στοιχείο που αναζητούμε
Αλγόριθμος binarySearch
Δεδομένα // array, size //
Διάβασε key
lower <- 0
upper <- size - 1
Όσο lower <= upper Επανάλαβε
    middle <- (upper + lower) / 2
    Αν key = array[middle] τότε
        Εκτύπωση "Το στοιχείο βρέθηκε στη θέση ", middle
        Τέλος binarySearch
    Τέλος_Αν
    Αν key < array[middle] τότε
        upper <- middle - 1
    Αλλιώς
        lower <- middle + 1
    Τέλος_Αν
Τέλος_επανάληψης
Εκτύπωσε "Το στοιχείο δεν βρέθηκε"
Τέλος binarySearch
```

24. Να σχεδιαστεί ο αλγόριθμος της ταξινόμησης φυσαλίδας.

Αλγόριθμος bubbleSort

```
Δεδομένα // array, size //      ! size το μέγεθος πίνακα
Για i Από 0 Μέχρι size - 2
    Για j Από 0 Μέχρι size - 2 - i
        Αν array[j] > array[j + 1] τότε
            Αντιμετάθεσε array[j], array[j + 1]
        Τέλος_Αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος bubbleSort
```

25. Τι είναι ένα σύστημα Πολυδιεργασίας (Multitasking System);

Ένα λειτουργικό σύστημα Πολυδιεργασίας επιτρέπει την εκτέλεση πολλών διεργασιών (tasks) ταυτόχρονα. Δηλαδή το λειτουργικό σύστημα μπορεί να χειριστεί πολλές διαφορετικές εργασίες δίνοντας την εντύπωση ότι οι εφαρμογές εκτελούνται ταυτόχρονα ακόμη κι αν τρέχουν σε μονοπύρρηνο επεξεργαστή.

26. Τι εννοούμε με τον όρο διεργασία; Σε τι διαφέρει από το πρόγραμμα;

Διεργασία είναι το στιγμιότυπο ενός προγράμματος όταν εκτελείται στον υπολογιστή. Το πρόγραμμα είναι οι στατικές εντολές που το αποτελούν ενώ η διεργασία συνίσταται όταν αυτές εκτελούνται. Μια διεργασία περιλαμβάνει εκτός από τις εντολές και τα δεδομένα που δημιουργεί στη μνήμη ή τους καταχωρητές του επεξεργαστή.

Σε αναλογία θα λέγαμε ότι πρόγραμμα είναι η συνταγή ενός φαγητού και διεργασία η διαδικασία μαγειρέματος του.

27. Πως λειτουργεί η σελιδοποίηση; Τι συμβαίνει όταν μια σελίδα της εικονικής μνήμης δεν αντιστοιχεί σε ενότητα της φυσικής μνήμης;

Η σελιδοποίηση είναι ένας τρόπος διαχείρισης μνήμης που χρησιμοποιούν τα λειτουργικά συστήματα ώστε να είναι δυνατή η εκτέλεση προγραμμάτων που δεν χωρούν στη φυσική μνήμη RAM.

Η εικονική μνήμη χωρίζεται σε ίσα τμήματα που καλούνται σελίδες. Η φυσική μνήμη χωρίζεται κι αυτή σε τμήματα που έχουν ίδιο μέγεθος με τις σελίδες και καλούνται πλαίσια.

Όταν το λειτουργικό σύστημα αντιλαμβάνεται ότι χρειάζεται μια σελίδα της εικονικής μνήμης που δεν υπάρχει σε πλαίσιο της φυσικής μνήμης τότε μεταφέρει κάποιο άλλο πλαίσιο που δεν χρειάζεται στη δευτερεύουσα μνήμη και φορτώνει τη σελίδα της εικονικής μνήμης στο άδειο πλαίσιο.

28. Τι λέγεται εικονική μηχανή (virtual machine);

Η εικονική μηχανή είναι μια εξομίωση ενός ξεχωριστού υπολογιστή μέσα στον υπολογιστή μας. Λειτουργεί σαν ένας κανονικός υπολογιστής, δηλαδή κάνει boot, επανεκκίνηση και τερματισμό λειτουργίας. Για το λόγο αυτό θα πρέπει να εγκαταστήσουμε ένα λειτουργικό σύστημα που μπορεί να είναι διαφορετικό από αυτό του φυσικού υπολογιστή. Χρησιμοποιεί τις μονάδες εισόδου/εξόδου και αποθηκευτικά μέσα του φυσικού υπολογιστή.

29. Πόσα είδη διεργασιών υπάρχουν; Σε ποιες καταστάσεις μπορεί να βρίσκεται μια διεργασία;

Υπάρχουν 2 βασικά είδη διεργασιών:

Διεργασίες συστήματος: Είναι διεργασίες που δημιουργούνται από το λειτουργικό σύστημα. Εκτελούν κρίσιμες λειτουργίες όπως διαχείριση μνήμης και άλλων υποσυστημάτων.

Διεργασίες χρήστη: Είναι διεργασίες που δημιουργούνται από τα προγράμματα που εκτελούν οι χρήστες.

Καταστάσεις διεργασιών:

- Νέα (New): Η διεργασία έχει δημιουργηθεί αλλά δεν εκτελείται ακόμη.
- Έτοιμη (Ready): Η διεργασία αναμένει την ΚΜΕ να την εκτελέσει.

- Εκτελείται (Running): Η διεργασία εκτελείται από την ΚΜΕ.
- Σε αναμονή (Waiting/Blocked): Η διεργασία έχει τεθεί σε αναμονή γιατί περιμένει κάποιο συμβάν.
- Τερματίστηκε (Terminated): Η διεργασία εκτελέστηκε και τερματίστηκε.

30. Τι ονομάζεται εσωτερικός και εξωτερικός κατακερματισμός;

Εσωτερικός κατακερματισμός είναι η μνήμη που δαπανάται επιπλέον για μια διεργασία. Είναι ορατή μόνο από τη διεργασία που ζητά μνήμη. Αυτό συμβαίνει γιατί η ποσότητα μνήμης που δεσμεύεται είναι μεγαλύτερη ή ίση από αυτή που ζητά η διεργασία. Για παράδειγμα ας υποθέσουμε ότι ένα λειτουργικό σύστημα δεσμεύει τη μνήμη σε τμήματα των 4KB. Η διεργασία ζητά 6KB. Θα της παραχωρηθούν 8KB και επομένως 2KB θα μείνουν ανεκμετάλλευτα.

Εξωτερικός κατακερματισμός είναι η ελεύθερη μνήμη που δεν είναι συνεχόμενη. Αυτό καθιστά δύσκολη την εξυπηρέτηση διεργασιών που χρειάζονται μεγάλο συνεχόμενο χώρο μνήμης.

31. Τι συμβαίνει κατά την ανταλλαγή μνήμης;

Το λειτουργικό σύστημα δεσμεύει ένα χώρο από τη δευτερεύουσα μνήμη (σκληρό δίσκο) και τον χρησιμοποιεί σαν επέκταση της κύριας μνήμης (RAM). Αυτό είναι το swar file. Όταν τρέχουν πολλές διεργασίες που δεν χωρούν ταυτόχρονα στην κύρια μνήμη τότε κάποια δεδομένα αποθηκεύονται στο swar file. Όταν πρέπει να εξυπηρετηθεί μια διεργασία που βρίσκεται στο swar file τότε γίνεται ανταλλαγή των δεδομένων. Δηλαδή κάποια άλλα δεδομένα αποθηκεύονται στο πολύ αργότερο swar file και τα δεδομένα της διεργασίας που θα εξυπηρετηθεί μεταφέρονται στην κύρια μνήμη.

32. Να ορισθούν οι έννοιες "Μερισμός χρόνου" (Time sharing) και "επεξεργασία Πραγματικού χρόνου" (Real Time processing).

Μερισμός χρόνου είναι η διαχείριση των πόρων ενός υπολογιστή ώστε πολλοί χρήστες ή διεργασίες να μοιράζονται τον χρόνο επεξεργασίας της ΚΜΕ. Το λειτουργικό σύστημα κατανέμει το χρόνο του επεξεργαστή σε πολλές διεργασίες έτσι ώστε να φαίνεται ότι αυτές τρέχουν ταυτόχρονα.

Στην επεξεργασία πραγματικού χρόνου η διαχείριση των γεγονότων πρέπει να γίνεται σε αυστηρά καθορισμένη χρονική διάρκεια.

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 μήκος κύματος εκπομπής. Η μετάδοση του σήματος είναι αξονική.

Στα πολύτροπα καλώδια το οπτικό σήμα ανακλάται ανάλογα με την είσοδο σε διαφορετικές γωνίες κι έτσι έχουμε πολλούς δρόμους μετάδοσης.

34. Περιγράψτε την λειτουργία του δρομολογητή (router) και του μεταγωγέα (Ethernet Switch).

Δρομολογητής (Router)

Μπορεί να ενώσει διαφορετικά υποδίκτυα μεταξύ τους και αναλαμβάνει τη δρομολόγηση των πακέτων μεταξύ τους. Λειτουργεί στο επίπεδο OSI 3. Ένας δρομολογητής για παράδειγμα μπορεί να συνδέσει ένα τοπικό δίκτυο στο διαδίκτυο μετατρέποντας τις ιδιωτικές διευθύνσεις σε δημόσιες.

Μεταγωγέας (Ethernet Switch)

Λειτουργεί στο ίδιο τοπικό δίκτυο. Γνωρίζει σε ποια διεύθυνση βρίσκεται κάθε συσκευή του δικτύου και προωθεί τα πακέτα προς τη συσκευή για την οποία προορίζονται. Αποφεύγεται έτσι η άσκοπη αναμετάδοση των πακέτων προς άλλους κόμβους.

35. Τι είναι η οπτική ίνα; Να περιγράψετε την λειτουργία της.

Η οπτική ίνα είναι ένα μέσο μετάδοσης που χρησιμοποιεί το φως για τη μεταφορά δεδομένων.

Ο πυρήνας είναι κατασκευασμένος από γυαλί ή πλαστικό και επιτρέπει τη μετάδοση του οπτικού σήματος με πολύ υψηλές ταχύτητες και χαμηλές απώλειες σήματος.

Ο πυρήνας περιβάλλεται από το εσωτερικό κάλυμμα που έχει χαμηλότερο δείκτη διάθλασης και διατηρεί το φως μέσα στον πυρήνα με εσωτερική ανάκλαση.

Τέλος υπάρχει το εξωτερικό περίβλημα που προσφέρει προστασία από φθορές και μηχανικές καταπονήσεις.

Τα δεδομένα που πρόκειται να μεταδοθούν μετατρέπονται σε παλμούς οπτικού σήματος συνήθως από μια πηγή φωτός LED ή Laser. Ταξιδεύουν μέσω της οπτικής ίνας και στο άλλο άκρο μετατρέπονται σε ηλεκτρικά σήματα.

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.

Στην επικοινωνία Half Duplex όταν το ένα άκρο του καναλιού επικοινωνίας είναι πομπός, το άλλο είναι δέκτης. Όταν το άλλο άκρο είναι πομπός το πρώτο γίνεται δέκτης. Τα δύο άκρα δηλαδή μπορούν είναι πομποί και δέκτες αλλά όχι ταυτόχρονα.

Στην επικοινωνία Full Duplex τα δύο άκρα επικοινωνίας μπορούν να είναι ταυτόχρονα πομποί και δέκτες.

37. Τι είναι το X.25; Να περιγράψετε τα επίπεδα του.

Το X.25 είναι ένα πρωτόκολλο επικοινωνίας που χρησιμοποιεί τη μεταγωγή πακέτων για τη μετάδοση δεδομένων. Χρησιμοποιεί τα 3 πρώτα επίπεδα του OSI.

Φυσικό επίπεδο (Physical Level): Ασχολείται με τη φυσική σύνδεση των συσκευών. Καθορίζει τον τρόπο μετάδοσης των ηλεκτρικών ή οπτικών σημάτων.

Επίπεδο ζεύξης δεδομένων (Data Link Layer): Ασχολείται με την αξιοπιστία στη μετάδοση δεδομένων. Χωρίζει τα δεδομένα σε πλαίσια (frames). Εντοπίζει και διορθώνει τα σφάλματα μετάδοσης.

Επίπεδο δικτύου (Network Layer): Ασχολείται με τη μεταφορά δεδομένων μέσω πακέτων. Συνδέει λογικά δύο τερματικούς σταθμούς ή λογικά κυκλώματα.

38. Τι γνωρίζετε για την τεχνολογία ADSL;

Το ADSL χρησιμοποιεί το υπάρχον δίκτυο τηλεφωνίας για τη μετάδοση δεδομένων σε σχετικά υψηλούς ρυθμούς. Χρησιμοποιεί το ανεκμετάλλευτο εύρος ζώνης από τη μετάδοση φωνής. Έτσι από το ίδιο κανάλι περνούν φωνή και δεδομένα.

Χαρακτηριστικό του ADSL είναι ότι η μετάδοση δεδομένων είναι ασύμμετρη. Η μέγιστη ταχύτητα μετάδοσης είναι 24/1 Mbps, δηλαδή 24 Mbps downstream και 1 Mbps upstream.

Η σύνδεση ADSL είναι μόνιμη μεταξύ των κόμβων, δηλαδή δεν υπάρχει σύνδεση και αποσύνδεση όπως συμβαίνει με τις τηλεφωνικές κλήσεις.

39. Περιγράψτε την δομή ενός δικτύου ATM καθώς και τις κατηγορίες υπηρεσιών που προσφέρει.

Το ATM είναι μια τεχνολογία μετάδοσης δεδομένων που βασίζεται στη μεταγωγή πακέτων αλλά τα πακέτα έχουν σταθερό μέγεθος και ονομάζονται κυψέλες (cells). Σχεδιάστηκε για τη μεταφορά δεδομένων αλλά και τη μετάδοση πληροφορίας σε πραγματικό χρόνο (φωνή, βίντεο).

Χρησιμοποιεί τη μεταγωγή κυκλώματος για τη δρομολόγηση των πακέτων. Τα μέσα μετάδοσης μπορεί να είναι καλώδια χαλκού, ομοαξονικά ή οπτικές ίνες.

Υπάρχουν 2 κύριες κατηγορίες υπηρεσιών:

- Σταθερού ρυθμού (Constant Bit Rate): Χρησιμοποιούνται για εφαρμογές με απαιτήσεις σταθερής ροής μετάδοσης (πχ. φωνή ή ζωντανό βίντεο).
- Μεταβαλλόμενου ρυθμού (Variable Bit Rate): Χρησιμοποιούνται για εφαρμογές με απαιτήσεις μεταβαλλόμενης ροής όπως video on demand ή διαδραστικές εφαρμογές.

40. Τι είναι το PSTN και το ISDN; Να αναφέρετε τις λειτουργίες τους.

Το PSTN είναι το παραδοσιακό δημόσιο τηλεφωνικό δίκτυο. Για την επικοινωνία χρησιμοποιεί τη μεταγωγή κυκλώματος. Αρχικά σχεδιάστηκε για αναλογικές τηλεφωνικές κλήσεις αλλά στη συνέχεια προστέθηκε δυνατότητα ψηφιακής μετάδοσης. Η υποδομή των μέσων μετάδοσης μπορεί να είναι καλώδια χαλκού ή οπτικές ίνες.

Το ISDN είναι ψηφιακή τεχνολογία για μετάδοση φωνής, δεδομένων, βίντεο ή άλλων υπηρεσιών μέσα από την υποδομή του παραδοσιακού τηλεφωνικού δικτύου. Σχεδιάστηκε σαν εξέλιξη του PSTN και υποστηρίζει ταυτόχρονη μετάδοση φωνής και δεδομένων με υψηλότερη ταχύτητα και αξιοπιστία. Σήμερα η χρήση του είναι περιορισμένη καθώς έχει αντικατασταθεί από άλλες τεχνολογίες (ADSL, VOIP, οπτικές ίνες).

41. Ποια είναι τα βασικά στάδια κατά την επεξεργασία εικόνων bitmap;

Κατά την επεξεργασία μιας εικόνας bitmap μπορούμε να ορίσουμε:

Την ανάλυση: Εκφράζεται σε dpi (dots per inch) δηλαδή πόσα pixels θα υπάρχουν ανά ίντσα. Όσο υψηλότερη η ανάλυση τόσο καλύτερη η ευκρίνεια της εικόνας.

Το χρωματικό μοντέλο: Υπάρχουν διάφορα χρωματικά μοντέλα για να επιλέξουμε (RGB, CMYK κλπ).

Το βάθος χρώματος: Πόσα χρώματα θα υποστηρίζει η εικόνα. Για βάθος 8bit έχουμε ασπρόμαυρη εικόνα (με 256 διαβαθμίσεις του γκρι), για βάθος 24bit έχουμε true color (16.7 εκατομμύρια χρώματα) κλπ.

Το μέγεθος: Εξαρτάται από τις διαστάσεις της εικόνας (ύψος και πλάτος), την ανάλυση, το βάθος χρώματος και τον αλγόριθμο συμπίεσης.

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 φίλτρα χρησιμοποιούνται για τη δημιουργία εφέ σε ολόκληρη την εικόνα ή μέρος της. Υπάρχουν πολλά είδη φίλτρων. Τέσσερα βασικά είναι:

- Blur: Θάμπωμα εικόνας.
- Sharpen: Όξυνση εικόνας.
- Render: Δημιουργεί οπτικά φαινόμενα (φωτιά, σύννεφα κλπ).
- Distort: Παραμόρφωση εικόνας.

43. Περιγράψτε ποιο είναι το περιεχόμενο σ' ένα αρχείο διανυσματικών γραφικών.

Τα διανυσματικά γραφικά αποτελούνται από γεωμετρικά σχήματα (γραμμές, κύκλοι, πολύγωνα κλπ) και περιγράφονται με μαθηματικές σχέσεις μεταξύ των συντεταγμένων τους, των γωνιών και των αποστάσεων. Το μέγεθός τους εξαρτάται από τον αριθμό και την πολυπλοκότητα των γεωμετρικών σχημάτων. Είναι κατά κανόνα σημαντικά μικρότερα σε μέγεθος από εικόνες bitmap καθώς περιέχουν μόνο πληροφορίες για τη σχεδίαση των γεωμετρικών σχημάτων.

44. Αναφέρετε πλεονεκτήματα και μειονεκτήματα των διανυσματικών γραφικών.

Πλεονεκτήματα:

- Μικρό μέγεθος.
- Η ποιότητα της εικόνας παραμένει αναλλοίωτη αν αλλάξουμε διαστάσεις ή την περιστρέψουμε.
- Είναι ανεξάρτητα ανάλυσης και προσαρμόζονται αυτόματα στην ανάλυση του μέσου που προβάλλονται.

Μειονεκτήματα:

- Δεν μπορούν να απεικονίσουν φωτορεαλιστικές εικόνες όπως φωτογραφίες. Αυτό γιατί κάθε γεωμετρικό σχήμα μπορεί να έχει ένα καθορισμένο χρώμα και δεν μπορούμε να επέμβουμε στο χρώμα κάθε pixel όπως στις εικόνες bitmap.

45. Ποιες είναι οι βασικές διαφορές μεταξύ bitmap graphics και vector graphics;

Τα διανυσματικά (vector) γραφικά χρησιμοποιούν γεωμετρικά σχήματα (γραμμές, κύκλους, πολύγωνα κλπ) για να σχεδιάσουν μια εικόνα.

Τα bitmap είναι ουσιαστικά ένας δισδιάστατος πίνακας όπου ορίζουμε το χρώμα κάθε σημείου του. Για το λόγο αυτό:

Τα vectors έχουν μικρότερο μέγεθος αρχείου καθώς περιέχουν μόνο πληροφορίες για τις συντεταγμένες, γωνίες και απόσταση των γεωμετρικών σχημάτων. Σε αντίθεση τα bitmap αποθηκεύουν πληροφορίες χρώματος για κάθε κουκίδα της εικόνας.

Τα vectors είναι ανεξάρτητα από την ανάλυση (dpi). Προσαρμόζονται αυτόματα στην ανάλυση του μέσου εμφάνισης κάτι που δεν ισχύει για τα bitmap.

Στα vectors μπορούμε να αλλάξουμε τις διαστάσεις ή να περιστρέψουμε την εικόνα χωρίς απώλειες στην ποιότητα. Στα bitmap η ποιότητα της εικόνας υποβαθμίζεται αν μεγαλώσουμε τις διαστάσεις της.

Στα διανυσματικά γραφικά το κάθε γεωμετρικό σχήμα μπορεί να έχει ένα χρώμα οπότε δεν είναι δυνατή η αναπαράσταση φωτορεαλιστικών εικόνων. Στα bitmap μπορούμε να επέμβουμε στο χρώμα κάθε σημείου της εικόνας οπότε είναι ιδανικά για την απεικόνιση φωτογραφιών.

- 46. Περιγράψτε τις διαφορές μεταξύ των προτύπων TIFF, JPEG, GIF. Που χρησιμοποιείται το καθένα;**
- TIFF:** Υποστηρίζει πραγματικό χρώμα και συμπίεση χωρίς απώλειες. Επειδή δεν υπάρχουν απώλειες είναι κατάλληλο για αρχειοθέτηση εικόνων. Χρησιμοποιείται στην τυπογραφία. Στο διαδίκτυο προτιμάται το πρότυπο jpeg καθώς έχει μικρότερο μέγεθος και φορτώνει γρηγορότερα.
- JPEG:** Υποστηρίζει πραγματικό χρώμα. Ο αλγόριθμος συμπίεσης είναι απωλεστικός δηλαδή κατά την αποθήκευση κάποιες πληροφορίες χάνονται μόνιμα. Στις φωτογραφίες αυτές οι απώλειες δεν είναι εμφανείς και χρησιμοποιείται για την απεικόνιση φωτογραφιών στο διαδίκτυο καθώς οι εικόνες έχουν μικρό μέγεθος και φορτώνουν γρήγορα.
- GIF:** Υποστηρίζει έως 256 χρώματα οπότε δεν είναι κατάλληλο για φωτογραφίες. Η συμπίεση κατά την αποθήκευση δεν έχει απώλειες κι έτσι είναι ιδανικό για δημιουργία λογότυπων ή μακετών. Υποστηρίζει επίσης διαφάνεια και animation.
- 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.**
- Τα χρώματα που αναφέρονται είναι σε δεκαεξαδική μορφή. Κάθε ζευγάρι τιμών αναφέρεται σε ένα χρώμα του χρωματικού μοντέλου RGB και παίρνει τιμές 00-FF δηλαδή 0-255 στο δεκαδικό σύστημα. Στο RGB (Red, Green, Blue) υπάρχουν 3 βασικά κανάλια χρώματος. Η τιμή 0 αντιστοιχεί στην απουσία χρώματος για το κανάλι ενώ η 255 στο πλήρες χρώμα. Έτσι το RGB(255, 0, 0) είναι το κόκκινο χρώμα, RGB(0, 255, 0) το πράσινο και RGB(0, 0, 255) το μπλε.
- Στο χρωματικό μοντέλο RGB συμπληρωματικό είναι το χρώμα που η μίξη του με το άλλο χρώμα δημιουργεί λευκό φως, δηλαδή ο συνδυασμός τους παράγει RGB(255, 255, 255).
- Έτσι για τα βασικά χρώματα ισχύει:
- Το κόκκινο RGB(255, 0, 0) έχει συμπληρωματικό το κυανό RGB(0, 255, 255)
- Το πράσινο RGB(0, 255, 0) έχει συμπληρωματικό το ματζέντα RGB(255, 0, 255)
- Το μπλε RGB(0, 0, 255) έχει συμπληρωματικό το κίτρινο RGB(255, 255, 0).
- Τα συμπληρωματικά χρώματα προσφέρουν τη μέγιστη αντίθεση μεταξύ τους.
- a) #FF0000 = RGB(255, 0, 0): Έχει συμπληρωματικό το κυανό RGB(0, 255, 255) ή #00FFFF.
b) #FFFFFF = RGB(255, 255, 255): Έχει συμπληρωματικό το μαύρο RGB(0, 0, 0) ή #000000.
c) #00FFFF = RGB(0, 255, 255): Έχει συμπληρωματικό το κόκκινο RGB(255, 0, 0) ή #FF0000.
d) #000000 = RGB(0, 0, 0): Έχει συμπληρωματικό το λευκό RGB(255, 255, 255) ή #FFFFFF.
e) #A1A1A1 = RGB(161, 161, 161): Έχει συμπληρωματικό άλλη απόχρωση του γκρι RGB(94, 94, 94).
- 48. Εξηγήστε την τεχνική Antialiasing και Alpha channel που χρησιμοποιείται σε εφαρμογές επεξεργασίας εικόνας και φωτορεαλιστικής απεικόνισης αντικειμένων.**
- Το antialiasing είναι μια τεχνική που χρησιμοποιείται για να μειώσει τα "οδοντωτά" άκρα στις εικόνες (πχ. σε μια διαγώνια γραμμή). Για να επιτευχθεί χρησιμοποιούνται ενδιάμεσα χρώματα ανάμεσα στις άκρες της γραμμής και του περιβάλλοντα καμβά ώστε να εξομαλύνουν την οδόντωση.

Το Alpha channel είναι ένα επιπλέον κανάλι που περιέχει πληροφορίες για τη διαφάνεια χρώματος του κάθε στοιχείου της εικόνας. Το 0 αντιστοιχεί σε πλήρη διαφάνεια και το 1 (ή 255 σε κάποια χρωματικά μοντέλα) σε πλήρη αδιαφάνεια.

49. Τι είναι το encapsulation; Δώστε ένα παράδειγμα σε γλώσσα προγραμματισμού Java.

Η ενθυλάκωση (encapsulation) είναι ένα βασικό χαρακτηριστικό του αντικειμενοστραφούς προγραμματισμού. Αποκρύπτει τις εσωτερικές λεπτομέρειες ενός αντικειμένου και εμφανίζει μόνο τα στοιχεία που είναι απαραίτητα για την αλληλεπίδραση με το αντικείμενο. Με την ενθυλάκωση προστατεύονται τα δεδομένα από εξωτερικές παρεμβάσεις και η πρόσβαση σ' αυτά γίνεται μόνο μέσω κατάλληλων μεθόδων του αντικειμένου.

```
class Person {
    // Η πρόσβαση στις μεταβλητές name, age
    // δεν επιτρέπεται έξω από την κλάση
    private String name;
    private int age;

    Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}

// Τρόπος κλήσης της Person
public class Main {
    public static void main(String[] args) {
        Person person = new Person("George", 20);
        // Μπορούμε να αλλάξουμε μόνο την ηλικία γιατί μόνο για αυτή
        // υπάρχει κατάλληλη δημόσια μέθοδος
        person.setAge(30);

        // Η παρακάτω εντολή θα προκαλέσει σφάλμα καθώς δεν
        // επιτρέπεται η πρόσβαση απευθείας στη name
        // person.name = "Dimitris";

        // Μπορούμε να εμφανίσουμε όνομα και ηλικία μόνο
        // μέσω των κατάλληλων δημόσιων μεθόδων
        System.out.println("Name: " + person.getName());
        System.out.println("Age: " + person.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 Vowels {
    public static void main(String[] args) {
        // Δημιουργώ ένα string με τα φωνήεντα και ένα παράλληλο
        // πίνακα που θα κρατάει τις εμφανίσεις κάθε φωνήεντος
        String vowels = "aeiou";
        int[] vCount = new int[vowels.length()];

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter text: ");
        String text = scanner.nextLine();
        scanner.close();
        text = text.toLowerCase();

        for (int i = 0; i < text.length(); i++) {
            // Σημειώνω σε ποια θέση του string φωνηέντων
            // υπάρχει ο χαρακτήρας του κειμένου
            int index = vowels.indexOf(text.charAt(i));

            // Αν είναι φωνήεν αύξησε την αντίστοιχη
            // θέση του παράλληλου πίνακα
            if (index != -1) vCount[index]++;
        }

        System.out.println("Vowels in text:");
        for (int i = 0; i < vowels.length(); i++) {
            System.out.println(vowels.charAt(i) + " -> " + vCount[i]);
        }
    }
}
```

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.

```
import javax.swing.*;
import java.awt.*;
import java.util.Scanner;

public class Circle extends JFrame {
    private int radius, colorIndex;
    private Color[] color = new Color[]
        {Color.red, Color.green, Color.blue};

    Circle(int radius, int colorIndex) {
        this.radius = radius;
        this.colorIndex = colorIndex;
        setSize(400, 400);
    }
}
```

```
setVisible(true);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

@Override
public void paint(Graphics g) {
    g.setColor(color[colorIndex]);
    // Σχεδίαση κύκλου στο κέντρο του παραθύρου
    g.fillOval((getWidth() - radius * 2) / 2,
               (getHeight() - radius * 2) / 2,
               radius * 2,
               radius * 2);
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter circle radius: ");
    int radius = scanner.nextInt();
    System.out.println("Colors: 0. Red, 1. Green, 2. Blue");
    System.out.print("Choose a color: ");
    int color = scanner.nextInt();
    scanner.close();

    new Circle(radius, color);
}
}
```

52. Τι είναι τα Demon Threads και τί τα Non-Demon Threads; Από που εκτελούνται και πως;

Τα demon threads είναι νήματα που δεν εμποδίζουν τον τερματισμό του προγράμματος, δηλαδή η εφαρμογή μπορεί να τερματιστεί ενώ ακόμη δεν έχει ολοκληρωθεί η εκτέλεση των demon threads. Χρησιμοποιούνται για να εκτελέσουν απαραίτητες εργασίες στο παρασκήνιο. Εκτελούνται όπως τα κανονικά νήματα αλλά δηλώνονται ως demon πριν ξεκινήσουν (πχ. στη Java χρησιμοποιώντας τη μέθοδο `setDeamon(true)`).

Τα non-demon threads ή απλά threads δεν επιτρέπουν στο πρόγραμμα να τερματίσει μέχρι να ολοκληρωθεί η εκτέλεσή τους. Αποτελούν τα κύρια νήματα της εφαρμογής, είναι τα προκαθορισμένα νήματα και δεν χρειάζονται κάποια ιδιαίτερη δήλωση.

53. Δίνεται το μη ASCII αρχείο DATA.dbs όπου κάθε εγγραφή είναι αντικείμενο που έχει την ακόλουθη δομή: IBAN λογαριασμού (20 χαρακτήρες) Ανάλυση/Κατάθεση ('0'/'1' 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.

Η κύρια κλάση είναι η `BankAccount`. Τα στοιχεία του πελάτη θα κρατούνται με την κλάση `Record`

```
import java.io.*;
```

```
public class BankAccount {
    public static void main(String[] args) {
```

```
double withdrawalSum = 0, depositSum = 0;
double withdrawalMax = 0, depositMax = 0;
Record withdrawalRec = null;
Record depositRec = null;
String fileName = "DATA.dbs";

try {
    RandomAccessFile file = new RandomAccessFile(fileName, "r");
    while (file.getFilePointer() < file.length()) {
        String iban = readFixedString(file, 20);
        char type = file.readChar();
        double amount = file.readDouble();
        String name = readFixedString(file, 40);
        int afm = file.readInt();

        withdrawalSum += type == '0' ? amount : 0;
        depositSum += type == '1' ? amount : 0;

        if (type == '0' && amount > withdrawalMax) {
            withdrawalRec = new Record(iban, type, amount, name, afm);
            withdrawalMax = amount;
        }
        if (type == '1' && amount > depositMax) {
            depositRec = new Record(iban, type, amount, name, afm);
            depositMax = amount;
        }
    }
    file.close();
} catch (Exception e) {
    System.out.println(e.getMessage());
}

System.out.println("Maximum withdrawal: " + withdrawalRec);
System.out.println("Maximum Deposit: " + depositRec);
System.out.println("Total withdrawals: " + withdrawalSum);
System.out.println("Total deposits: " + depositSum);
}

// Διαβάζει ένα len αριθμό bytes από το αρχείο file
// και τα μετατρέπει σε String
public static String readFixedString(RandomAccessFile file, int len)
    throws IOException {
    byte[] bytes = new byte[len];
    file.readFully(bytes);
    return new String(bytes).trim();
}

class Record {
    private String iban;
    private char type;
    private double amount;
    private String name;
    private int afm;

    Record(String iban, char type, double amount,
```

```
        String name, int afm) {
            this.iban = iban;
            this.type = type;
            this.amount = amount;
            this.name = name;
            this.afm = afm;
        }

        public String toString() {
            return "Record [IBAN: " + iban + ", Type: " + type() +
                ", Amount: " + amount + ", Name: " +
                name + ", AFM: " + afm + "]\n";
        }

        public String type() {
            return type == '0' ? "withdrawal" : "Deposit";
        }
    }
}
```

54. Περιγράψτε τη βασική δομή μιας εφαρμογής φτιαγμένης σε OpenGL.

- Αρχικοποίηση και άνοιγμα παραθύρου OpenGL.
- Καθορισμός παραμέτρων σκηνής.
- Καταχώριση callback functions για rendering, resize, είσοδο δεδομένων από πληκτρολόγιο, ποντίκι κλπ.
- Είσοδος στο main event processing loop.
- Καθαρισμός μνήμης και έξοδος.

55. Να αναφέρετε και να περιγράψετε σύντομα τα callback functions της OpenGL.

glutDisplayFunc(): Καλείται όταν χρειάζεται να επανασχεδιαστούν τα pixels στο παράθυρο.

glutOverlayDisplayFunc(): Καλείται για επανασχεδίαση των επικαλυπτόμενων τμημάτων παραθύρου.

glutVisibilityFunc(): Καλείται όταν αλλάζει η ορατότητα του παραθύρου.

glutEntryFunc(): Καλείται όταν το ποντίκι εισέρχεται ή φεύγει από το παράθυρο.

glutReshapeFunc(): Καλείται όταν αλλάζει το μέγεθος του παραθύρου.

glutKeyboardFunc(): Καλείται όταν πατιέται ένα πλήκτρο του πληκτρολογίου.

glutSpecialFunc(): Καλείται όταν πατιούνται τα ειδικά πλήκτρα (function keys, βελάκια κλπ).

glutMouseFunc(): Καλείται όταν πατιέται ένα κουμπί του ποντικιού.

glutMotionFunc(): Καλείται όταν κουνιέται το ποντίκι και πατιέται ένα πλήκτρο του.

glutPassiveMotionFunc(): Καλείται όταν κουνιέται το ποντίκι και δεν πατιέται κάποιο πλήκτρο του.

glutIdleFunc(): Καλείται όταν τίποτα άλλο δεν συμβαίνει. Πολύ χρήσιμο για animation.

56. Ποιες εντολές χρησιμοποιούμε για τις εξής λειτουργίες της OpenGL: α. translate β. rotate γ. scale ή stretch;

α) Translate (μετατόπιση)

Μετατόπιση του αντικειμένου κατά x, y, z σε διαφορετική θέση στο χώρο.

`glTranslated(GLdouble x, GLdouble y, GLdouble z)`

`glTranslatef(GLfloat x, GLfloat y, GLfloat z)`

β) Rotate (περιστροφή)

Περιστροφή του αντικειμένου κατά γωνία angle μοιρών γύρω από τον άξονα x, y, z.

```
glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
```

```
glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
```

γ) Scale (κλιμάκωση)

Αλλάζει το μέγεθος του αντικειμένου κατά διαφορετικούς συντελεστές στους άξονες x, y, z.

```
glScaled(GLdouble x, GLdouble y, GLdouble z)
```

```
glScalef(GLfloat x, GLfloat y, GLfloat z)
```

57. Για ποιο λόγο είναι καλό να χρησιμοποιούμε τους τύπους δεδομένων της OpenGL;

Τα λειτουργικά συστήματα μπορεί να έχουν διαφορετικό μέγεθος bits για τους ίδιους τύπους δεδομένων. Η OpenGL είναι ανεξάρτητη λειτουργικού συστήματος. Χρησιμοποιώντας τους τύπους δεδομένων της OpenGL μπορούμε να κάνουμε compile τον κώδικα χωρίς αλλαγές σε οποιαδήποτε πλατφόρμα.

58. Περιγράψτε τη σύνταξη και τις παραμέτρους της glLookAt() (στην OpenGL).

```
glLookAt(GLdouble eyex, GLdouble eyey, GLdouble eyez,  
          GLdouble centerx, GLdouble centery, GLdouble centerz,  
          GLdouble upx, GLdouble upy, GLdouble upz)
```

Όπου:

[eyex, eyey, eyez]: Οι συντεταγμένες που βρίσκεται η κάμερα.

[centerx, centery, centerz]: Το σημείο που κοιτάει η κάμερα.

[upx, upy, upz]: Το "επάνω" της κάμερας δηλαδή η περιστροφή της ως προς τον άξονα.

59. Ποια είναι η λειτουργία της glClear() και ποια της glClearColor3f() (στην OpenGL);

Η glClear() χρησιμοποιείται για να καθαρίσει ένα ή περισσότερα buffer.

Δεν υπάρχει συνάρτηση glClearColor3f(). Υπάρχει η glClearColor() που καθορίζει το χρώμα που θα χρησιμοποιηθεί κατά τον καθαρισμό του color buffer από την glClear().

60. Στην OpenGL όπως και στα περισσότερα API γραφικών χρησιμοποιούμε "normalized values", τι σημαίνει αυτό;

Οι κανονικοποιημένες τιμές (normalized values) χρησιμοποιούνται για να έχουμε την ίδια απεικόνιση ανεξάρτητα από το μέσο προβολής. Συνήθως επιλέγονται τιμές στο εύρος [-1.0, 1.0] ή [0.0, 1.0] όπου για παράδειγμα 0 σημαίνει 0% και 1 σημαίνει 100%. Έτσι στο χρωματικό μοντέλο RGB που οι τιμές κυμαίνονται 0-255 για κάθε χρώμα το 1 αντιστοιχεί στην τιμή 255. Το ίδιο ισχύει και για να καθορίσουμε τη θέση ενός αντικειμένου στην οθόνη.

61. Αναφέρατε τις εντολές (μαζί με τις παραμέτρους τους) για προοπτική (perspective) και για ορθογραφική (orthographic) απεικόνιση (στην OpenGL).

Προοπτική απεικόνιση:

glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far)

gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble near, GLdouble far)

Ορθογραφική απεικόνιση:

glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far)

gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)

62. Με ποιους τρόπους μπορούμε να μετατοπίσουμε ένα αντικείμενο -σχήμα στην OpenGL;

Για να μετακινήσουμε ένα αντικείμενο μπορούμε να χρησιμοποιήσουμε τη συνάρτηση

glTranslatef(GLfloat x, GLfloat y, GLfloat z).

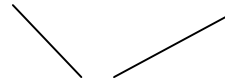
Μπορούμε επίσης να χρησιμοποιήσουμε μήτρες μετασχηματισμού, shaders ή να αλλάξουμε τις συντεταγμένες των κόμβων του σχήματος.

63. Περιγράψτε με εντολές και παραμέτρους της OpenGL, τα βασικά γεωμετρικά σχήματα α) Γραμμές, β) Τρίγωνα, γ) Πολύγωνα. Παραθέστε σχήματα.

α) Γραμμές:

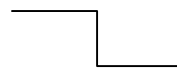
GL_LINES: Γραμμές που δεν ενώνονται. Ορίζουμε τις συντεταγμένες αρχής και τέλους της γραμμής.

```
glBegin(GL_LINES);  
    glVertex2f(-0.5, 0.5);  
    glVertex2f(0.0, 0.0);  
glEnd();
```



GL_LINE_STRIP: Συνεχόμενες γραμμές. Το τέλος της μίας είναι η αρχή της άλλης.

```
glBegin(GL_LINE_STRIP);  
    glVertex2f(-0.5, 0.5);  
    glVertex2f(0.0, 0.5);  
    glVertex2f(0.0, 0.0);  
    glVertex2f(0.5, 0.0);  
glEnd();
```



GL_LINE_LOOP: Συνεχόμενες γραμμές που ενώνονται. Η τελευταία γραμμή ενώνεται με την πρώτη.

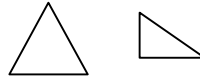
```
glBegin(GL_LINE_LOOP);  
    glVertex2f(-0.5, 0.5);  
    glVertex2f(0.0, 0.0);  
    glVertex2f(0.2, 0.2);  
    glVertex2f(0.5, 0.1);  
glEnd();
```



β) Τρίγωνα:

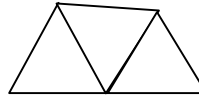
GL_TRIANGLES: Τρίγωνα που δεν ενώνονται. Ορίζουμε τις συντεταγμένες των 3 κορυφών κάθε τριγώνου.

```
glBegin(GL_TRIANGLES);  
    glVertex2f(0.0, 0.5);  
    glVertex2f(-0.5, 0.0);  
    glVertex2f(0.5, 0.0);  
glEnd();
```



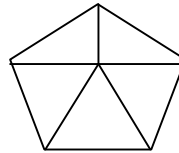
GL_TRIANGLE_STRIP: Τα τρίγωνα μοιράζονται πλευρές και δημιουργούν λωρίδα. Κάθε κορυφή μετά τις 3 πρώτες δημιουργεί τρίγωνο.

```
glBegin(GL_TRIANGLE_STRIP);  
    glVertex2f(0.0, 0.5);  
    glVertex2f(-0.5, 0.0);  
    glVertex2f(0.5, 0.0);  
    glVertex2f(0.75, 0.45);  
    glVertex2f(1.0, 0.0);  
glEnd();
```



GL_TRIANGLE_FAN: Τα τρίγωνα μοιράζονται ένα κοινό κέντρο και σχηματίζουν βεντάλια.

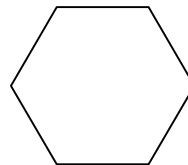
```
glBegin(GL_TRIANGLE_FAN);  
    glVertex2f(0.0, 0.0); // κέντρο  
    glVertex2f(-0.5, -0.5);  
    glVertex2f(0.5, -0.5);  
    glVertex2f(0.7, 0.0);  
    glVertex2f(0.0, 0.3);  
    glVertex2f(-0.7, 0.0);  
glEnd();
```



γ) Πολύγωνα:

GL_POLYGON: Ορίζουμε τις συντεταγμένες κάθε κορυφής του πολυγώνου.

```
glBegin(GL_POLYGON);  
    glVertex2f(-0.5f, -0.5f);  
    glVertex2f(0.5f, -0.5f);  
    glVertex2f(0.7f, 0.0f);  
    glVertex2f(0.5f, 0.5f);  
    glVertex2f(-0.5f, 0.5f);  
    glVertex2f(-0.7f, 0.0f);  
glEnd();
```



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 'c':
```

```
case 'C':  
    glClear(GL_COLOR_BUFFER_BIT);  
    break;  
case 'q':  
case 'Q':  
    exit(0);  
}  
}
```

Θα κληθεί από το κυρίως πρόγραμμα με την εντολή glutKeyboardFunc(keyboard).

65. Περιγράψτε τα βήματα για τη δημιουργία animation στην OpenGL.

Αρχικοποίηση της OpenGL

Δημιουργία παραθύρου, ορισμός συμβάντων για διαχείριση πληκτρολογίου, mouse κλπ. Αυτά μπορούν να γίνουν με συναρτήσεις της βιβλιοθήκης glut.

Δημιουργία της σκηνής

Καθορίζουμε τα γεωμετρικά σχήματα που θα εμφανίζονται στη σκηνή, το φωτισμό κλπ.

Δημιουργία του κύκλου animation

Ο κύκλος animation είναι μια επαναλαμβανόμενη διαδικασία που ενημερώνει και ανανεώνει συνεχώς την οθόνη. Αυτή η διαδικασία περιλαμβάνει:

- Ενημέρωση της κατάστασης: Υπολογίζουμε τις νέες θέσεις, περιστροφές ή άλλες ιδιότητες των αντικειμένων.
- Καθαρισμός της οθόνης: Καθαρίζουμε τα buffer χρώματος και βάθους για να ετοιμάσουμε το νέο καρέ.
- Σχεδίαση των αντικειμένων: Σχεδιάζουμε τα αντικείμενα στη νέα τους θέση.
- Ανταλλαγή των buffer: Εναλλάσσουμε τα buffer για να εμφανιστεί το νέο καρέ στην οθόνη.

Χρονισμός

Για να διασφαλίσουμε ότι ο ρυθμός εμφάνισης των καρέ είναι σταθερός χρησιμοποιούμε ένα χρονιστή, για παράδειγμα τη συνάρτηση glutTimerFunc().

66. Ποιες είναι οι ιδιότητές που έχουν οι επιφάνειες στην OpenGL;

Χρώμα: Ορίζεται από τη μίξη των 3 βασικών χρωμάτων (κόκκινο, πράσινο, μπλε) χρησιμοποιώντας τη συνάρτηση glColor3f() ή glColor4f().

Διαφάνεια: Η διαφάνεια της επιφάνειας καθορίζεται μέσω του alpha channel με τις glBlendFunc() και glColor4f().

Φωτισμός: Ο φωτισμός καθορίζει πως οι επιφάνειες αλληλεπιδρούν με τις πηγές φωτός. Ορίζονται μέσω των συναρτήσεων glLightfv() και glMaterialfv().

Σκίαση: Οι επιφάνειες μπορούν να δέχονται ή να δημιουργούν σκίαση. Χρησιμοποιούμε την glShadeModel().

Υφή: Οι υφές προσθέτουν λεπτομέρειες στην εμφάνιση. Η εφαρμογή υφών γίνεται με τις συναρτήσεις glTexImage2D() ή glTexImage3D().

Καμπυλότητα: Η συνάρτηση glNormal3f() καθορίζει τη γωνία της επιφάνειας σε κάθε σημείο.

- 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;**

Το double buffer είναι μια τεχνική για την αποφυγή τρεμοπαίγματος (flickering) στο animation. Χρησιμοποιούμε δύο περιοχές μνήμης (buffers) για την απεικόνιση των καρέ της εικόνας. Ενώ δείχνουμε την εικόνα στο ένα buffer σχεδιάζουμε το επόμενο καρέ στο άλλο buffer. Στη συνέχεια εναλλάσσουμε τα buffer. Έτσι η οθόνη δείχνει πάντα μια ολοκληρωμένη εικόνα της σκηνής.

Για να ενεργοποιήσουμε το double buffering χρησιμοποιούμε τη συνάρτηση `glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB)`.

Για την εναλλαγή των buffer χρησιμοποιούμε την `glutSwapBuffers()`.

- 68. Για ποιό σκοπό χρειάζονται οι συναρτήσεις κλήσης στην OpenGL και πώς ορίζονται;**

Οι συναρτήσεις κλήσης (callback functions) χρησιμοποιούνται για τη διαχείριση συμβάντων και τον έλεγχο συμπεριφοράς της εφαρμογής. Γράφονται από τον προγραμματιστή και περνιούνται σαν παράμετροι στις βιβλιοθήκες της OpenGL.

Για παράδειγμα μπορούμε να γράψουμε ένα callback για τη σχεδίαση της οθόνης:

```
void display()
{
    // Κώδικας για σχεδίαση της εικόνας
}
```

Στη συνέχεια το καλούμε με τη συνάρτηση βιβλιοθήκης `glutDisplayFunc(display)`. Έτσι κάθε φορά που η OpenGL θέλει να εμφανίσει τη σκηνή θα καλεί τη δική μας συνάρτηση `display()`.

Οι βασικές κατηγορίες συναρτήσεων κλήσης είναι:

`glutDisplayFunc()`: Καλείται όταν πρέπει να επανασχεδιαστεί η σκηνή.

`glutReshapeFunc()`: Καλείται όταν αλλάζει το μέγεθος παραθύρου.

`glutKeyboardFunc()`: Καλείται όταν πατιέται ένα πλήκτρο από το πληκτρολόγιο.

`glutMouseFunc()`: Καλείται όταν πατιέται ένα κουμπί του ποντικιού.

`glutMotionFunc()`: Καλείται όταν πατιέται ένα κουμπί του mouse ενώ κινείται.

`glutPassiveMotionFunc()`: Καλείται όταν κινείται το ποντίκι αλλά δεν πατιέται κάποιο κουμπί του.

`glutIdleFunc()`: Καλείται όταν δεν συμβαίνει κάποιο γεγονός. Πολύ χρήσιμο για animation.

Πριν ξεκινήσει το κύριο loop της εφαρμογής ορίζονται όλες οι callback functions.

- 69. Με ποιές συναρτήσεις γίνεται μοντελοποίηση επιφανειών που ανακλούν στην OpenGL;**

`glLightfv()`: Ρυθμίζει τις πηγές φωτός.

`glMaterialfv()`: Καθορίζει τις οπτικές ιδιότητες της επιφάνειας (πχ. συντελεστή ανάκλασης).

`glNormal3f()`: Χρησιμοποιείται για να καθορίσει τα κανονικά διανύσματα των επιφανειών που είναι απαραίτητα για τον υπολογισμό ανάκλασης του φωτός.

70. Σε πόσες και ποιες συνιστώσες διαχωρίζετε το φως πηγής κατά την μοντελοποίηση πηγών φωτισμού;

Το φως πηγής διαχωρίζεται συνήθως σε 3 συνιστώσες:

Φως περιβάλλοντος (Ambient Light)

Αντιπροσωπεύει το φως που βρίσκεται διάχυτο σε ολόκληρη τη σκηνή. Φωτίζει ομοιόμορφα όλες τις επιφάνειες. Για να το ρυθμίσουμε χρησιμοποιούμε τη συνάρτηση `glLightfv()` περνώντας σαν παράμετρο `GL_AMBIENT`.

Φως Διάχυσης (Diffuse Light)

Είναι το φως που προέρχεται από μια συγκεκριμένη κατεύθυνση και διαχέεται στην επιφάνεια του αντικειμένου. Για να το ρυθμίσουμε χρησιμοποιούμε τη συνάρτηση `glLightfv()` περνώντας σαν παράμετρο `GL_DIFFUSE`.

Φως κατοπτρισμού (Specular Light)

Είναι το φως που ανακλάται από την επιφάνεια του αντικειμένου. Για να το ρυθμίσουμε χρησιμοποιούμε τη συνάρτηση `glLightfv()` περνώντας σαν παράμετρο `GL_SPECULAR`.

71. Με ποια σειρά γίνονται οι διεργασίες για τον υπολογισμό της τελικής εικόνας (rendering) στην OpenGL;

Ο υπολογισμός της τελικής εικόνας στην OpenGL περιλαμβάνει διάφορα στάδια που είναι γνωστά σαν graphics pipeline.

Vertex Specification: Καθορίζει τα αρχικά δεδομένα όπως γεωμετρικά σχήματα, φωτισμό, υφές κλπ.

Vertex Post-processing: Σ' αυτό το στάδιο ο προγραμματιστής έχει ελάχιστη έως καθόλου συμμετοχή. Καθορίζονται οι ορατές κορυφές των σχημάτων και μόνο αυτές περνούν στα επόμενα στάδια.

Primitive Assembly: Οι κορυφές ενώνονται σε πρωτογενή γεωμετρικά σχήματα (γραμμές, τρίγωνα, πολύγωνα).

Rasterization: Τα γεωμετρικά αντικείμενα μετατρέπονται σε θραύσματα (fragments) δηλαδή εικονοστοιχεία (pixels). Κάθε pixel περιέχει πληροφορίες για χρώμα, βάθος, υφή κλπ.

72. Με ποια εντολή και ποιες συναρτήσεις θα πραγματοποιήσετε απόδοση υφής σε γραμμές και επιφάνειες στην OpenGL;

Πριν αποδώσουμε την υφή θα πρέπει να την φορτώσουμε στη μνήμη και να την ενεργοποιήσουμε:

`glTexImage2D`: Φορτώνει την υφή στη μνήμη.

`glEnable()`: Ενεργοποιεί την υφή.

Στη συνέχεια ορίζουμε τις συντεταγμένες της υφής πριν από κάθε κορυφή της γραμμής ή της επιφάνειας με την `glTexCoord2f(GLfloat s, GLfloat t)`.

73. Για ποιο λόγο χρησιμοποιούνται οι λίστες απεικόνισης (display lists) στην γλώσσα OpenGL και πώς συντάσσονται; Δώστε και από ένα παράδειγμα για κάθε περίπτωση.

Οι λίστες απεικόνισης μας δίνουν τη δυνατότητα να αποθηκεύσουμε ένα σύνθετο γεωμετρικό σχήμα χωρίς να το σχεδιάζουμε κάθε φορά από την αρχή.

Για να δημιουργήσουμε ένα display list χρησιμοποιούμε την `GLuint glGenLists(GLint range);`

Με την range καθορίζουμε το πλήθος των display lists που θέλουμε να δημιουργήσουμε. Η συνάρτηση επιστρέφει ένα αναγνωριστικό για το πρώτο στοιχείο της λίστας. Για παράδειγμα:

```
GLuint listIndex = glGenLists(2)
```

δημιουργεί δύο αναγνωριστικά για display lists και επιστρέφει στην listIndex το αναγνωριστικό για το πρώτο στοιχείο. Το δεύτερο στοιχείο θα είναι listIndex + 1.

Στην συνέχεια η λίστα απεικόνισης περικλείεται μέσα στις εντολές glNewList και glEndList δίνοντας το αναγνωριστικό της λίστας στην οποία αναφερόμαστε. Για παράδειγμα:

```
glNewList(listIndex, GL_COMPILE);  
// εντολές για τη δημιουργία του σχήματος  
glEndList();
```

Για να εμφανίσουμε τη λίστα στη σκηνή χρησιμοποιούμε την glCallList(listIndex).

74. Έστω κώδικας: <?php \$a = "6"; \$b = 8; echo \$b - \$a; ?> Ποιο θα είναι το αποτέλεσμα;

Δικαιολογήστε την απάντησή σας.

Η μεταβλητή \$a έχει τιμή "6" (string) ενώ η \$b έχει 8 (integer). Η PHP κάνει αυτόματα μετατροπή του string σε ακέραιο ώστε να είναι έγκυρη η πράξη \$b - \$a. Επομένως θα τυπωθεί 2.

75. Ποιες είναι οι διαφορές μεταξύ των πινάκων της PHP και των πινάκων της C++;

- Στην PHP οι πίνακες είναι δυναμικοί. Μπορούμε να αλλάξουμε το μέγεθος κατά τη διάρκεια εκτέλεσης του προγράμματος. Στη C++ οι πίνακες είναι στατικοί. Το μέγεθός τους πρέπει να δηλωθεί κατά τη δημιουργία τους.
- Στην PHP μπορούμε να ορίσουμε οποιοδήποτε string ή ακέραιο σαν δείκτη προς ένα στοιχείο του πίνακα. Στη C++ οι δείκτες είναι ακέραιοι και υποδηλώνουν την απόσταση του κάθε στοιχείου από την αρχή του πίνακα.
- Στην PHP τα στοιχεία ενός πίνακα μπορεί να είναι διαφόρων τύπων (string, integer, δεκαδικοί κλπ). Στη C++ όλα τα στοιχεία πρέπει να είναι του ίδιου τύπου.

76. Περιγράψτε τα στάδια επεξεργασίας μιας PHP σελίδας από τη στιγμή της αίτησής της μέχρι την εμφάνιση στον browser του χρήστη.

- Υποβολή αιτήματος: Ο χρήστης εισάγει τη διεύθυνση της PHP σελίδας ή υποβάλλει μια φόρμα. Ο browser υποβάλλει ένα αίτημα προς τον server που διαχειρίζεται τις php σελίδες.
- Παραλαβή αιτήματος από τον Web Server: Ο server (πχ Apache) παραλαμβάνει το αίτημα, αναγνωρίζει ότι αφορά αρχείο PHP (από την κατάληξη .php) και το προωθεί στον PHP Interpreter.
- Επεξεργασία από PHP Interpreter: Ο PHP Interpreter εκτελεί τον κώδικα, δημιουργεί το κατάλληλο στατικό ή δυναμικό περιεχόμενο που ζητήθηκε και το επιστρέφει στον web server.
- Επιστροφή αποτελέσματος: Ο browser παραλαμβάνει από το web server το περιεχόμενο που ζητήθηκε και το εμφανίζει στο χρήστη.

77. Αναφέρετε τους κανόνες που αφορούν στην ονομασία των μεταβλητών της PHP;

- Ξεκινούν με το σύμβολο \$.
- Ο επόμενος χαρακτήρας πρέπει να είναι γράμμα ή κάτω παύλα _. Δεν μπορεί να είναι αριθμός.

- Οι υπόλοιποι χαρακτήρες μπορούν να είναι γράμματα, αριθμοί ή κάτω παύλα _.
- Δεν πρέπει να περιέχουν κενά.
- Δεν πρέπει να είναι δεσμευμένη λέξη (πχ echo).
- Είναι case sensitive.

78. Η εντολή \$c = 1; echo ' \$c' . \$c; τι θα δώσει ως τύπωση;
\$c1

79. Έστω κώδικας:

```
<?php $a = "Hello, "; $b = "world!"; echo $a; echo $b; ?>
```

Ποιό θα είναι το αποτέλεσμα; Δικαιολογήστε την απάντησή σας.

Το αποτέλεσμα θα είναι Hello, world!

Θα τυπωθεί το \$a ("Hello, ") και δίπλα του το \$b ("world!") όπως ακριβώς ζητήθηκε.

80. Έστω κώδικας php:

```
$a = 2; echo $a; ++$a; echo $a; --$a; echo $a;
```

Ποιό θα είναι το αποτέλεσμα; Δικαιολογήστε την απάντησή σας.

Θα τυπωθεί 232.

Στην αρχή τυπώνεται η τιμή 2 που έχει το \$a. Στη συνέχεια με την ++\$a αυξάνεται κατά 1 η τιμή του \$a και τυπώνεται. Τέλος με την --\$a μειώνεται κατά 1 η τιμή του \$a και τυπώνεται.

81. Να γραφεί ένα πρόγραμμα PHP το οποίο θα εμφανίζει με αλλαγή γραμμής το μήνυμα "Καλωσήρθατε στην PHP" 5 φορές.

```
<?php
for ($i = 0; $i < 5; $i++) {
    echo "Καλωσήρθατε στην PHP<br>";
}
?>
```

82. Ποιοί είναι οι βασικοί τύποι σχέσεων μεταξύ πινάκων στην PHP(MySql);

Υπάρχουν 3 βασικοί τύποι σχέσεων μεταξύ πινάκων στην MySql:

- Ένα προς Ένα (1:1): Κάθε γραμμή του ενός πίνακα αντιστοιχεί σε μία και μόνο γραμμή του άλλου πίνακα και το αντίστροφο.
- Ένα προς Πολλά (1:N): Μία γραμμή του ενός πίνακα σχετίζεται με πολλές γραμμές του άλλου πίνακα αλλά όχι το αντίστροφο.
- Πολλά προς Πολλά (N:M): Μία γραμμή του ενός πίνακα σχετίζεται με πολλές γραμμές του 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.

Η σελίδα HTML

```
<html>
<body>
  <form method="post" action="text.php">
    <input type="text" name="text1"><br>
    <input type="text" name="text2"><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

Η σελίδα text.php

```
<?php
$text1 = $_POST["text1"];
$text2 = $_POST["text2"];
echo "<i>$text1<br>$text2</i><br>";
?>
```

84. Να γραφεί μία φόρμα η οποία θα δέχεται από τον χρήστη ένα όνομα, ένα επίθετο και έναν αριθμητικό κωδικό. Αν κάποιο 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.

Στην σελίδα HTML που περιέχει τη φόρμα δημιουργώ μια συνάρτηση JavaScript που ελέγχει αν κάποιο πεδίο της φόρμας είναι κενό. Αν υπάρχουν κενά πεδία η συνάρτηση τα εκτυπώνει στο div με id="output" και δεν επιτρέπει την υποβολή της φόρμας (return false) αλλιώς καλείται η print.php

```
<html>
<head>
  <script>
    function checkForm() {
      let message = "";
      const fName = document.getElementById("f-name");
      const lName = document.getElementById("l-name");
      const code = document.getElementById("code");
      const output = document.getElementById("output");

      if (fName.value == "")
        message += "First Name must not be empty<br>";
      if (lName.value == "")
        message += "Last Name must not be empty<br>";
      if (code.value == "")
        message += "Code must not be empty";

      if (message != "") {
        output.innerHTML = message;
        return false;
      }
    }
  </script>
```

```
</head>
<body>
  <form method="post" action="print.php" onsubmit="return checkForm()">
    First Name:<input type="text" id="f-name" name="f-name"><br>
    Last Name: <input type="text" id="l-name" name="l-name"><br>
    Code: <input type="number" id="code" name="code"><br>
    <input type="submit" value="Submit"><br>
  </form>
  <div id="output" style="color: red;"></div>
</body>
</html>
```

Η σελίδα print.php:

```
<?php
$fName = $_POST["f-name"];
$lName = $_POST["l-name"];
$code = $_POST["code"];
echo "Your data:<br>$fName<br>$lName<br>$code<br>";
?>
```

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.

```
<?php
// τυχαίες τιμές στις μεταβλητές
$var1 = "Text1";
$var2 = "Text2";
$Ex1 = 10;
$Ex2 = 20;
?>

<html>
  <body>
    <!-- Επειδή η φόρμα δεν έχει action θα ποστάρει στον εαυτό της -->
    <form method="post">
      Variable name: <input type="text" name="input"><br>
      <input type="submit" value="Submit"><br>
    </form>

    <?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
      $input = $_POST["input"];

      // Το $$ δηλώνει μεταβλητή μεταβλητής. Δηλαδή επιτρέπει να
      // ορίσουμε την τιμή μιας μεταβλητής σαν όνομα μιας άλλης.
      // Η isset() ελέγχει αν υπάρχει αυτή η μεταβλητή.
      // Έτσι αν υπάρχει μεταβλητή που το όνομα της είναι η τιμή
      // της $input τότε την αναθέτουμε στην $output. Αλλιώς
      // η $output θα πάρει τιμή "Not found".
      $output = isset($$input) ? $$input : "Not found";
```

```

        echo "$input = $output";
    }
    ?>

</body>
</html>

```

86. Ποιες είναι οι βασικές διαφορές μεταξύ PHP και JAVASCRIPT;

Η βασική διαφορά είναι ότι η PHP είναι server-side γλώσσα προγραμματισμού ενώ η JavaScript είναι client-side. Έτσι οι βασικές τους διαφορές είναι:

JavaScript	PHP
Εκτελείται στον τοπικό υπολογιστή του χρήστη	Εκτελείται στον server
Σχεδιάστηκε για διαχείριση του frontend	Σχεδιάστηκε για διαχείριση του backend
Εξαρτάται από τον browser του χρήστη. Μπορεί ο χρήστης να την απενεργοποιήσει ή να υπάρχουν διαφορετικές εκδόσεις μεταξύ χρηστών	Δεν μπορεί να απενεργοποιηθεί από το χρήστη. Εκτελείται στο server και τα αποτελέσματα εμφανίζονται σαν HTML στον browser των χρηστών. Η έκδοση είναι κοινή για όλους
Τα δεδομένα και ο κώδικας μπορεί να είναι ορατά στο χρήστη μέσω του browser. Δεν είναι κατάλληλη για φύλαξη ευαίσθητων πληροφοριών	Είναι κατάλληλη για φύλαξη ευαίσθητων πληροφοριών.
Εκτελείται τοπικά και δεν επιβαρύνει τους υπόλοιπους χρήστες	Εκτελείται στο server και η απόδοση εξαρτάται από τον αριθμό χρηστών

87. Πώς μπορούμε να έχουμε καθολικές μεταβλητές μεταξύ πολλών συναρτήσεων; Δώστε από ένα παράδειγμα σε κάθε περίπτωση.

Μπορούμε μέσα στις συναρτήσεις είτε να δηλώσουμε τις μεταβλητές σαν global είτε να κάνουμε χρήση του πίνακα \$GLOBALS.

```

<?php
$counter = 0;

increment1(); // Τυπώνει 1
increment2(); // Τυπώνει 2

$counter++;
echo $counter;    // Τυπώνει 3

function increment1() {    // χρήση της global
    global $counter;
    $counter++;
    echo $counter;
}

function increment2() {    // χρήση της $GLOBALS
    $GLOBALS["counter"]++;
    echo $GLOBALS["counter"];
}
?>

```

88. Γιατί δεν μπορούμε να εκτελέσουμε ένα αρχείο PHP απλά ανοίγοντάς το με έναν διαχειριστή αρχείων όπως γίνεται με τις HTML σελίδες; Δικαιολογήστε την απάντησή σας.

Η PHP εκτελείται στον server και όχι στον τοπικό υπολογιστή. Οι HTML σελίδες εκτελούνται στον browser του υπολογιστή μας και επομένως μπορούμε να τις δούμε άμεσα τοπικά. Οι εντολές PHP θα πρέπει πρώτα να μεταφραστούν σε HTML από το διερμηνευτή εντολών του server για να δούμε τη σελίδα στον browser του υπολογιστή μας.

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
$num = 1;
while ($num < 50) {
    echo "$num ";
    $num += 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.

```
<html>
<body>
<!-- Επειδή δεν ορίζουμε action η φόρμα ποστάρει στον εαυτό της -->
<form method="post">
    Number 1: <input type="number" name="num1" required><br>
    Number 2: <input type="number" name="num2" required><br>
    <input type="radio" name="operation" value="+" required>Add<br>
    <input type="radio" name="operation" value="-" required>Subtract<br>
    <input type="submit" value="Execute"><br>
</form>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $num1 = $_POST["num1"];
    $num2 = $_POST["num2"];
    $operation = $_POST["operation"];
    $result = $operation == "+" ? $num1 + $num2 : $num1 - $num2;
    echo "$num1 $operation $num2 = $result";
}
?>
```

```
</body>
</html>
```

92. Να υλοποιηθεί ένας απλός υπολογισμός για το εμβαδόν ενός κύκλου χρησιμοποιώντας τον τύπο $\text{Εμβαδόν} = \pi \cdot (r \cdot r)$. 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 π ισούται με 3,14).

```
<?php
function getCircleArea($radius) {
    return 3.14 * $radius * $radius;
}
?>

<html>
<body>
<p>Area of circle calculation</p>
<!-- Επειδή η φόρμα δεν περιέχει action ποστάρει στον εαυτό της -->
    <form method="post">
        Radius: <input type="number" step="0.01" name="radius" required><br>
        <input type="submit" value="Calculate"><br>
    </form>
    <?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $r = $_POST["radius"];
        echo "Area of circle is: ".getCircleArea($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
// Προσπαθούμε να συνδεθούμε στον τοπικό DB server (localhost)
// και στη βάση δεδομένων my_db.
// Αν η σύνδεση αποτύχει η εφαρμογή τερματίζει εμφανίζοντας
// το μήνυμα λάθους αλλιώς εμφανίζει μήνυμα επιτυχίας
$dbServer = "localhost";
$user = "admin";
$pswd = "fail";
$db = "my_db";

try {
    $conn = mysqli_connect($dbServer, $user, $pswd, $db);
} catch (Exception $e) {
    die("Error connecting to database: ".$e->getMessage());
}
echo "Connect to $db successfully";
// Εδώ βάζουμε τον υπόλοιπο κώδικα...
?>
```

94. Πού χρησιμοποιούμε τις μεταβλητές συνόδου στην PHP (PHP session variables); Δώστε δύο παραδείγματα.

Τα session variables χρησιμοποιούνται για να αποθηκεύσουν πληροφορίες που πρέπει να παραμείνουν διαθέσιμες κατά τη διάρκεια σύνδεσης ενός χρήστη. Είναι χρήσιμα για να παρακολουθούμε την κατάσταση ενός χρήστη ή όταν θέλουμε να μοιραζόμαστε δεδομένα σε πολλαπλές σελίδες της εφαρμογής.

Για να δηλώσουμε μια συνεδρία χρησιμοποιούμε το session_start(). Τα session variables δηλώνονται σαν \$_SESSION["variable name"]. Για παράδειγμα έχουμε ένα ιστότοπο που σε χρήστες admin επιτρέπει να βλέπουν κάποιες εμπιστευτικές σελίδες. Οι υπόλοιποι χρήστες δεν έχουν πρόσβαση.

Σελίδα ελέγχου πρόσβασης χρήστη:

```
<?php
// Αν ο χρήστης δώσει αυτά τα στοιχεία έχει πρόσβαση σε επιπλέον
// σελίδες. Αλλιώς η πρόσβαση απαγορεύεται
$username = "admin";
$password = "123456";

session_start();
// Ο χρήστης έδωσε τα στοιχεία του σε κάποια άλλη σελίδα και
// τώρα κάνουμε έλεγχο των στοιχείων
if ($_POST["user"] == $username && $_POST["pswd"] == $password) {
    $_SESSION["username"] = $username;
    echo "welcome $username. You have access to all pages.";
} else {
    $_SESSION["username"] = "guest";
    echo "welcome guest. You have restricted access.";
}
?>
```

Σε άλλη σελίδα ελέγχουμε αν ο χρήστης είναι admin και άρα έχει δικαίωμα πρόσβασης:

```
<?php
session_start();
if (!isset($_SESSION["username"]) ||
    $_SESSION["username"] != "admin") {
    echo "You are not allowed to view this page";
    exit();
}
echo "welcome ".$_SESSION["username"]." You have full access";
// Rest of page
?>
```

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(int $num1, int $num2) {
    echo "$num1 + $num2 = ".$num1 + $num2;
}
?>
```


96. Φτιάξτε ένα script στην PHP που θα διαβάσει την τιμή της μεταβλητής συνόδου (session) "logged" και εάν αυτή δεν είναι αληθής να κάνει ανακατεύθυνση (redirect) στην σελίδα login.html.

```
<?php
session_start();
if (!isset($_SESSION["logged"]) || !$_SESSION["logged"]) {
    header("Location: login.html");
    exit();    // Stop execution
}
echo "welcome to this page";
// Υπόλοιπος κώδικας...
?>
```

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(int $num1, int $num2) {
    echo "$num1 + $num2 = ".$num1 + $num2."<br>";
}
sum(5, 10); // Εμφανίζει 5 + 10 = 15
sum(7, 13); // Εμφανίζει 7 + 13 = 20
sum(2, 4);  // Εμφανίζει 2 + 4 = 6
?>
```

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
$query = "SELECT * FROM subscribers;";
$rows = mysqli_query($conn, $query);    // $conn είναι η σύνδεση με DB

// Διατρέχουμε τις εγγραφές με την fetch_assoc()
while ($row = mysqli_fetch_assoc($rows)) {
    $to = $row["email"];
    $user = $row["username"];
    $pswd = $row["password"];
    $body = "Your username is $user and your password is $pswd";
    mail($to, "Login info", $body);
}
mysqli_free_result($rows);    // Ελευθέρωσε τη μνήμη
mysqli_close($conn);         // Κλείσε τη σύνδεση με DB
?>
```

99. Τι εννοούμε με τον όρο static mesh στην αρχιτεκτονική των Game Engines (Unreal Engine , Source και Unity);

Ο όρος static mesh αναφέρεται στα μοντέλα που δεν μπορούν να κινηθούν ή να παραμορφωθούν. Είναι στατικά. Για παράδειγμα ένας τοίχος που δεν γκρεμίζεται, ένα βουνό κλπ.

100. Τι είναι τα collision boxes στα Game Engines; Τι εξυπηρετούν και με ποιο σκεπτικό τα κατασκευάζουμε;

Τα collision boxes είναι απλά γεωμετρικά σχήματα (πχ ορθογώνια, σφαίρες, κύβοι, κύλινδροι) που χρησιμοποιούνται για τον έλεγχο συγκρούσεων μεταξύ αντικειμένων στον κόσμο του παιχνιδιού. Δεν είναι ορατά. Περιβάλλουν τα αντικείμενα και καθορίζουν αν αυτά επικαλύπτονται (συγκρούονται). Με τα collision boxes μπορούμε για παράδειγμα να δούμε αν ένας παίκτης χτύπησε στον τοίχο, μια σφαίρα πέτυχε τον εχθρό κλπ.

101. Αναφέρετε ποιες τεχνικές φωτισμού χρησιμοποιούνται στα video games και ποια τα χαρακτηριστικά τους.

Οι τεχνικές φωτισμού μπορούν να χωριστούν σε δύο κατηγορίες: στατικές (static) και δυναμικές (dynamic).

Στατικός φωτισμός (Static Lighting): Χρησιμοποιείται για αντικείμενα που δεν επηρεάζονται από την κίνηση του φωτός όπως τοίχοι και πατώματα. Αποδοτικός καθώς δεν απαιτείται επεξεργασία σε πραγματικό χρόνο.

Δυναμικός φωτισμός (Dynamic Lighting): Αυτός ο φωτισμός υπολογίζεται συνεχώς με βάση την κίνηση των πηγών φωτός και αντικειμένων.

Φωτισμός Περιβάλλοντος (Global Illumination): Είναι ο φυσικός φωτισμός για εσωτερικούς και εξωτερικούς χώρους. Περιλαμβάνει τις αντανάκλασεις των αντικειμένων.

Directional Lighting: Προσομοιώνει φωτισμό που προέρχεται από μακρινή απόσταση και φωτίζει μεγάλες περιοχές, πχ ο ήλιος.

Point Lighting: Χρησιμοποιείται για μικρότερης έκτασης φωτισμό όπου το φως προέρχεται από μία πηγή, πχ. μία λάμπα σε ένα δωμάτιο.

Spot Lighting: Μοιάζει με το point lighting αλλά το εύρος είναι ακόμη πιο περιορισμένο, πχ ένας φακός.

102. Τι είναι το bill boarding και ποια τα πλεονεκτήματά του στην κατασκευή 3D games;

Είναι μία τεχνική για την απεικόνιση 2D αντικειμένων ώστε να φαίνονται τρισδιάστατα σε ένα 3D game. Ουσιαστικά το billboard είναι μια επίπεδη επιφάνεια με την εικόνα του αντικειμένου που προσανατολίζεται πάντα προς την κάμερα. Με τον τρόπο αυτό δημιουργείται η ψευδαίσθηση 3D χωρίς να απαιτούνται πολύπλοκοι υπολογισμοί.

103. Πολλές φορές στους κατασκευαστές παιχνιδιών (πχ Unity) εφαρμόζεται το Level of Detail (LOD), περιγράψτε τι είναι και ποια είναι η χρησιμότητά του.

Το LOD είναι μια τεχνική που απεικονίζει τα μοντέλα με διαφορετική λεπτομέρεια ανάλογα με την απόσταση τους από την κάμερα. Έτσι για μακρινότερες αποστάσεις χρησιμοποιούνται λιγότερα πολύγωνα και για κοντινές περισσότερα. Αυτό μειώνει την απαιτούμενη επεξεργαστική ισχύ χωρίς η απώλεια της λεπτομέρειας να γίνεται αντιληπτή από το χρήστη.

104. Να αναφερθούν συνοπτικά οι βασικές κλάσεις του Unreal Editor και ποια η λογική της χρησιμοποίησής τους.

Object: Είναι η βασική κλάση του παιχνιδιού και από αυτήν κληρονομούνται τα πάντα, πχ. τα αντικείμενα που υπάρχουν στη σκηνή, οι χαρακτήρες, οι κανόνες κλπ.

Actor: Είναι παιδί της Object και περιλαμβάνει τα αντικείμενα της σκηνής, πχ τοίχους, πόρτες, δέντρα κλπ.

Pawn: Είναι παιδί της Actor και περιλαμβάνει όλους τους χαρακτήρες, δηλαδή τον χαρακτήρα του παίκτη, δράκους, εχθρούς κλπ.

Υπάρχουν βέβαια και άλλες κλάσεις όπως Controller για το χειρισμό των χαρακτήρων της Pawn, GameState που διαχειρίζεται την κατάσταση του παιχνιδιού με βάση του κανόνες κλπ.

105. Περιγράψτε τη κλάση actor (στην UnrealScript), αναφέρατε τη χρησιμότητά της.

Η κλάση Actor είναι παιδί της Object και διαχειρίζεται όλα τα αντικείμενα που βρίσκονται στη σκηνή, όπως τοίχους, πατώματα, φωτισμό, δέντρα κλπ. Διαχειρίζεται τον προσανατολισμό και τις συντεταγμένες όλων των αντικειμένων του κόσμου. Οι υπόλοιπες κλάσεις κληρονομούν από αυτή, για παράδειγμα η Pawn που διαχειρίζεται τους χαρακτήρες του παιχνιδιού.

106. Περιγράψτε τη κλάση pawn (στην UnrealScript), αναφέρατε τη χρησιμότητά της.

Η κλάση pawn είναι παιδί της κλάσης actor. Διαχειρίζεται όλους τους χαρακτήρες του παιχνιδιού καθώς και την τεχνητή νοημοσύνη. Για παράδειγμα χειρίζεται τους χαρακτήρες του παίκτη, τέρατα, εχθρούς κλπ.

107. Να αναφερθούν οι τύποι μεταβλητών που υποστηρίζονται από την Unrealscript καθώς και οι περιγραφές τους.

byte: Ακέραιοι 8-bit με τιμές 0-255.

int: Ακέραιοι αριθμοί.

float: Αριθμοί κινητής υποδιαστολής.

bool: Λογικές μεταβλητές. Δέχονται τιμές true/false.

string: Σειρές αλφαριθμητικών χαρακτήρων.

name: Αποθηκεύουν ονόματα συναρτήσεων, αντικειμένων κλπ για ταχύτερη προσπέλαση.

Πέραν αυτών υποστηρίζονται και πιο σύνθετες δομές δεδομένων όπως array, vector, struct, enum, object κλπ.

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.

Idle: Το όπλο είναι αδρανές και δεν εκτελεί κάποια ενέργεια.

Firing: Το όπλο πυροβολεί.

Reloading: Ενεργοποιείται όταν το όπλο πρέπει να γεμίσει ξανά με πυρομαχικά.

AltFiring: Το ίδιο με το firing αλλά για δεύτερη λειτουργία του όπλου.

Jammed: Κατάσταση στην οποία το όπλο έχει μπλοκάρει και απαιτείται κάποια ενέργεια.

Overheating: Το όπλο έχει υπερθερμανθεί από την πολλή χρήση.

109. Τι σημαίνουν οι όροι Modeling, Rendering και Animation;

Το Modeling είναι η κατασκευή των αντικειμένων που υπάρχουν στη σκηνή. Δηλαδή ο ορισμός των κορυφών κάθε αντικειμένου, τα γεωμετρικά σχήματα που θα ενώνουν τις κορυφές κλπ.

Το Rendering συμπληρώνει τη μοντελοποίηση με χρώματα, φωτισμό κλπ και κάνει τους απαραίτητους υπολογισμούς ώστε να απεικονιστεί η τρισδιάστατη εικόνα στην οθόνη.

Το Animation είναι η δημιουργία κίνησης είτε των αντικειμένων είτε της κάμερας ή λοιπών στοιχείων που βρίσκονται στη σκηνή.

110. Τι είναι φωτοαπόδοση και ποιοι παράγοντες την επηρεάζουν;

Η φωτοαπόδοση (rendering) είναι η διαδικασία υπολογισμών για να παρουσιαστεί ένα 3D μοντέλο στην οθόνη. Το rendering συμπληρώνει τα γεωμετρικά σχήματα με χρώμα, φωτισμό κλπ και τα μετατρέπει στην bitmap εικόνα που θα εμφανιστεί στην οθόνη.

Οι παράγοντες που επηρεάζουν την φωτοαπόδοση είναι το πλήθος των χρωμάτων, οι υφές (textures), τα υλικά (materials), ο φωτισμός, οι σκιάσεις και η υπολογιστική ισχύς.

111. Αναφέρετε τα στάδια δημιουργίας κινούμενης φωτορεαλιστικής απεικόνισης ενός θέματος πολυμέσων με τη χρήση κάποιου εργαλείου τρισδιάστατης σχεδίασης και κίνησης (π.χ. 3d Studio).

Τα στάδια δημιουργίας είναι:

Modeling: Η μοντελοποίηση περιλαμβάνει τη δημιουργία των τρισδιάστατων γεωμετρικών σχημάτων που θα απαρτίζουν τη σκηνή.

Animation: Δίνουμε κίνηση στα μοντέλα προσδιορίζοντας τη θέση τους στις διάφορες χρονικές στιγμές.

Υφές και υλικά: Προσδιορίζουμε τα υλικά (materials) και τις υφές (textures) από τις οποίες θα αποτελούνται τα αντικείμενα.

Φωτισμός: Καθορίζουμε τις πηγές φωτός που θα υπάρχουν στη σκηνή.

Κάμερα: Ορίζουμε τη θέση και κίνηση της κάμερας

Rendering: Μετατρέπει τα τρισδιάστατα μοντέλα σε bitmap εικόνες ώστε να απεικονισθούν στην οθόνη. Λαμβάνονται υπόψη οι υφές, τα υλικά, ο φωτισμός και όλα τα υπόλοιπα στοιχεία που έχουμε χρησιμοποιήσει ώστε να υπάρχει φωτορεαλιστική απεικόνιση. Ανάλογα με την πολυπλοκότητα των μοντέλων απαιτείται περισσότερη υπολογιστική ισχύς.

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.

Οι τεχνικές φωτισμού και σκίασης δημιουργούν πηγές φωτός, αντανakλάσεις, χρώματα και σκιές για να έχουμε φωτορεαλιστική απεικόνιση τρισδιάστατων μοντέλων στην οθόνη.

Βασικές τεχνικές φωτισμού:

Φωτισμός Περιβάλλοντος (Global Illumination): Είναι ο φυσικός φωτισμός για εσωτερικούς και εξωτερικούς χώρους. Περιλαμβάνει τις αντανakλάσεις των αντικειμένων.

Directional Lighting: Προσομοιώνει φωτισμό που προέρχεται από μακρινή απόσταση και φωτίζει μεγάλες περιοχές, πχ ο ήλιος.

Point Lighting: Χρησιμοποιείται για μικρότερης έκτασης φωτισμό όπου το φως προέρχεται από μία πηγή, πχ. μία λάμπα σε ένα δωμάτιο.

Spot Lighting: Μοιάζει με το point lighting αλλά το εύρος είναι ακόμη πιο περιορισμένο, πχ ένας φακός.

Βασικές τεχνικές σκίασης:

Flat shading: Κάθε επιφάνεια έχει ένα ενιαίο χρώμα σκίασης. Χρησιμοποιείται για απλές και γρήγορες απεικονίσεις.

Phong shading: Η σκίαση κάθε σημείου μιας επιφάνειας καθορίζεται από τις πηγές φωτός και τη γωνία της κάμερας. Χρησιμοποιείται για πιο φωτορεαλιστικές απεικονίσεις.

113. Ποιες τεχνικές γνωρίζετε για την απόδοση υλικών (materials & textures) πάνω στα τρισδιάστατα μοντέλα;

Diffuse mapping: Προσδίδει το βασικό χρώμα και υφή σε ένα μοντέλο. Είναι η πιο κοινή τεχνική και εφαρμόζεται σε όλες τις επιφάνειες.

Texture mapping: Εφαρμόζει εικόνες στις επιφάνειες των μοντέλων για να προσθέσει λεπτομέρειες και χρώματα.

Normal mapping: Προσθέτει λεπτομέρειες στην επιφάνεια του μοντέλου για να δημιουργήσει ανωμαλίες και να προσδώσει βάθος.

Specular mapping: Καθορίζει τις περιοχές που δημιουργούν αντανakλάσεις

Displacement mapping: Αλλάζει την πραγματική γεωμετρία του μοντέλου για να δημιουργήσει πιο ρεαλιστικές απεικονίσεις.

114. Τι είναι το Normal Mapping και ποια η χρησιμότητά του; Περιγράψτε αναλυτικά τον τρόπο λειτουργίας του.

Το normal mapping χρησιμοποιείται για να προσδώσει την αίσθηση μικρών ανωμαλιών και περισσότερων λεπτομερειών στις επιφάνειες ενός μοντέλου.

Ένα normal map είναι μια εικόνα που περιέχει πληροφορίες για την κατεύθυνση των κανονικών (normals) διανυσμάτων της επιφάνειας. Αυτές οι πληροφορίες αποθηκεύονται στα χρώματα των pixels της εικόνας. Κατά το rendering τα κανονικά διανύσματα που έχουν τροποποιηθεί χρησιμοποιούνται για τον υπολογισμό του φωτισμού και τη δημιουργία μικρών ανωμαλιών. Αυτό δημιουργεί πιο λεπτομερή και ρεαλιστική απεικόνιση της εικόνας.

115. Τι είναι οι τεχνικές σχεδίασης και φωτισμού; Περιγράψτε περιληπτικά τρεις τεχνικές δημιουργίας της τελικής εικόνας (rendering).

Οι τεχνικές σχεδίασης και φωτισμού είναι οι μέθοδοι που χρησιμοποιούνται για να περιγράψουν πως το φως επιδρά στις επιφάνειες ενός μοντέλου. Είναι βασικές για την ρεαλιστική αναπαράσταση των εικόνων. Αυτές οι τεχνικές είναι άρρηκτα συνδεδεμένες με το rendering.

Rasterization: Είναι η πιο κοινή και γρήγορη τεχνική. Μετατρέπει τα τρισδιάστατα αντικείμενα σε εικονοστοιχεία υπολογίζοντας το χρώμα που θα έχει το κάθε pixel στην τελική εικόνα.

Χρησιμοποιείται ευρέως σε εφαρμογές πραγματικού χρόνου, όπως βιντεοπαιχνίδια.

Ray Tracing: Υπολογίζει τον τρόπο που επιδρούν οι ακτίνες φωτός με τα αντικείμενα. Προσφέρει πολύ πιο ρεαλιστικές εικόνες από το rasterization αλλά με αυξημένο υπολογιστικό κόστος.

Global Illumination: Υπολογίζει πως αλληλεπιδρά το φως με τα αντικείμενα συμπεριλαμβανομένων των αντανακλάσεων και της διάχυσης. Αποδίδει τις πιο φωτορεαλιστικές εικόνες αλλά με μεγάλο υπολογιστικό κόστος.

116. Γιατί είναι απαραίτητος ο φωτισμός της σκηνής; Περιγράψτε τρεις τρόπους με τους οποίους μπορεί να επιτευχθεί το παραπάνω.

Ο φωτισμός μας δίνει τη σωστή αίσθηση χρωμάτων, σκίασης και βάθους για αυτό είναι απαραίτητος για τη φωτορεαλιστική απεικόνιση της σκηνής.

Για να δημιουργήσουμε τον απαραίτητο φωτισμό μπορούμε να χρησιμοποιήσουμε:

Φωτισμός Περιβάλλοντος (Global Illumination): Είναι ο φυσικός φωτισμός για εσωτερικούς και εξωτερικούς χώρους. Περιλαμβάνει τις αντανακλάσεις των αντικειμένων.

Directional Lighting: Προσομοιώνει φωτισμό που προέρχεται από μακρινή απόσταση και φωτίζει μεγάλες περιοχές, πχ ο ήλιος.

Point Lighting: Χρησιμοποιείται για μικρότερης έκτασης φωτισμό όπου το φως προέρχεται από μία πηγή, πχ. μία λάμπα σε ένα δωμάτιο.

117. Τι είναι τα function curves (στα προγράμματα τρισδιάστατης απεικόνισης) και που χρησιμεύουν;

Τα function curves γνωστά και ως animation curves είναι γραφικές αναπαραστάσεις των μεταβολών των ιδιοτήτων ενός αντικειμένου (πχ. θέση, περιστροφή, διαφάνεια κλπ). Ένα function curve είναι μια καμπύλη που δείχνει πως αλλάζει η ιδιότητα ενός αντικειμένου στη διάρκεια του χρόνου.

Μπορούμε να επεξεργαστούμε αυτή την καμπύλη και να αποδώσουμε για παράδειγμα πιο γρήγορη ή πιο αργή κίνηση σε ένα αντικείμενο.

118. Τι είναι το UV Mapping;

Το UV mapping είναι μια τεχνική για την εφαρμογή υφών (textures) σε τρισδιάστατα αντικείμενα. Το όνομα προέρχεται από τις μεταβλητές U και V που αντιπροσωπεύουν τις διαστάσεις συντεταγμένες στο χώρο που θα προβληθεί η εικόνα (καθώς οι X, Y, Z είναι δεσμευμένες για την τρισδιάστατη αναπαράσταση του αντικειμένου).

119. Τι είναι το Diffuse Map και ποια η διαφορά του από το Texture Map;

Diffuse mapping: Προσδίδει το βασικό χρώμα και υφή σε ένα μοντέλο. Είναι η πιο κοινή τεχνική και εφαρμόζεται σε όλες τις επιφάνειες.

Texture mapping: Εφαρμόζει εικόνες στις επιφάνειες των μοντέλων για να προσθέσει λεπτομέρειες και χρώματα.

120. Τι είναι το Bump Map;

Το bump map είναι μια τεχνική που δημιουργεί την ψευδαίσθηση ότι η επιφάνεια έχει βαθουλώματα ή εξογκώματα. Είναι το κανάλι της εικόνας που περιγράφει σε τόνους του γκρι πως συμπεριφέρεται το κάθε pixel της εικόνας σε σχέση με το φως. Συνήθως μαύρο σημαίνει βαθούλωμα και λευκό εξόγκωμα.

121. Τι είναι το Opacity Map;

Το opacity map είναι μια τεχνική που καθορίζει τη διαφάνεια ή αδιαφάνεια σε διαφορετικές περιοχές της επιφάνειας ενός αντικειμένου. Βασίζεται σε μια ασπρόμαυρη εικόνα (αποχρώσεις του γκρι) όπου το κάθε pixel αντιπροσωπεύει το βαθμό διαφάνειας (μαύρο για πλήρη διαφάνεια, λευκό για πλήρη αδιαφάνεια).

122. Τι είναι η τεχνική extrude και πώς μπορεί να αξιοποιηθεί;

Η τεχνική extrude δημιουργεί τρισδιάστατα γεωμετρικά σχήματα από δισδιάστατα. Η extrude εξωθεί το δισδιάστατο σχήμα προς μια κατεύθυνση δημιουργώντας τρισδιάστατη γεωμετρία. Για παράδειγμα αν ανυψώσουμε με την extrude ένα κύκλο θα δημιουργηθεί ένας κύλινδρος.

123. Στην τεχνική του Box Modeling, ποια η διαφορά μεταξύ του Convert to editable poly και πότε του convert to editable mesh; Πότε χρησιμοποιείται το δεύτερο;

Τα αντικείμενα editable mesh αποτελούνται από συνδεδεμένες τριγωνικές επιφάνειες ενώ τα editable poly αποτελούνται από συνδεδεμένα πολύγωνα και προσφέρουν πολύ περισσότερες δυνατότητες επεξεργασίας. Επιλέγουμε convert to editable mesh όταν έχουμε τελειώσει με την επεξεργασία του μοντέλου και θέλουμε να ελευθερώσουμε μνήμη ή για συμβατότητα με παλαιότερες μηχανές γραφικών που δεν υποστηρίζουν editable poly.

124. Τι είναι το Specular Map;

Το specular map είναι μια ασπρόμαυρη εικόνα (αποχρώσεις του γκρι) που καθορίζει τον τρόπο που ανακλούν το φως τα σημεία της επιφάνειας ενός αντικειμένου. Τα λευκά pixel υποδηλώνουν πλήρη ανάκλαση, τα μαύρα καθόλου και οι ενδιάμεσες τιμές το βαθμό ανάκλασης.

125. Τι είναι το modifier Lathe; Τι μπορεί να σχεδιαστεί χρησιμοποιώντας το;

Το modifier Lathe περιστρέφει ένα spline (δισδιάστατο σχήμα) γύρω από κάποιον άξονα δημιουργώντας έτσι ένα τρισδιάστατο γεωμετρικό σχήμα. Με τρόπο αυτό μπορούμε να σχεδιάσουμε ποτήρια, βάζα, μπουκάλια κλπ.

126. Τι είναι το Displacement Mapping;

Το displacement map είναι ένας χάρτης υφής (texture) που καθορίζει την ποσότητα μετατόπισης κάθε σημείου της επιφάνειας ενός αντικειμένου. Σε αντίθεση με τα bump map και normal map το displacement αλλάζει την γεωμετρία του μοντέλου. Προσφέρει εξαιρετική φωτορεαλιστική απεικόνιση και λεπτομέρεια αλλά με αντίστοιχο υπολογιστικό κόστος.

Μπορεί να χρησιμοποιηθεί για διάφορα εφέ, όπως δημιουργία κυμάτων ή κίνηση της επιφάνειας ενός υγρού.

127. Δημιουργείστε ένα παράθυρο 300X400 (window 300X400) σε JAVA, το οποίο θα μπορεί να περιλαμβάνει ένα υπό δημιουργία παιχνίδι.

```
import javax.swing.*;
```

```
// Δημιουργία ενός παράθυρου 300X400 κεντραρισμένου στην οθόνη
```



```
public class Game extends JFrame {
    Game() {
        setTitle("Sample Game");
        setSize(300, 400);
        setLocationRelativeTo(null);
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {
        new Game();
    }
}
```

128. Περιγράψτε τη χρησιμότητα των μεταβλητών κατά τη δημιουργία παιχνιδιών σε κώδικα JAVA (JAVA Game Variables).

Για τη δημιουργία παιχνιδιών σε java, οι μεταβλητές είναι απαραίτητες για τον έλεγχο της κατάστασης του παιχνιδιού.

Τα game variables μπορεί να κρατούν το σκορ του παιχνιδιού, τις ζωές του παίκτη, να ελέγχουν την κίνηση του παίκτη ή των αντιπάλων, αν συγκρούονται μεταξύ τους, αν τελείωσε το επίπεδο ή το παιχνίδι κλπ.

129. Να γραφεί απλό πρόγραμμα με κώδικα σε Java, που να τυπώνει ως αποτέλεσμα στον χρήστη τις λέξεις "I love JAVA Game Programming"

```
import javax.swing.*;

public class Game {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Game");
        JLabel label = new JLabel("I love JAVA Game Programming");
        JPanel panel = new JPanel();
        panel.add(label);
        frame.add(panel);
        frame.setSize(400, 400);
        frame.setLocationRelativeTo(null);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

130. Να αναφέρετε τη χρησιμότητα των μεθόδων (methods) στο JAVA Game Programming.

Οι μέθοδοι είναι εξαιρετικά χρήσιμες για διάφορους λόγους:

α) DRY (Don't Repeat Yourself): Αποτρέπουν τη δημιουργία επαναλαμβανόμενου κώδικα για να επιλύσει το ίδιο πρόβλημα. Μπορούμε να δημιουργήσουμε μια μέθοδο που επιλύει το πρόβλημα και να την καλούμε από διάφορα σημεία του παιχνιδιού. Για παράδειγμα ανανέωση του σκορ ή υπολογισμός ζωνών.

- β) Clean Code: Μία μέθοδος μπορεί να καλέσει άλλες μεθόδους. Με την τεχνική διαίρει και βασίλευε μπορούμε να σπάσουμε μια μέθοδο σε περισσότερες ώστε κάθε μία να επιλύει μόνο ένα πρόβλημα. Με τον τρόπο αυτό δημιουργούμε πιο ευανάγνωστο κώδικα.
- γ) Ενθυλάκωση: Οι μεταβλητές και ο τρόπος που επιλύει ένα πρόβλημα μια μέθοδος δεν είναι ορατές έξω από αυτήν. Αυτό προσφέρει ασφάλεια και ελαχιστοποίηση λαθών.
- δ) Επεκτασιμότητα: Μπορούν να προστεθούν επιπλέον δυνατότητες στο παιχνίδι με την προσθήκη επιπλέον μεθόδων χωρίς να αλλάξει ο υπάρχων κώδικας.

131. Ένα παιχνίδι που έχει δημιουργηθεί σε JAVA, συνήθως περιέχει: 1) Frame 2) Canvas 3) GameLoop 4) Rendering method 5) Update method 6) Mouse and Key input Εξηγείστε συνοπτικά τις έννοιες που αναφέρθηκαν, καθώς και τη χρησιμότητά τους.

Frame

Είναι το κύριο παράθυρο του παιχνιδιού. Περιέχει όλα τα γραφικά στοιχεία (όπως το canvas). Χρησιμοποιείται για την εμφάνιση του παιχνιδιού στην οθόνη του χρήστη.

Canvas

Είναι μια επιφάνεια σχεδίασης μέσα στο frame. Χρησιμοποιείται για την εμφάνιση (rendering) των γραφικών του παιχνιδιού. Όλα τα γραφικά στοιχεία (χαρακτήρες και αντικείμενα) σχεδιάζονται επάνω στο canvas.

Game Loop

Είναι ο κύριος βρόγχος του παιχνιδιού και εκτελείται συνέχεια μέχρι να τερματίσει το παιχνίδι. Είναι υπεύθυνο για την ενημέρωση της κατάστασης του παιχνιδιού και των γραφικών (για αυτό καλεί τις μεθόδους render και update).

Rendering method

Αυτή η μέθοδος είναι υπεύθυνη για τον σχεδιασμό των χαρακτήρων και αντικειμένων επάνω στο canvas. Καλείται σε κάθε καρέ του Game Loop

Update method

Αυτή η μέθοδος ενημερώνει την κατάσταση του παιχνιδιού, την κίνηση των χαρακτήρων, την ανίχνευση συγκρούσεων κλπ. Καλείται επίσης σε κάθε καρέ του Game Loop.

Mouse & key input

Χρησιμοποιούμε μεθόδους που χειρίζονται την είσοδο από το πληκτρολόγιο ή το ποντίκι για να καταγράψουν τις ενέργειες του παίκτη. Είναι απαραίτητες για την αλληλεπίδραση με τον χρήστη.

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.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Game {
    public static void main(String[] args) {
        JFrame frame = new JFrame();
        Ball ball = new Ball();
```

```
frame.add(ball);
frame.setSize(500, 500);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

}

class Ball extends JPanel implements ActionListener {
    private int x = 0, y = 0; // Το κέντρο της μπάλας
    private int speedX = 5, speedY = 5; // Πόσο γρήγορα κινείται
    private int ballSize = 20;

    Ball() {
        // Καλώ τον ActionListener κάθε 10ms
        Timer timer = new Timer(10, this);
        timer.start();
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        x += speedX;
        y += speedY;

        // Αν η μπάλα έφτασε στα άκρα αντέστρεψε τη φορά της
        if (x + ballSize > getWidth() || x < 0) {
            speedX *= -1;
        }
        if (y + ballSize > getHeight() || y < 0) {
            speedY *= -1;
        }
        repaint();
    }

    // Ζωγράφισε ένα κύκλο σε κόκκινο χρώμα
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.red);
        g.fillOval(x, y, ballSize, ballSize);
    }
}
```

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.

Τύποι πολυμέσων:

- Εικόνες: GIF, PNG, JPG, BMP, SVG
- Βίντεο: MP4
- Ήχος: MP3
- Διαδραστικά πολυμέσα: Flash, Unity web player, WebGL

Η εγκατάσταση add-ons είναι απαραίτητη για πολυμέσα που δεν υποστηρίζονται εγγενώς από όλους τους browsers. Με την ευρεία χρήση της HTML5 η ανάγκη για εγκατάσταση πρόσθετων έχει μειωθεί δραματικά καθώς οι browsers υποστηρίζουν τους περισσότερους τύπους πολυμέσων. Ωστόσο η χρήση πρόσθετων είναι απαραίτητη για παλαιότερους browsers και λειτουργικά συστήματα ώστε να αναπαραχθούν κάποιοι τύποι πολυμέσων (πχ Flash, QuickTime κλπ).

134. Τι είναι τα keywords, γιατί είναι απαραίτητη η χρησιμοποίηση τους στην δημιουργία μιας ιστοσελίδας και με ποιον τρόπο γίνεται η σωστή επιλογής τους;

Τα keywords (λέξεις-κλειδιά) είναι λέξεις ή φράσεις που χρησιμοποιούνται για να περιγράψουν το περιεχόμενο μιας ιστοσελίδας. Χρησιμοποιούνται από τις μηχανές αναζήτησης ώστε να κατανοήσουν το περιεχόμενο της σελίδας και να την κατατάξουν στα αποτελέσματα αναζήτησης όταν οι χρήστες ζητούν σχετικές πληροφορίες.

Η σωστή χρήση λέξεων-κλειδιών βελτιώνει το SEO (Search Engine Optimization) και κατατάσσει τη σελίδα υψηλότερα στα αποτελέσματα αναζήτησης. Η σωστή επιλογή μπορεί να γίνει με διάφορα εργαλεία που υπάρχουν στο Internet (αρκετά δωρεάν) ή μετά από έρευνα για τις συνήθεις φράσεις που χρησιμοποιούν οι χρήστες κατά την αναζήτηση σελίδων παρόμοιων με τη δική μας.

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;

Το SEO είναι η διαδικασία βελτιστοποίησης μιας ιστοσελίδας ώστε να εμφανίζεται υψηλότερα στα αποτελέσματα μιας μηχανής αναζήτησης όπως το Google. Όσο υψηλότερα εμφανίζεται η σελίδα στα αποτελέσματα τόσο πιθανότερο είναι να την δει και να την επισκεφτεί ένας χρήστης.

Το SEO επιτυγχάνεται με διάφορες τεχνικές που βοηθούν τις μηχανές αναζήτησης να κατανοήσουν το περιεχόμενο της ιστοσελίδας και να την κατατάξουν στα αποτελέσματα σχετικών αναζητήσεων. Με το SEO αυξάνεται η ορατότητα και η επισκεψιμότητα της ιστοσελίδας ενώ βελτιώνεται και η εμπειρία του χρήστη καθώς το περιεχόμενο είναι πιο κατανοητό όχι μόνο για τις μηχανές αναζήτησης αλλά και για τους χρήστες.

136. Τί είναι οι εφαρμογές CMS (Content Management systems); Αναφέρατε δύο τουλάχιστον εργαλεία CMS.

Τα Συστήματα Διαχείρισης Περιεχομένου (CMS) είναι εφαρμογές που επιτρέπουν την εύκολη δημιουργία, διαχείριση και τροποποίηση ενός ιστότοπου χωρίς εξειδικευμένες γνώσεις προγραμματισμού. Προσφέρουν ένα φιλικό περιβάλλον χειρισμού όπου οι χρήστες μπορούν να δημιουργήσουν πλούσιο περιεχόμενο με κείμενο και πολυμέσα.

Δύο δημοφιλείς εφαρμογές CMS είναι τα WordPress και Drupal.

137. Ποια η χρησιμότητα του RSS και πώς επιτυγχάνεται η ενσωμάτωση του σε μια ιστοσελίδα τύπου CMS;

Το RSS είναι μια τεχνολογία που επιτρέπει στους χρήστες να λαμβάνουν αυτόματα ενημερώσεις από ιστοσελίδες που παρακολουθούν χωρίς να χρειάζεται να τις επισκέπτονται συνέχεια. Είναι ιδιαίτερα χρήσιμο για ιστοσελίδες που ανανεώνουν συχνά το περιεχόμενο τους όπως ειδησεογραφικά sites.

Αρκετά CMS όπως το WordPress έχουν ενσωματωμένη υποστήριξη RSS και μπορεί να ενεργοποιηθεί από τις ρυθμίσεις. Υπάρχουν όμως και αρκετά plugins που βελτιώνουν τη λειτουργία του RSS.

138. Πώς ελέγχουμε την επισκεψιμότητα μιας ιστοσελίδας που δημιουργήθηκε με CMS;

Πολλά CMS όπως το WordPress περιέχουν ενσωματωμένα εργαλεία ή πρόσθετα για την παρακολούθηση της επισκεψιμότητας.

Το Google Analytics είναι ένα από τα δημοφιλέστερα εργαλεία και μπορεί να εγκατασταθεί εύκολα σε ένα CMS μέσω plugin. Προσφέρει πληθώρα πληροφοριών για την επισκεψιμότητα των σελίδων, τον χρόνο παραμονής των χρηστών και πολλά άλλα.

139. Αναφέρετε τις διαφορές μεταξύ component, module και plugin.

Component

Είναι η κεντρική μονάδα λειτουργικότητας ενός CMS. Τα components εμφανίζονται ως κύριο περιεχόμενο σε μια ιστοσελίδα.

Module

Είναι μικρότερα κομμάτια λειτουργικότητας και μπορούν να τοποθετηθούν σε διάφορες θέσεις μιας σελίδας. Για παράδειγμα ένα module μπορεί να είναι ένα μενού πλοήγησης ή ένα πλαίσιο αναζήτησης.

Plugin

Είναι μικρά προγράμματα που επεκτείνουν τη λειτουργικότητα ενός CMS. Τα plugins μπορεί να προσθέσουν νέες δυνατότητες ή να τροποποιήσουν υπάρχουσες λειτουργίες χωρίς να επηρεάσουν τον πυρήνα του CMS. Ένα plugin για παράδειγμα θα μπορούσε να προσθέσει λειτουργίες για μαζική αποστολή emails.

140. Τι κάνουν τα εργαλεία τύπου filezilla?

Τα εργαλεία τύπου filezilla χρησιμοποιούν το πρωτόκολλο FTP (File Transfer Protocol) για τη μεταφορά αρχείων από ένα τοπικό υπολογιστή προς ένα απομακρυσμένο server και το αντίστροφο. Μπορούν επίσης να δημιουργήσουν, μετακινήσουν, αντιγράψουν, μετονομάσουν ή διαγράψουν αρχεία στον απομακρυσμένο server.

141. Ποιοι τύποι χρηστών υπάρχουν σε μια ιστοσελίδα και ποια τυπικά δικαιώματα έχει ο καθένας;

Σε μία ιστοσελίδα που έχει δημιουργηθεί με CMS υπάρχουν τα εξής δικαιώματα:

Διαχειριστής (Administrator): Έχει πλήρη δικαιώματα και πρόσβαση σε όλο το περιεχόμενο. Μπορεί να αφαιρεί ή να προσθέτει χρήστες, να εγκαθιστά ή να αφαιρεί πρόσθετα και να τροποποιεί το περιεχόμενο ή τις ρυθμίσεις του CMS.

Συντάκτης (Editor): Έχει δικαίωμα να δημιουργεί, επεξεργάζεται και να δημοσιεύει περιεχόμενο. Μπορεί επίσης να επεξεργάζεται περιεχόμενο που έχουν δημιουργήσει άλλοι χρήστες αλλά δεν έχει πρόσβαση στις ρυθμίσεις της ιστοσελίδας ή τα δικαιώματα χρηστών.

Συγγραφέας (Author): Μπορεί να δημιουργεί και να δημοσιεύει περιεχόμενο αλλά δεν μπορεί να επεξεργάζεται περιεχόμενο που έχουν δημιουργήσει άλλοι χρήστες.

Συνδρομητής (Subscriber): Μπορεί να διαβάσει ή να σχολιάσει το περιεχόμενο μιας σελίδας, αλλά δεν έχει δικαίωμα να δημιουργήσει ή να επεξεργαστεί το περιεχόμενο.

Επισκέπτης (Guest): Έχει περιορισμένη πρόσβαση και μπορεί μόνο να διαβάσει το περιεχόμενο που είναι διαθέσιμο στο κοινό.

142. Τι είναι το ψηφιακό Video; Αναφέρετε ονομαστικά τις γνωστότερες μεθόδους συμπίεσης και αντίστοιχους τύπους ψηφιακών αρχείων video.

Το ψηφιακό βίντεο είναι μια τεχνολογία που καταγράφει, αποθηκεύει και αναπαράγει κινούμενες εικόνες χρησιμοποιώντας ψηφιακά σήματα αντί αναλογικά. Αποτελείται από μια σειρά ψηφιακών εικόνων που προβάλλονται διαδοχικά δημιουργώντας την αίσθηση της κίνησης.

Γνωστότερες μέθοδοι συμπίεσης:

H.262: Είναι η μέθοδος συμπίεσης του προτύπου MPEG-2 που χρησιμοποιείται για αναπαραγωγή DVD και τηλεοπτική μετάδοση. Η κατάληξη των αρχείων είναι .mp2.

H.264: Είναι μέθοδος συμπίεσης του προτύπου MPEG-4. Το MPEG-4 είναι ευρέως διαδεδομένο και χρησιμοποιείται για streaming και άλλες εφαρμογές. Τα αρχεία έχουν κατάληξη .mp4.

H.265: Αποτελεί βελτίωση του H.264 και υποστηρίζει 4K/8K. Τα αρχεία έχουν και πάλι κατάληξη .mp4 ή .mkv.

VP9: Μέθοδος συμπίεσης ελεύθερου λογισμικού. Χρησιμοποιείται συχνά από το YouTube. Τα αρχεία έχουν κατάληξη .webm.

143. Τι είναι Component video σήμα;

Το component video είναι ένας τύπος αναλογικού σήματος βίντεο που διαχωρίζει το σήμα σε 3 κανάλια. Το ένα κανάλι (Y) μεταφέρει τη φωτεινότητα και τα άλλα δύο (U, V) πληροφορίες χρώματος.

Αυτός ο διαχωρισμός επιτρέπει την μετάδοση υψηλής ποιότητας εικόνας (720p, 1080i) με μειωμένες παρεμβολές. Χρησιμοποιείται σε συσκευές υψηλής ευκρίνειας όπως DVD players, τηλεοράσεις και κονσόλες παιχνιδιών.

144. Τι είναι Composite σήμα video;

Το composite video είναι ένας τύπος αναλογικού σήματος βίντεο. Μέσα από το ίδιο καλώδιο περνούν 3 διαφορετικά σήματα με παλμούς συγχρονισμού. Το ένα σήμα (Y) μεταφέρει τη φωτεινότητα και τους παλμούς χρονισμού. Τα άλλα δύο (U, V) μεταφέρουν πληροφορίες χρώματος. Είναι από τις παλαιότερες τεχνολογίες μετάδοσης αναλογικού σήματος και έχει χαμηλό κόστος υλοποίησης. Όμως επειδή όλες οι πληροφορίες συνδυάζονται στο ίδιο κανάλι υπάρχουν παρεμβολές και απώλεια ποιότητας εικόνας. Δεν υποστηρίζει HD.

145. Περιγράψτε τις διαφορές μεταξύ των σημάτων VGA και Βίντεο.

Το σήμα VGA είναι ένας τύπος μετάδοσης αναλογικού σήματος βίντεο. Χρησιμοποιείται κυρίως σε υπολογιστές για μετάδοση της εικόνας στην οθόνη. Το σήμα μεταφέρεται με ένα καλώδιο 15 ακροδεκτών. Οι πληροφορίες χρώματος μεταφέρονται με 3 ζεύγη, ένα ζεύγος για κάθε βασικό χρώμα (κόκκινο, πράσινο, μπλε). Με άλλα ζεύγη μεταφέρονται πληροφορίες για κάθετο και οριζόντιο συγχρονισμό. Προσφέρει πολύ υψηλή ανάλυση και ευκρίνεια εικόνας.

Υπάρχουν διάφορες τεχνολογίες για μεταφορά αναλογικών σημάτων βίντεο. Δύο από αυτές είναι:
Composite video: Το σήμα περνάει μέσα από το ίδιο καλώδιο με παλμούς συγχρονισμού. Ένα σήμα (Υ) μεταφέρει τη φωτεινότητα και τους παλμούς χρονισμού. Τα άλλα δύο (U, V) μεταφέρουν πληροφορίες χρώματος. Επειδή όλα τα σήματα περνούν μέσα από το ίδιο καλώδιο, δημιουργούνται παρεμβολές και απώλεια ποιότητας εικόνας. Δεν υποστηρίζει HD.

Component video: Το σήμα μεταφέρεται με 3 ξεχωριστά καλώδια. Το ένα καλώδιο (Υ) μεταφέρει τη φωτεινότητα και τα άλλα δύο (U, V) πληροφορίες χρώματος. Προσφέρει καλή ποιότητα εικόνας με μειωμένες απώλειες. Υποστηρίζει HD (720p, 1080i).

146. Ποια είναι τα βασικά μέρη του ψηφιακού βίντεο;

Frame size: Ο αριθμός εικονοστοιχείων (pixel) του κάθε καρέ.

Frame rate: Ο αριθμός των καρέ που προβάλλεται ανά δευτερόλεπτο.

Χρώμα και βάθος χρώματος: Στο χρωματικό μοντέλο RGB υπάρχουν 3 βασικά χρώματα (κόκκινο, πράσινο, μπλε). Το βάθος χρώματος εκφράζεται σε bit και καθορίζει πόσες διαβαθμίσεις κάθε χρώματος μπορούν να απεικονιστούν. Για βάθος 24bit έχουμε true color (16.7 εκατομμύρια).

Συμπίεση: Η μέθοδος συμπίεσης ώστε να ελαττωθεί το μέγεθος του αρχείου.

147. Σε τι διαφέρει το Video από το Animation και ποιο κοινό χαρακτηριστικό έχουν;

Το κοινό χαρακτηριστικό είναι η κίνηση. Τόσο το βίντεο όσο και το animation προβάλλουν διαδοχικά εικόνες που διαφέρουν ελάχιστα μεταξύ τους δημιουργώντας την αίσθηση της κίνησης.

Το video δεν χρειάζεται καμία προετοιμασία για να δημιουργηθεί. Οποιοσδήποτε με μία βιντεοκάμερα ή ένα κινητό μπορεί να καταγράψει μια σκηνή και στη συνέχεια να την αναπαράγει.

Το animation δημιουργείται από καλλιτέχνες που σχεδιάζουν ένα-ένα το κάθε καρέ. Στη συνέχεια με κατάλληλα προγράμματα αυτές οι εικόνες μετατρέπονται σε βίντεο.

Επίσης ενώ το video αποτελείται πάντα από bitmap γραφικά, τα καρέ του animation συνήθως σχεδιάζονται με vector γραφικά.

148. Εξηγήστε τις διαφορές μεταξύ γραμμικής (linear) και μη γραμμικής (non linear) επεξεργασίας video.

Η γραμμική (linear) επεξεργασία γίνεται με τη σειρά που εμφανίζονται τα καρέ στο βίντεο. Για να επεξεργαστούμε το τέλος του βίντεο θα πρέπει να προσπελάσουμε όλα τα καρέ από την αρχή μέχρι να φτάσουμε στο τέλος. Αυτός ο τρόπος χρησιμοποιείται σε σειριακές μονάδες αποθήκευσης όπως κασέτες ή μαγνητικές ταινίες.

Στη μη γραμμική (non-linear) επεξεργασία μπορούμε να επεξεργαστούμε οποιοδήποτε καρέ του βίντεο χωρίς την ανάγκη να προσπελάσουμε τα προηγούμενα. Η επεξεργασία γίνεται με ψηφιακά εργαλεία και εφαρμογές όπως το Adobe Premiere.

149. Πότε χρησιμοποιούνται οι λειτουργίες προεπισκόπηση (preview video) και σύλληψη (capture video), κατά την επεξεργασία ψηφιακού video;

Η σύλληψη (capture) χρησιμοποιείται για την καταγραφή βίντεο από εξωτερικές πηγές όπως βιντεοκάμερες, βίντεο ή άλλες συσκευές. Η διαδικασία αυτή μετατρέπει το αναλογικό ή ψηφιακό σήμα σε αρχείο βίντεο που στη συνέχεια μπορούμε να επεξεργαστούμε στον υπολογιστή.

Η προεπισκόπηση (preview) μας επιτρέπει να δούμε πως θα φαίνεται το βίντεο μετά την επεξεργασία και πριν την τελική εξαγωγή, ώστε να προβούμε σε αλλαγές αν είναι απαραίτητο.

150. Τι είναι τα “Transition Effect” σε ένα πρόγραμμα επεξεργασίας video; Δώστε ένα παράδειγμα.

Τα transition effects είναι εφέ που χρησιμοποιούνται για την ομαλή ή εντυπωσιακή μετάβαση μεταξύ δύο καρέ ή σκηνών.

Ένα δημοφιλές transition effect είναι το Fade όπου η εικόνα εξαφανίζεται σταδιακά (fade out) ή εμφανίζεται σταδιακά (fade in) δημιουργώντας μια ομαλή μετάβαση μεταξύ των σκηνών.

151. Εξηγήστε τη λειτουργία των εργαλείων trimming (ψαλίδισμα) και cropping (ξάκρισμα), κατά την επεξεργασία μοντάζ ψηφιακού video.

Το trimming χρησιμοποιείται για να αφαιρεθούν καρέ από την αρχή ή το τέλος του βίντεο. Δεν επηρεάζει το πλαίσιο (frame) ή τις διαστάσεις του βίντεο αλλά μόνο τη χρονική περίοδο που διαρκεί. Με το trimming μπορούμε για παράδειγμα να ψαλιδίσουμε ένα βίντεο διάρκειας 20 λεπτών και να κρατήσουμε μόνο τα 3 πρώτα και τα 2 τελευταία λεπτά, μειώνοντας τη συνολική διάρκεια σε 5 λεπτά.

Το cropping αφαιρεί τμήματα από το οπτικό πλαίσιο του βίντεο δηλαδή κόβει τις άκρες του κάδρου. Το cropping δεν επηρεάζει τη χρονική διάρκεια του βίντεο αλλά τις διαστάσεις του. Με το cropping μπορούμε για παράδειγμα να ξακρίσουμε τις μαύρες μπάρες που εμφανίζονται στο επάνω και κάτω ενός βίντεο για να τις εξαλείψουμε.

152. Τι είναι ο υπέρτιτλος (Superimposed clip) στη διαδικασία μοντάζ ψηφιακού video;

Ο υπέρτιτλος (superimposed clip) είναι ένα στοιχείο που τοποθετείται επάνω από το κύριο βίντεο σε ξεχωριστό επίπεδο (layer). Το στοιχείο αυτό μπορεί να είναι κείμενο, γραφικά, εικόνες ή ακόμη και ένα άλλο βίντεο που προβάλλεται ταυτόχρονα με το κύριο βίντεο.

Συχνά στον υπέρτιτλο χρησιμοποιούμε διαφάνεια ώστε να φαίνεται από κάτω το κύριο βίντεο. Ο υπέρτιτλος μπορεί να έχει μικρή χρονική διάρκεια ή να διαρκεί όσο το κανονικό βίντεο.

Συνήθως χρησιμοποιείται για να προσθέσουμε πληροφορίες στο βίντεο, όπως τίτλους, υπότιτλους ή πληροφορίες παρουσιαστών.

153. Να περιγράψετε τα κύρια χαρακτηριστικά ενός ασυμπίεστου ψηφιακού αρχείου video.

Υψηλή ποιότητα εικόνας: Στα ασυμπίεστα βίντεο κάθε καρέ διατηρεί όλη την ποιότητα της αρχικής στατικής εικόνας χωρίς καμία παραμόρφωση ή απώλεια λεπτομερειών.

Μεγάλο μέγεθος αρχείου: Η έλλειψη συμπίεσης οδηγεί σε πολύ μεγάλα μεγέθη αρχείων που είναι δύσκολο να αποθηκευτούν και να μεταφερθούν. Το μέγεθος αρχείου εξαρτάται από:

- Το frame size, δηλαδή τις διαστάσεις (ύψος X πλάτος) του κάδρου.
- Το frame rate, δηλαδή τον αριθμό καρέ που προβάλλονται ανά δευτερόλεπτο.
- Το color-depth, δηλαδή τον αριθμό χρωμάτων που υπάρχουν.
- Τη χρονική διάρκεια του βίντεο.

Υψηλές απαιτήσεις: Η αναπαραγωγή ή επεξεργασία ενός ασυμπίεστου βίντεο απαιτεί μεγάλη υπολογιστική ισχύ.

Υψηλός ρυθμός μετάδοσης δεδομένων: Επειδή τα καρτέ είναι ασυμπίεστα, το bitrate είναι εξαιρετικά υψηλό και απαιτούνται ταχύτατοι δίσκοι για αναπαραγωγή σε πραγματικό χρόνο.

Τα ασυμπίεστα βίντεο χρησιμοποιούνται συνήθως από επαγγελματίες για την παραγωγή και επεξεργασία master αρχείων και κρατούνται ως μήτρες.

154. Τι είναι κωδικοποιητής ψηφιακών αρχείων video πραγματικού χρόνου (real time video encoders);

Οι κωδικοποιητές ψηφιακών αρχείων πραγματικού χρόνου είναι συσκευές ή λογισμικό που μετατρέπουν αναλογικά ή ψηφιακά σήματα βίντεο σε κατάλληλη μορφή για αποθήκευση ή μετάδοση σε πραγματικό χρόνο. Δηλαδή την ίδια στιγμή που γίνεται η σύλληψη, γίνεται και η καταγραφή ή μετάδοση.

155. Περιγράψτε την τεχνική συμπίεσης ψηφιακού video MPEG.

Το MPEG είναι μια μέθοδος μείωσης του μεγέθους ενός αρχείου βίντεο χωρίς να θυσιάζεται σε μεγάλο βαθμό η ποιότητα της εικόνας.

Ένα βίντεο αποτελείται από στατικές εικόνες που διαφέρουν ελάχιστα μεταξύ τους και προβάλλονται διαδοχικά ώστε να δίνουν την αίσθηση της κίνησης. Το πρότυπο MPEG επιλέγει κάποιες από αυτές τις εικόνες, τις συμπιέζει και τις χρησιμοποιεί σαν σημεία αναφοράς (I-Frames). Τα υπόλοιπα καρτέ δεν αποθηκεύουν την πλήρη εικόνα αλλά μόνο τις διαφορές που υπάρχουν μεταξύ του προηγούμενου ή επόμενου I-Frame (P-Frames, B-Frames). Με τη διαδικασία αυτή το μέγεθος ενός αρχείου βίντεο μειώνεται δραματικά.

156. Περιγράψτε τα τυπικά βήματα της διαδικασίας Μοντάζ για την παραγωγή ενός ψηφιακού αρχείου video με τη χρήση ενός εργαλείου επεξεργασίας video (πχ. Premiere).

- Δημιουργία ενός νέου project και ρύθμιση παραμέτρων όπως ανάλυση, frame rate και aspect ratio.
- Εισαγωγή στο project των αρχείων βίντεο, εικόνων, γραφικών, ήχου κλπ που θα συμμετέχουν στο μοντάζ.
- Δημιουργία timeline.
- Προσθήκη στο timeline των αρχείων που εισάγαμε στο project.
- Trimming δηλαδή αποκοπή των τμημάτων που δεν θέλουμε στο τελικό μοντάζ.
- Δημιουργία εφέ μετάβασης κατά την εναλλαγή σκηνών.
- Επεξεργασία ήχου.
- Προσθήκη υπέρτιτλων.
- Προεπισκόπηση και αν είμαστε ικανοποιημένοι εξαγωγή του τελικού βίντεο.

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.

Σε ασυμπίεστο βίντεο για να υπολογίσουμε το μέγεθος του αρχείου χρησιμοποιούμε τον τύπο:
Πλάτος Χ Ύψος Χ (Βάθος χρώματος / 8) Χ Frame rate Χ διάρκεια σε δευτερόλεπτα.

Στο σήμα PAL ή μετάδοση γίνεται με 25 καρέ το δευτερόλεπτο, άρα:
 $320 \times 240 \times (16/8) \times 25 \times 120 = 460.800.000 \text{ bytes}$, δηλαδή 450.000KB.

158. Αναφέρατε τους τύπους των symbols που υπάρχουν στο Flash και τα χαρακτηριστικά του καθενός από αυτούς.

Υπάρχουν 3 τύποι symbols:

Graphic: Είναι το πιο απλό symbol. Δεν έχει δικό του timeline και δεν μπορεί να ελεγχθεί από την ActionScript. Χρησιμοποιούνται για δημιουργία στατικών εικόνων ή απλών animations που δεν απαιτούν διαδραστικότητα.

Movie clip: Είναι ανεξάρτητα animations με δικό τους timeline που μπορεί να περιέχουν άλλα symbols. Μπορούν να ελεγχθούν μέσω ActionScript. Χρησιμοποιούνται για τη δημιουργία διαδραστικών ή σύνθετων animations.

Button: Χρησιμοποιούνται για τη δημιουργία διαδραστικών κουμπιών που ανταποκρίνονται στις ενέργειες του χρήστη. Έχουν 4 καταστάσεις: Up, Down, Over, Hit.

159. Αναφέρατε τα πεδία του classic text που μπορούν να δημιουργηθούν από την εργαλειοθήκη του Flash και σε ποιες περιπτώσεις χρησιμοποιούνται.

Τα πεδία που μπορούν να δημιουργηθούν είναι:

Static text: Το πεδίο περιέχει κείμενο μόνο για προβολή. Χρήσιμο για εμφάνιση τίτλων, οδηγιών κλπ.

Dynamic text: Είναι κείμενο που αλλάζει δυναμικά και μπορεί να ελεγχθεί από την ActionScript. Θα μπορούσε για παράδειγμα να εμφανίζει το σκορ, ημερομηνία κλπ.

Input text: Είναι κείμενο που μπορεί να εισάγει ο χρήστης. Χρήσιμο σε συμπλήρωση φορμών κλπ.

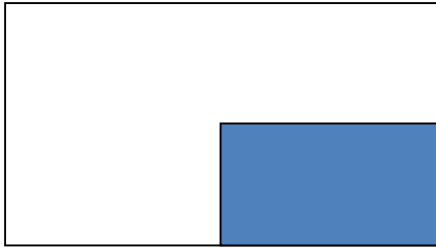
160. Ποιες οι διαφορές μεταξύ keyframe, blank keyframe και frame στο flash;

Frame: Είναι ένα απλό καρέ στο timeline που δεν περιέχει καμία αλλαγή ή κίνηση. Χρησιμοποιείται για να διατηρήσει σταθερή την κατάσταση του animation μεταξύ των keyframes.

Keyframe: Είναι ένα καρέ που χρησιμοποιείται για να ορίσει μια αλλαγή στην εμφάνιση ή την κίνηση του animation.

Blank keyframe: Είναι ένα keyframe χωρίς περιεχόμενο και χρησιμοποιείται για να εισάγει ένα κενό καρέ. Κανονικά όταν εισάγουμε ένα keyframe τα περιεχόμενα της σκηνής αντιγράφονται στο νέο keyframe. Αν όμως θέλουμε να δημιουργήσουμε μια σκηνή που είναι εντελώς διαφορετική από την προηγούμενη τότε χρησιμοποιούμε το blank keyframe για να μη σβήσουμε όλα τα αντικείμενα που έχουν αντιγραφεί.

- 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 μικρά κομμάτια κώδικα που εκτελούν εργασίες που χρησιμοποιούνται συχνά. Μπορεί να υπάρχουν σε έτοιμες βιβλιοθήκες ή ο προγραμματιστής να γράψει τα δικά του. Εξοικονομούν χρόνο και αποφεύγουμε την επαναλαμβανόμενη συγγραφή κώδικα.

- 163. Πότε χρησιμοποιείται η εντολή trace bitmap;**

Η trace bitmap χρησιμοποιείται όταν θέλουμε να μετατρέψουμε ένα bitmap γραφικό σε διανυσματικό.

- 164. Ποια η διαφορά μεταξύ symbol και instance;**

Το symbol είναι ένα graphic, movie clip ή button που το δημιουργούμε μία φορά και προστίθεται στη βιβλιοθήκη.

Το instance είναι ένα στιγμιότυπο του symbol που δημιουργείται όταν το μεταφέρουμε στη σκηνή. Όταν αλλάζουμε τις ιδιότητες ενός symbol αλλάζουν και οι ιδιότητες όλων των instance που χρησιμοποιούνται στη σκηνή. Το αντίθετο όμως δεν ισχύει.

- 165. Πού βρίσκεται και πώς λειτουργεί το εργαλείο onion skin στο flash;**

Το onion skin βρίσκεται στα εικονίδια εργαλείων του timeline.

Όταν είναι ενεργοποιημένο το onion skin μας επιτρέπει να δούμε πολλά καρέ ταυτόχρονα, δηλαδή τα προηγούμενα και τα επόμενα καρέ σε σχέση με το τρέχον. Τα καρέ που εμφανίζονται με το onion είναι ημιδιαφανή κι έτσι μπορούμε να δούμε τις διαφορές που υπάρχουν μεταξύ των καρέ. Αυτό μας διευκολύνει να πετύχουμε ομαλή κίνηση προσαρμόζοντας τις θέσεις των αντικειμένων του τρέχοντος καρέ.

- 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 καθορίζει την ενεργή περιοχή του κουμπιού που θα ανταποκρίνεται σε ενέργειες όπως το κλικ ή το πέρασμα του mouse πάνω από το κουμπί. Το frame HIT δεν είναι ορατό κατά την αναπαραγωγή.

Όταν το κουμπί περιέχει μόνο κείμενο το frame HIT περιορίζεται στο περίγραμμα των γραμμάτων του κειμένου. Έτσι είναι δύσκολο για τον χρήστη να πετύχει την ακριβή περιοχή που θα

ενεργοποιηθεί το κουμπί. Η λύση σ' αυτό είναι να προσθέσουμε στο frame HIT ένα ορθογώνιο που θα καλύπτει όλη την ενεργή περιοχή του κουμπιού. Ο χρήστης δεν θα βλέπει το ορθογώνιο αφού το frame HIT είναι αόρατο.

167. Ποια είδη tweening υπάρχουν στο Flash και ποια είναι τα χαρακτηριστικά τους;

Για να δημιουργήσουμε ένα tween χρειαζόμαστε δύο keyframes. Στο πρώτο keyframe τοποθετούμε τα αντικείμενα στην σκηνή και στο δεύτερο keyframe αλλάζουμε τη θέση, μέγεθος, χρώμα κλπ των αντικειμένων όπως θέλουμε να είναι στο τέλος του animation. Το flash αναλαμβάνει να δημιουργήσει όλα τα ενδιάμεσα καρέ για τη δημιουργία του animation.

Υπάρχουν 3 είδη tweening:

Classic tween: Τα αντικείμενα πρέπει να είναι symbols. Χρησιμοποιείται για τη δημιουργία βασικών κινήσεων ή αλλαγών σε ιδιότητες όπως μέγεθος, περιστροφή, θέση, διαφάνεια.

Motion tween: Πρέπει και εδώ τα αντικείμενα να είναι symbols ή κείμενο. Είναι πιο ευέλικτο και ισχυρό από το classic tween, προσφέροντας επιπλέον δυνατότητες όπως το motion path όπου μπορούμε να ορίσουμε καμπύλες για την πορεία των αντικειμένων στα ενδιάμεσα καρέ.

Shape tween: Δεν απαιτεί τα αντικείμενα να είναι symbols. Χρησιμοποιείται συνήθως για την μετάβαση από ένα σχήμα σε άλλο (πχ από ένα κύκλο σε αστέρι).

168. Πώς δουλεύει το εργαλείο Bone Tool;

Το Bone Tool χρησιμοποιείται για τη δημιουργία σκελετών που επιτρέπουν ρεαλιστικές κινήσεις χαρακτήρων ή αντικειμένων. Η λειτουργία του βασίζεται στην εισαγωγή αρθρώσεων που ενώνουν τα διάφορα μέρη ενός χαρακτήρα ή αντικειμένου. Τα μέρη που ενώνονται μπορεί να είναι symbols ή σχήματα.

Για να σχεδιάσουμε τον σκελετό επιλέγουμε το Bone Tool από τη γραμμή εργαλείων. Κάθε κόκκαλο συνδέει 2 σημεία. Το σημείο έναρξης και το σημείο σύνδεσης με το επόμενο κόκκαλο. Δημιουργείται έτσι ένας σκελετός με συνδέσεις που επιτρέπει στο αντικείμενο να κινείται σαν ένα σώμα με αρθρώσεις.

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, handleKeyDown);
```

```
function handleKeyDown(e:KeyboardEvent):void {  
    switch (e.keyCode) {  
        case Keyboard.LEFT:  
            mClip1.x -= 5;  
            break;  
        case Keyboard.RIGHT:  
            mClip1.x += 5;  
            break;  
        case Keyboard.UP:  
            mClip1.y -= 5;  
            break;  
        case Keyboard.DOWN:  
            mClip1.y += 5;  
    }  
}
```

```
        break;  
    }  
}
```

170. Με ποιο τρόπο κάνουμε path animation ενός tween στο Flash;

- Στο timeline κάνουμε δεξί κλικ στο Layer και επιλέγουμε Add Classic Motion Guide. Θα δημιουργηθεί ένα Guide Layer.
- Στο timeline επιλέγουμε ένα χρονικό διάστημα (πχ 60 sec) και εισάγουμε ένα απλό frame και για τα 2 layers.
- Πηγαίνουμε στην αρχή του timeline, επιλέγουμε το Guide Layer και με το εργαλείο brush σχεδιάζουμε το μονοπάτι που θέλουμε να ακολουθεί το animation.
- Επιλέγουμε το κανονικό Layer και σχεδιάζουμε το σχήμα μας, για παράδειγμα ένα κύκλο.
- Επιλέγουμε τον κύκλο που σχεδιάσαμε και τον μετατρέπουμε σε symbol (πχ graphic).
- Τοποθετούμε το κέντρο του κύκλου στην αρχή του μονοπατιού που είχαμε σχεδιάσει.
- Πηγαίνουμε στο τέλος του timeline (60 sec) και εισάγουμε ένα keyframe.
- Ενώ είμαστε στο keyframe μετακινούμε τον κύκλο στο τέλος του μονοπατιού.
- Κάνουμε δεξί κλικ κάπου στο ενδιάμεσο του timeline και επιλέγουμε Create Classic Tween.
- Το Flash δημιουργεί αυτόματα το animation του κύκλου που θα κινείται επάνω στο μονοπάτι.

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;

Οτιδήποτε σχεδιάζουμε πάνω στη μάσκα λειτουργεί σαν ορατή περιοχή του layer που μασκάρεται. Για αυτό θα βάζαμε τον τίτλο στη μάσκα και στον layer που μασκάρεται θα δημιουργούσαμε ένα animation με νερό που κυματίζει. Έτσι το animation του νερού θα φαινόταν μόνο μέσα από τα περιγράμματα των γραμμάτων της μάσκας.

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).

- Καταρχήν θα πρέπει να δημιουργήσουμε ένα symbol τύπου Movie Clip.
- Μέσα σ' αυτό το symbol θα δημιουργήσουμε 2 κύκλους, ένα κόκκινο μεγαλύτερο και ένα μπλε μικρότερο.
- Στην συνέχεια επιλέγουμε και τους 2 κύκλους -> δεξί κλικ -> Create Motion Tween.

- Κάνουμε κλικ στο timeline που δημιουργήσε αυτόματα το Flash και στα Properties επιλέγουμε Rotate -> Clockwise.

Με τον τρόπο αυτό έχουμε δημιουργήσει ένα μπλε κύκλο που περιστρέφεται γύρω από τον κόκκινο. Όμως ο κόκκινος δεν κινείται.

- Βγαίνουμε από την επεξεργασία του symbol.
- Τραβάμε το symbol στο αριστερό άκρο της σκηνής.
- Στο timeline δημιουργούμε ένα keyframe (πχ στα 100sec) και μεταφέρουμε το symbol οριζόντια στο δεξιό άκρο της σκηνής.
- Δεξί κλικ μέσα στο timeline -> Create Classic Tween.

Με τον τρόπο αυτό κάναμε το symbol να διασχίζει οριζόντια τη σκηνή. Δηλαδή ο κόκκινος κύκλος θα διασχίζει οριζόντια τη σκηνή αλλά ταυτόχρονα ο μπλε θα περιφέρεται γύρω του.

174. Δώστε ένα παράδειγμα (στο Flash) που γίνεται collision detection μεταξύ του movie clip mc_1 και του mc_2. Το movie clip 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 movie clip mc_2 η κατεύθυνσή του αναστρέφεται.

```
var step:int = 5;

stage.addEventListener(Event.ENTER_FRAME, mainLoop);

function mainLoop(e:Event):void {
    mc_1.x += step;

    // Αν φτάσεις στο τέλος ή την αρχή της σκηνής
    // συνέχισε από την άλλη μεριά
    if (mc_1.x > stage.stageWidth) {
        mc_1.x = 0;
    }
    if (mc_1.x < 0 - mc_1.width) {
        mc_1.x = stage.stageWidth;
    }

    // Αν υπάρχει σύγκρουση άλλαξε κατεύθυνση
    if (mc_1.hitTestObject(mc_2)) {
        step *= -1;
    }
}
```