

Στις εργασίες που ακολουθούν μπορείτε και πρέπει να χρησιμοποιήσετε έτοιμες συναρτήσεις κώδικα που θα βρείτε είτε στις σημειώσεις του μαθήματος είτε ακόμα στο διαδίκτυο. Αυτό που ζητείται είναι να κατανοείτε τη λειτουργία τους και να είστε σε θέση να τις προσαρμόξετε στις ανάγκες των προβλημάτων σας.

Αυτό που θα παραδώσετε είναι τους πλήρης πηγαίους κώδικες σε κάθε πρόβλημα και βέβαια θα τους κρατήσετε και εσείς για να τους προσαρμόσετε στις ανάγκες των εξετάσεων για τα ερωτήματα που θα σας τύχουν.

Είναι ευκαιρία να περάσετε το μάθημα γιατί έχετε μπροστά σας περίπου 40 ημέρες και στις εξετάσεις δεν θα έχετε κάτι που μπορεί να σας αιφνιδιάσει.

Επαναλαμβάνω ότι ο βαθμός σας θα βγει από αυτό που θα μου παραδώσετε αλλά και από την επιλογή των σωστών απαντήσεων κατά τις εξετάσεις σε 3 από τα 15 πιο κάτω προβλήματα. Δηλαδή στις εξετάσεις θα σας ζητήσω για παράδειγμα να τρέξετε το πρόγραμμα της άσκησης 5 με δικό μου AEM και να επιλέξετε τη σωστή από τις απαντήσεις που θα αναφέρονται στις απαντήσεις.

Τις εργασίες θα τις παραδώσετε μέχρι τα μεσάνυχτα της Κυριακής 12/6/2022 και θα εξεταστείτε την Κυριακή 26/6/2022 στις 16:00 με 17:00 στο εργαστήριο A4

Δεν θα βαθμολογηθεί κανένας φοιτητής που δεν θα παραδώσει εμπρόθεσμα τις εργασίες μέσα από την ηλεκτρονική τάξη. Δεν βαθμολογώ μόνο την εξέταση αλλά τις εργασίες που θα παραδώσετε. Η εξέταση επιβεβαιώνει ή διαψεύδει ότι ΕΣΕΙΣ κάνατε τις εργασίες.

1. Τριδιαγώνιος είναι ένας αραιός πίνακας, που εμφανίζεται συχνά στην Αριθμητική Ανάλυση. Είναι ένας τετραγωνικός πίνακας $n \times n$ που έχει υποχρεωτικά όλα του τα στοιχεία 0 εκτός από τα στοιχεία της κυρίας διαγωνίου και των αμέσως παραπλεύρων διαγωνίων του, επάνω και κάτω από αυτή. Παράδειγμα τέτοιου πίνακα 5×5 είναι πιο κάτω

$$A = \begin{pmatrix} 5 & 8 & 0 & 0 & 0 \\ -3 & 4 & 7 & 0 & 0 \\ 0 & 12 & -6 & 11 & 0 \\ 0 & 0 & 5 & 17 & -13 \\ 0 & 0 & 0 & -14 & 12 \end{pmatrix}$$

Να κατασκευάσετε με αποτελεσματικό τρόπο, πρόγραμμα σε C που να παράγει με τυχαίο τρόπο, αρχικά τη διάσταση n τριδιαγώνιου πίνακα με $4 \leq n \leq 8$. Έπειτα, να παράγονται και να καταχωρούνται με τυχαίο επίσης τρόπο, τα μη μηδενικά στοιχεία του, με εύρος $-20 \leq a_{ij} \leq 20$ για $i-1 \leq j \leq i+1$ και $0 < j \leq n$.

Η συνάρτηση τυχαιότητας να αρχικοποιείται με το AEM σας. Δηλαδή srand(AEM);

Όταν ολοκληρωθεί η καταχώρηση να εμφανίζονται στην οθόνη με την τετραγωνική μορφή πίνακα, όπως ο A. Στη συνέχεια να αποθηκεύει τον τριδιαγώνιο αυτό πίνακα, σε νέα δομή μονοδιάστατου πίνακα, στον οποίο θα καταχωρούνται μόνο τα στοιχεία των τριών διαγωνίων του. Τον πίνακα αυτόν να τον εμφανίζει επίσης στην οθόνη οριζόντια, το ένα στοιχείο πίσω από το άλλο με τη λογική να εμφανίζονται με τη σειρά τα στοιχεία κάθε γραμμής του τετραγωνικού πίνακα, π.χ. για τον A: 5, 8, -3, 4, 7, 12, -6, 11

Στη συνέχεια το πρόγραμμα να εμφανίζει στην οθόνη το μεγαλύτερο και το μικρότερο στοιχείο του πίνακα.

Τέλος να εμφανίζεται στην οθόνη, ο αντίστοιχος μονοδιάστατος ανάστροφος πίνακας του A και στη συνέχεια σε τετραγωνική μορφή πίνακα. Ο ανάστροφος πίνακας A^T προκύπτει αν οι γραμμές του πίνακα A γίνουν στήλες και οι στήλες του γραμμές του A^T .

2. Η γενική μορφή του πολωνύμου βαθμού n είναι :

$$P_n(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + \dots + a_{n-2}x^{n-2} + a_{n-1}x^{n-1} + a_nx^n, \quad a_n \neq 0$$

Το μονώνυμο $a_i x^i$ μπορεί να παρασταθεί ως δομή (struct) θα αποτελεί κόμβο με δυο μέλη, το ένα θα περιέχει το συντελεστή a_i και το άλλο τον εκθέτη i .

Συνολικά το πολυνόμο μπορεί να αποθηκευθεί σε μια δομή δεδομένων **λίστας**. Τα μέλη της λίστας (κόμβοι) θα είναι οι δομές των μονωνύμων.

Να κατασκευάσετε πρόγραμμα, που επιλέγει αρχικά, με τυχαίο τρόπο την τάξη (βαθμό) του πολωνύμου $4 \leq n \leq 7$ και στη συνέχεια με τυχαίο τρόπο θα διαλέγει τους συντελεστές $-20 \leq a_i \leq 20$.

Η αρχικοποίηση της συνάρτησης τυχαιότητας θα γίνεται με τον AEM σας δηλαδή srand(AEM); και θα αποθηκεύει σε κατάλληλη δομή λίστας το πολυνόμο που προκύπτει.

Το πρόγραμμα θα εμφανίζει στη συνέχεια στην οθόνη το πολυνόμο για παράδειγμα πέμπτου βαθμού με τη μορφή: $P_5(x)=12+32x-25x^2+44x^3-27x^4-33x^5$ όπου το $-27x^4$ σημαίνει ο συντελεστής στον όρο με εκθέτη 4 είναι -27.

Στη συνέχεια το πρόγραμμα με τον ίδιο τρόπο δημιουργεί και δεύτερο πολυνόμο το οποίο καταχωρεί σε **δεύτερη** λίστα και το εμφανίζει με τον ίδιο τρόπο στην οθόνη.

Τελικά σε τρίτη λίστα να αποθηκεύει το άθροισμα των δυο πολωνύμων και να το εμφανίζει στην οθόνη με τον ίδιο τρόπο. Το άθροισμα των πολωνύμων προκύπτει από την πρόσθεση των συντελεστών των ομοιοβάθμων όρων του.

3. Σε πρόγραμμα της C, να δηλώνεται μια δομή “student” που αφορά φοιτητή. Η δομή περιλαμβάνει με τη σειρά τα μέλη-πεδία aem ως μη προσημασμένο μικρό ακέραιο στο οποίο καταχωρείται

ο αριθμός μητρώου, lname και fname ως συμβολοσειρές μήκους 16 χαρακτήρων η κάθε μια, και το year που αφορά το έτος εισαγωγής στο Τμήμα ως μικρό μη προσημασμένο ακέραιο και αυτό.

Με βάση την δομή που ορίσατε, να δηλώσετε πίνακα 100 θέσεων τύπου δομής φοιτητών. Με κατάλληλη δομή επανάληψης να καταχωριστούν με τυχαίο τρόπο, τα στοιχεία ενενήντα φοιτητών, δηλαδή τυχαίο AEM από 3000-4000, ως επώνυμο τυχαίος κεφαλαίος χαρακτήρας και ως όνομα τυχαίος πεζός χαρακτήρας και για το έτος εισαγωγής τυχαίος αριθμός στο εύρος 2014-2020. Η αρχικοποίηση της παραγωγής των τυχαίων θα γίνει με τη συνάρτηση srand(AEM) όπου AEM ο αριθμός μητρώου σας. Για παράδειγμα αν ο αριθμός μητρώου μου είναι το 9999 θα καλείται με srand(9999).

Στη συνέχεια το πρόγραμμα θα βρίσκει και θα εμφανίζει ποιος ή ποιοι φοιτητές φοιτούν περισσότερο από 4 χρόνια με δεδομένο ότι βρισκόμαστε στο 2021.

4. Να δημιουργήσετε πρόγραμμα σε C το οποίο να διαχειρίζεται σύνολα. Για το σκοπό αυτό πρέπει να ορισθεί κατάλληλη δομή δεδομένων συνόλου η οποία θα είναι εφοδιασμένη με τις εξής βασικές λειτουργίες :

1. **Δημιουργία** κενού συνόλου, δηλαδή δημιουργία της δομής χωρίς να περιλαμβάνει στοιχεία.
2. **Καταχώρηση** στοιχείου. Καταχωρεί, αν δεν υπάρχει, δεδομένο στοιχείο στο σύνολο.
3. **Διαγραφή** στοιχείου. Αφαιρεί συγκεκριμένο στοιχείο από το σύνολο
4. **Ανήκει**. Εξετάζει αν δεδομένο στοιχείο είναι μέλος του συνόλου.
5. **Πλήθος** στοιχείων, ενημερώνει για τον αριθμό των μελών του συνόλου.
6. **Ένωση** συνόλων. Δηλαδή έστω τα δύο σύνολα $A = \{4, 1, 7\}$ και $B = \{7, 4, 2, 8\}$
 $\Gamma = A \cup B = \{1, 2, 4, 7, 8\}$. Δηλαδή η ένωση περιλαμβάνει όλα τα στοιχεία των αρχικών συνόλων από μια φορά το καθένα.
7. **Τομή** συνόλων. Δηλαδή τα κοινά μόνο στοιχεία των δυο συνόλων $\Delta = A \cap B = \{4, 7\}$
8. **Υποσύνολο**. Λογική συνάρτηση η οποία εξετάζει αν το πρώτο όρισμά της είναι υποσύνολο του δευτέρου ορίσματος. Υποσύνολο είναι το σύνολο που όλα του τα στοιχεία του είναι και μέλη του άλλου. Παράδειγμα $A \subseteq \Gamma$, $\Delta \subseteq A$, $\Delta \subseteq B$, $\Delta \subseteq \Gamma$ κλπ.
9. **Διαφορά**. $B - A = \{2, 8\}$ δηλαδή η διαφορά περιλαμβάνει τα μέλη του B που δεν ανήκουν στο A, ενώ $A - B = \{1\}$.
10. **Ισότητα**. Λογική συνάρτηση που εξετάζει αν δύο σύνολα έχουν ακριβώς τα ίδια στοιχεία

Αφού δημιουργηθεί η δομή του συνόλου με τις πιο πάνω συναρτήσεις λειτουργίες, να δημιουργήσετε εφαρμογή σε C η οποία αρχικά, με τη συνάρτηση Δημιουργία να δημιουργεί δύο κενά σύνολα τα A και B. Στη συνέχεια με τη συνάρτηση τυχαιότητας rand() να επιλέγει τα μεγέθη τους στο εύρος [3-8]. Η αρχικοποίηση της συνάρτησης τυχαιότητας να γίνεται με την srand(AEM), όπου AEM ο αριθμός μητρώου σας.

Με το επιλεγμένο μέγεθος για κάθε σύνολο, να δημιουργεί μέλη για καθένα από τα δύο σύνολα A

και B και να τα καταχωρεί με τη συνάρτηση Καταχώρηση. Τα μέλη αυτά θα δημιουργούνται με τυχαίο τρόπο μέσω της rand() και θα είναι ακέραιοι αριθμοί στο εύρος [1-8]. Παράδειγμα αν για το σύνολο A προέκυψε μέγεθος 5 με την τυχαία σειρά ενδέχεται να δημιουργηθεί το σύνολο $A=\{6,2,8,1,5\}$.

Στη συνέχεια να προσδιοριστούν και να εμφανιστούν στην οθόνη τα εξής :

1. Να εμφανιστούν τα σύνολα A και B.
 2. Να προσδιοριστεί και να εμφανιστεί η ένωση Γ των δύο συνόλων A και B.
 3. Να εμφανιστεί το πλήθος των μελών του Γ .
 4. Να προσδιοριστεί και να εμφανιστεί η τομή Δ των δύο συνόλων A και B.
 5. Να προσδιοριστεί και να εμφανιστεί η διαφορά E του Δ από το Γ ($E = \Gamma - \Delta$)
 6. Να ελέγξει και να εμφανίσει σχετικό μήνυμα αν το σύνολο E είναι η όχι ίσο με το A και το B.
 7. Να ελέγξει και να εμφανίσει σχετικό μήνυμα αν το σύνολο Γ είναι υποσύνολο του B. Το ίδιο αν το σύνολο A είναι υποσύνολο του Γ .
 8. Να βρείτε και να εμφανίσετε στην οθόνη τα στοιχεία του E που ανήκουν και στο σύνολο B.
 9. Να διαγράψετε από το σύνολο A το μεγαλύτερό του μέλος και να εμφανίσετε τα εναπομείναντα.
5. Να δημιουργήσετε πρόγραμμα σε C τον οποίο να ορίζει **δυναμικά έναν μονοδιάστατο** πίνακα τυχαίου μεγέθους που θα δέχεται τυχαίους ακεραίους. Το μέγεθος θα προκύπτει με τη κλήση της συνάρτησης rand() στο εύρος 100-200. Η αρχικοποίηση της παραγωγής των τυχαίων θα γίνει με τη συνάρτηση srand(AEM) όπου AEM ο αριθμός μητρώου σας Για παράδειγμα αν ο αριθμός μητρώου μου είναι το 9999 θα καλείται με srand(9999).
- Να δεσμεύετε λοιπόν με τις κατάλληλες συναρτήσεις τον αναγκαίο χώρο μνήμης και όχι περισσότερο, από αυτόν που προκύπτει από τα προηγούμενα. Στη συνέχεια να καταχωριστούν στον πίνακα αυτό τόσοι τυχαίοι ακέραιοι **ζυγοί** (άρτιοι) αριθμοί όσο είναι και το μέγεθός του τυχαίου πίνακα και οι οποίοι βρίσκονται στο διάστημα από 400 μέχρι 500. Να προσδιορίσετε στη συνέχεια τον αριθμό που εμφανίστηκε τις περισσότερες φορές και πόσες.
6. Να δημιουργήσετε σε πρόγραμμα της C και να αξιοποιήσετε μια **στατική στοίβα**. Χρησιμοποιώντας τη στοίβα αυτή να υπολογίζετε την αναπαράσταση της χρονικής στιγμής στην οποία αντιστοιχεί το δεκαπλάσιο του AEM σας. Παράδειγμα για AEM 2437 (δηλαδή 24370 δευτερόλεπτα) να εμφανίζει 06:31:10 Το πρόγραμμά σας θα ορίζει τη στοίβα με διάσταση 6.
- Με τη χρήση της ίδιας στοίβας, να εμφανίσετε τις χρονικές στιγμές κατά τη διάρκεια μια ημέρας, ενός 24ώρου (00:00:00-23:59:59), που είναι παλίνδρομες και τους AEM στους οποίους αντιστοιχούν. Παλινδρομική χαρακτηρίζεται η παράσταση που διαβάζεται το ίδιο από το τέλος προς την αρχή π.χ. 01:33:10 ή 22:22:22 ενώ η 12:14:23 δεν είναι.



7. Στο διπλανό σχήμα εμφανίζονται διαγραμματικά η διάταξη βαγονιών εμπορικής αμαξοστοιχίας που μπορούν να κινηθούν στις τρεις γραμμές και

προς τις δυο κατευθύνσεις. Ο μηχανισμός χρειάζεται για να αλλάζουν τη σειρά των βαγονιών στην αμαξοστοιχία, στην περίπτωση μας κάθε γραμμή υλοποιείται με μια ξεχωριστή στοίβα στην οποία καταχωρούνται και αφαιρούνται βαγόνια μεταφέροντάς τα σε κάποια από τις άλλες δυο στοίβες.

Μια αμαξοστοιχία, με εννιά συνολικά εμπορικά βαγόνια, έφθασε στην Αθήνα προερχόμενη από την Πελοπόννησο και τερματικό προορισμό την Αλεξανδρούπολη. Κάθε ένα βαγόνι προορίζονται για τους σταθμούς, με αλφαβητική σειρά, Αλεξανδρούπολη Δράμα, Θεσσαλονίκη, Θήβα, Κατερίνη, Κομοτηνή, Λάρισα, Λειανοκλάδι και Λειβαδιά. Η σειρά των βαγονιών με την οποία φτάνει η αμαξοστοιχία στην

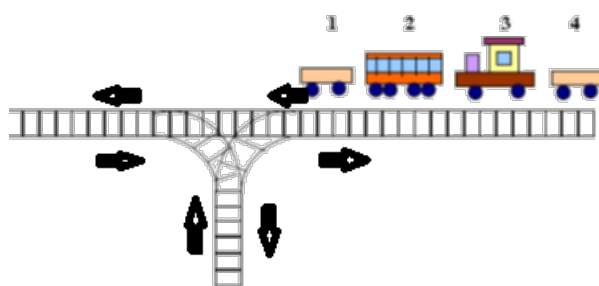
Αθήνα είναι τυχαία και προκύπτει από τη συνάρτηση τυχαιότητας την οποία πρέπει να αρχικοποιήσετε με τον AEM σας. Δηλαδή αρχικά τα βαγόνια θα είναι σε τυχαία σειρά διαφορετική για τον καθένα σας.

Στην Αθήνα λοιπόν πρέπει να αναδιαταχθούν τα βαγόνια ώστε η σειρά τους να είναι αντίστροφη της σειράς των πόλεων που διέρχεται το τραίνο στη διαδρομή που φαίνεται στον διπλανό χάρτη, ώστε σε κάθε σταθμό να αποσυνδέεται το τελευταίο βαγόνι.

Χρησιμοποιώντας ως δομή δεδομένων τρεις στοίβες να προγραμματίσετε σε C αλγόριθμο που να υλοποιεί την αναδιάταξη αυτή. Δηλαδή με κατάλληλη μεταφορά βαγονιών από στοίβα σε στοίβα να καταλήξετε σε διάταξη βαγονιών με πρώτο βαγόνι αυτό που προορίζεται για την Αλεξανδρούπολη και τελευταίο της Θήβας.

8 Σε Super Market υπάρχουν τρία αριθμημένα ταμεία για την εξυπηρέτηση των πελατών. Οι πελάτες εξυπηρετούνται με τη γνωστή σειρά σχηματίζοντας ουρές σε καθένα από αυτά. Ο μέγιστος αριθμός καταχωρήσεων στις ουρές είναι 10 πελάτες. Αν όλες οι ουρές υπερβούν τις 10 καταχωρήσεις το πρόγραμμα αγνοεί κάθε νέα προσέλευση πελάτη. Βέβαια κάθε πελάτης επιλέγει το ταμείο με την μικρότερη ουρά. Με τη σειρά της κάθε υπάλληλος του ταμείου χρειάζεται χρόνο για να εξυπηρετεί τους πελάτες ανάλογα με το πλήθος των αγορών τους.

Να θεωρηθεί ότι κάθε τρία προϊόντα εξυπηρετείται ένας πελάτης. Παράδειγμα ο πελάτης με 2 προϊόντα θεωρείται ως μια εξυπηρέτηση, ενώ ο πελάτης με 14 προϊόντα θα θεωρείται ως πέντε εξυπηρετήσεις. Αυ-



τό σημαίνει ότι ο αριθμός των πελατών που περιμένει σε κάθε ουρά εξαρτάται από το πόσα προϊόντα είχαν στο καλάθι οι προηγούμενοι.

Να δημιουργήσετε ένα πρόγραμμα που να προσομοιάζει τη λειτουργία αυτή του Super Market. Θα χρησιμοποιήσετε τη **δομή της ουράς**. Σε κάθε μια ουρά θα καταχωρούνται ως μέλη κόμβοι στους οποίους καταχωρούνται ο αριθμός του πελάτη και το πλήθος των προϊόντων του.

Για τις ανάγκες του προβλήματος θεωρίστε ότι όταν υπάρχει ο ίδιος αριθμός πελατών στις ουρές ο πελάτης διαλέγει τυχαία, μεταξύ των ισοδύναμων ουρών. Ο πελάτης που επέλεξε και καταχωρήθηκε σε κάποια ουρά δεν έχει τη δυνατότητα να αλλάξει ακόμα και αν έχει αδειάσει οποιαδήποτε άλλη ουρά.

Ο μηχανισμός με τον οποίο προσέρχεται ο επόμενος πελάτης ή εξυπηρετείται από την ταμιά καθορίζεται από το αποτέλεσμα **ζαριάς**. Αν το αποτέλεσμα της ζαριάς είναι μικρότερο του 3 θεωρείστε ότι προσέρχεται στα ταμεία ένας νέος πελάτης ο οποίος και θα διαλέξει μια από τις τρεις ουρές. Αν η ζαριά είναι μεγαλύτερη του 2 θεωρείστε ότι και οι τρεις ταμίες εξυπηρετούν έναν πελάτη κατά μία μονάδα δηλαδή κατά τρία προϊόντα.

Για την αρχικοποίηση του προγράμματος θα χρησιμοποιήσετε τη συνάρτηση `srand(AEM)` όπου AEM ο αριθμός μητρώου σας. Για παράδειγμα αν ο αριθμός μητρώου μου είναι το 9999 θα καλείται με `srand(9999)`. Με την τυχαία συνάρτηση `rand()` θα χειρίζεστε το πλήθος των προϊόντων που έχει κάθε πελάτης στο καλάθι αγορών του και αυτό θα κυμαίνεται από 2 μέχρι 17 προϊόντα. Επίσης με την `rand()` θα φέρνετε τις ζαριές που θα καθορίζουν την κατάσταση, αν δηλαδή προσέρχεται νέος πελάτης ή γίνεται ενέργεια εξυπηρέτησης από τις ταμίες.

Το σύστημα σε κάθε νέο κύκλο εμφανίζει στην οθόνη την κατάσταση, δηλαδή σε κάθε ταμείο πόσοι πελάτες περιμένουν στην ουρά του και πόσοι πελάτες έχουν ήδη εξυπηρετηθεί. Το πρόγραμμα ολοκληρώνεται όταν εξυπηρετηθούν 60 συνολικά πελάτες.

9. Να χρησιμοποιήσετε σε πρόγραμμα της C την **απλά συνδεδεμένη λίστα** στην οποία να καταχωρήσετε, με τη σειρά που προκύπτουν, 100 τυχαίους αριθμούς που δημιουργούνται με τη συνάρτηση `rand()` και κυμαίνονται στο $[100-1000]$. Για την αρχικοποίηση της σειράς των τυχαίων να χρησιμοποιήσετε την συνάρτηση `srand(AEM)` όπου AEM ο αριθμός μητρώου. Οι τυχαίοι αυτοί αριθμοί να εμφανίζονται στην οθόνη με τη σειρά που καταχωρούνται στη λίστα.

Στη συνέχεια από τη λίστα αυτή να αφαιρέσετε **διαδοχικά** πέντε τυχαία μέλη της, τα οποία θα προκύπτουν από την κλήση της συνάρτησης τυχειότητας (π.χ. το 5° το 13° το 8° κλπ) και να εμφανίσετε πάλι στην οθόνη τα εναπομείναντα μέλη της λίστας.

10. Να δημιουργήσετε **μονοδιάστατο πίνακα 50 θέσεων** στον οποίο να καταχωρείτε **διαδοχικά** σε κάθε του θέση τον AEM φοιτητή με τον τυχαίο βαθμό σε κάποιο μάθημα. Οι AEM των φοιτητών θα είναι σε αύξουσα σειρά, πολλαπλάσιοι του τέσσερα και θα είναι από το 2000 έως το 2200

(π.χ. 2000, 2004 όχι όμως ο 2005..., ο 2051 κλπ). Ο δε βαθμός κάθε φοιτητή, με ένα δεκαδικό ψηφίο, θα προκύπτει τυχαία με τη χρήση της συνάρτησης rand() και κυμαίνεται στο διάστημα [0.0, 10.0] . Για την αρχικοποίηση της σειράς των τυχαίων να χρησιμοποιήσετε την συνάρτηση srand(AEM) όπου AEM ο αριθμός μητρώου σας.

Στη συνέχεια, το πρόγραμμά σας να ζητά να του πληκτρολογηθεί ένα AEM προκειμένου να εμφανίσει την βαθμολογία του φοιτητή. Η αναζήτησή του να γίνεται με τον αλγόριθμο της **δυναμικής αναζήτησης**. Ως εφαρμογή να εντοπίσετε και να εμφανίσετε στην οθόνη τον βαθμό του φοιτητή με AEM 2104. Να εμφανίζονται επίσης στην οθόνη αναλυτικά τα βήματα της αναζήτησης. Δηλαδή ποια θέση εξέτασε πρώτα, ποια στη συνέχεια και τελικά να εμφανιστεί όλη η διαδοχική σειρά των θέσεων που εξέτασε. Αν δεν εντοπίσει την καταχώριση να εμφανίσει μήνυμα ότι δεν υπάρχει τέτοια καταχώριση (π.χ. με AEM 2037).

Να βρείτε και να εμφανίσετε στην οθόνη πόσοι από τους καταχωρημένους φοιτητές έχουν μεγαλύτερο βαθμό από τον βαθμό δεδομένου φοιτητή. Εφαρμόστε το για τον φοιτητή με AEM 2112. Να εμφανισθεί επίσης ο αριθμός των συνολικών βημάτων εξέτασης που έκανε το πρόγραμμά σας για να απαντήσει στο ερώτημα αυτό, δηλαδή εξέτασε-έλεγε για παράδειγμα 23 περιπτώσεις.

Υπόδειξη : Στο πιο πάνω διάστημα οι αποδεκτοί AEM είναι 2000, 2004, 2008, 2012, 2016, 2020, 2024, 2028, 2032, 2036, 2040, 2044, 2048, 2052, 2056, 2060, 2064, 2068, 2072, 2076, 2080, 2084, 2088, 2092, 2096, 2100, 2104, 2108, 2112, 2116, 2120, 2124, 2128, 2132, 2136, 2140, 2144, 2148, 2152, 2156, 2160, 2164, 2168, 2172, 2176, 2180, 2184, 2188, 2192, 2196, 2200.

11. Τοπικός ραδιοφωνικός σταθμός, μετά τις 15:00, δεν απασχολεί προσωπικό και εκπέμπει μόνο ελληνική μουσική. Για τις ανάγκες του προγράμματος αυτού δημιουργεί κατάλληλη **διπλά συνδεδεμένη δυναμική κυκλική λίστα** τραγουδιών.

Αρχικά στη λίστα αυτή καταχωρούνται 1000 διαφορετικά τραγούδια που για τις ανάγκες της εφαρμογής δημιουργούνται με τυχαίο τρόπο και τυποποιούνται ως εξής ssccII, όπου το ss, cc και II διψήφιος αριθμός που χαρακτηρίζει αντίστοιχα τον τραγουδιστή, το συνθέτη και το στιχουργό και η τιμή του καθενός βρίσκεται στην περιοχή από 01-29.

Θεωρείστε ότι η αντιστοίχιση των αριθμών προκύπτει από τον πιο κάτω πίνακα :

ΚΩΔΙΚΟΣ	ΣΥΝΘΕΤΗΣ	ΣΤΙΧΟΥΡΓΟΣ	ΕΡΜΗΝΕΥΤΗΣ
01	Βαμβακάρης Μάρκος	Αγγελάκας Γιάννης	Αγγελάκας Γιάννης
02	Βαρδής Αντώνης	Αλκαίος Άλκης	Αλεξίου Χαρούλα
03	Ζαμπέτας Γιώργος	Άσιμος Νικόλας	Αρλέτα
04	Θεοδωράκης Μίκης	Βαμβακάρης Μάρκος	Βάνου Τζένη
05	Καλδάρas Απόστολος	Βίρβος Κώστας	Βέμπο Σοφία
06	Καραλής Γιάννης	Γκάτσος Νίκος	Γαβαλάς Πάνος
07	Καρβέλας Νίκος	Δασκαλόπουλος Άκος	Γαλάνη Δήμητρα

08	Κατσαρός Γιώργος	Ελευθερίου Μάνος	Διονυσίου Στράτος
09	Κουγιουμτζής Σταύρος	Ελύτης Οδυσσέας	Ζερβουδάκης Δημήτρης
10	Κραουνάκης Σταμάτης	Θεοφάνους Γιώργος	Καζαντζίδης Στέλιος
11	Λοΐζος Μάνος	Ιωαννίδης Αλκίνοος	Κωχ Μαρίζα
12	Μαρκόπουλος Γιάννης	Καββαδίας Νίκος	Μάλαμας Σωκράτης
13	Μαχαιρίτσας Λαυρέντης	Καλδάρας Απόστολος	Μαρινέλλα
14	Μητσάκης Γιώργος	Καμπανέλλης Ιάκωβος	Μητσιας Μανώλης
15	Μικρούτσικος Θάνος	Κολοκοτρώνης Χρήστος	Μοσχολιού Βίκυ
16	Μουσαφίρης Τάκης	Μητσάκης Γιώργος	Μπέλλου Σωτηρία
17	Νικολόπουλος Χρήστος	Μουσαφίρης Τάκης	Μπιθικώτσης Γρηγόρης
18	Ξαρχάκος Σταύρος	Νικολακοπούλου Λίνα	Νταλάρας Γιώργος
19	Πάνου Άκης	Πάνου Άκης	Ξύλινα Σπαθιά
20	Παραδοσιακό	Παπαγιαννοπούλου Ευτυχία	Ξυλούρης Νίκος
21	Πλέσσας Μίμης	Παπαδόπουλος Λευτέρης	Παγιουμτζής Στράτος
22	Πολυκανδριώτης Θανάσης	Παπακωνσταντίνου Θανάσης	Παπάζογλου Νίκος
23	Σαββόπουλος Διονύσης	Παραδοσιακό	Παπακωνσταντίνου Βασίλης
24	Σούκας Τάκης	Πυθαγόρας	Πάριος Γιάννης
25	Σπανός Γιάννης	Ρασούλης Μανώλης	Περπινιάδης Στελλάκης
26	Τσιτσάνης Βασίλης	Ρίτσος Γιάννης	Σαμίου Δόμνα
27	Χατζηνάσιος Γιώργος	Τσιτσάνης Βασίλης	Σιδηρόπουλος Παύλος
28	Χατζιδάκις Μάνος	Τσώτου Σώτια	Τσαουσάκης Πρόδρομος
29	Χιώτης Μανώλης	Φοίβος	Χατζής Κώστας

Για παράδειγμα το τραγούδι 021127 το συνέθεσε ο Βαρδής Αντώνης, τους στίχους έγραψε Αλκίνοος Ιωαννίδης και το τραγουδά ο Σιδηρόπουλος Παύλος.

Τη λίστα αυτή τη χειρίζεται ο ραδιοφωνικός σταθμός προκειμένου να καλύπτει το απογευματινό του πρόγραμμα.

Να δημιουργηθεί εφαρμογή σε C στην οποία να δημιουργείται η λίστα που περιγράφηκε. Για το σκοπό αυτό θα χρειαστεί να ορίσετε τη δομή κατάλληλου κόμβου με τα στοιχεία που χαρακτηρίζουν κάθε τραγούδι και στη συνέχεια να καταχωρήσετε σε αυτή 1000 τέτοιους κόμβους που θα αντιστοιχούν σε διαφορετικά τραγούδια.

Τα στοιχεία που θα καταχωρείτε θα προκύπτουν με τυχαίο τρόπο. Δηλαδή μέσω της rand() θα δημιουργείτε διαδοχικά τρεις τυχαίους αριθμούς μεταξύ 1 και 29, για κάθε ένα από τα τρία γνωρίσματα των τραγουδιών, συνθέτη, στιχουργό και ερμηνευτή. Στη σπάνια περίπτωση που εμφανίζεται και επόμενη φορά η ίδια τριάδα θα θεωρείτε ότι παρότι οι συντελεστές του τραγουδιού είναι οι ίδιοι ο κόμβος αφορά άλλο τίτλο τραγουδιού.

Η αρχικοποίηση της συνάρτησης τυχαιότητας θα γίνεται με τον AEM σας δηλαδή srand(AEM);

Μετά τη δημιουργία της λίστας, να προγραμματίσετε:

1. Για πέντε περίπου ώρες 100 τραγούδια, θα παίζει τυχαία επιλογή τραγουδιών. Ο ευκολότερος τρόπος είναι να ξεκινήσει από έναν τυχαίο κόμβο της λίστας και να διατρέξει τους επόμενους 100 κόμβους, μια και ο τρόπος που καταχωρήθηκαν οι κόμβοι είναι τυχαίοι.

2. Για την επόμενη ώρα του προγράμματος (20 τραγούδια) θα έχει αφιέρωμα στον συνθέτη Τσιτσάνη. Δηλαδή θα ακούγονται μόνο τραγούδια που θα έχουν ως συνθέτη τον Βασίλη Τσιτσάνη.
3. Στη συνέχεια τα επόμενα 20 τραγούδια, θα έχουν στιχουργό τη Λίνα Νικολακοπούλου.
4. Επειδή όμως οι ακροατές συνηθίζουν τη λίστα, και μαθαίνουν τη σειρά των τραγουδιών που ακούγονται, στην πρώτη ιδιαίτερα περίπτωση, για αυτό ο προγραμματιστής της λίστας θέλει να βελτιώσει κάπως την κατάσταση. Για αυτό σκέφτηκε να μην περιορίζεται μόνο στην τυχαία επιλογή του πρώτου τραγουδιού, με τον τυχαίο αρχικό κόμβο, αλλά σε δεύτερη φάση να επιλέγει την κατεύθυνση που θα διατρέχει τη λίστα, δηλαδή αν το επόμενο τραγούδι είναι ο επόμενος ή ο προηγούμενος κόμβος.
5. Να εμπλουτίσετε τη λίστα με άλλα 200 τραγούδια, με τον ίδιο τρόπο που καταχωρήθηκαν τα 1000 αρχικά.

12. Σε ένα υπερσύγχρονο Parking IX αυτοκινήτων που λειτουργεί πλήρως αυτοματοποιημένα η διαχείριση των αυτοκινήτων γίνεται με την αξιοποίηση **δυαδικού δέντρου αναζήτησης**. Με δεδομένο ότι οι συνολικές διαθέσιμες θέσεις του Parking είναι 200, ο προγραμματιστής για την υλοποίηση του δυαδικού δέντρου αναζήτησης επέλεξε την **στατική υλοποίησή** του μέσω κατάλληλου πίνακα.

Στον πίνακα καταχωρούνται **οι κόμβοι** του δέντρου. Η διαχείριση του δέντρου γίνεται με βάση τις πινακίδες των αυτοκινήτων. Επομένως κάθε κόμβος σχηματίζεται με βάση την πινακίδα του αυτοκινήτου που θα είναι και το κλειδί αναζήτησης, την θέση του Parking η οποία είναι με βάση τη σειρά προσέλευσης του οχήματος, δηλαδή κάθε αυτοκίνητο που προσέρχεται τοποθετείται στην πρώτη κενή θέση στάθμευσης, με άλλα λόγια οι θέσεις καλύπτονται με προτεραιότητα από το μικρότερο διαθέσιμο αριθμό προς το μεγαλύτερο. Αυτό σημαίνει ότι αρχικά το πρώτο θα τοποθετηθεί στην θέση στάθμευσης 1 το δεύτερο στην 2 και ούτω καθεξής. Ο κόμβος συμπεριλαμβάνει βέβαια και τους αριθμοδείκτες των θέσεων του αριστερού και δεξιού παιδιού του.

Στη συνέχεια με τη χρήση της τυχαίας συνάρτησης rand() να δημιουργήσετε 200 τυχαίους τετραψήφιους αριθμούς τους οποίους θα θεωρήσετε ως τις πινακίδες των αυτοκινήτων και θα δημιουργήσετε το αντίστοιχο δυαδικό δέντρο αναζήτησης, με τις θέσεις στάθμευσης του κάθε αυτοκινήτου. Για την αρχικοποίηση της σειράς των τυχαίων να χρησιμοποιήσετε την συνάρτηση srand(AEM) όπου AEM ο αριθμός μητρώου σας. Στη σπάνια περίπτωση που εμφανιστεί για δεύτερη φορά ο ίδιος τετραψήφιος ως πινακίδα αυτοκινήτου, θεωρείστε τον αμέσως επόμενο του.

Να εμφανίσετε στην οθόνη τα στοιχεία καταχώρησης αν διασχίσετε το δέντρο **με ενδοδιατεταγμένη** διάσχιση. Δηλαδή θα εμφανίζετε τον αριθμό κυκλοφορίας του αυτοκινήτου με τη θέση στάθμευσής του, π.χ. 1325 3, 1875 120

Στη συνέχεια αποχωρεί το αυτοκίνητο που **ήρθε πρώτο στο Parking**, δηλαδή αφαιρείται η ρίζα

του δέντρου, οπότε το δέντρο ανασυντάσσεται σύμφωνα με τους κανόνες του. Αν αυτό γίνει 10 φορές, δηλαδή αποχωρήσουν 10 συνολικά αυτοκίνητα, κάθε φορά η εκάστοτε ρίζα του δέντρου, να εμφανίσετε τους 10 αριθμούς κυκλοφορίας και τις θέσεις στάθμευσης που ελευθέρωσαν στο Parking;

13. Μια εταιρία πρατηρίων καυσίμων για τη διαχείριση των «καλών» πελατών της χρησιμοποιεί εταιρική ηλεκτρονική κάρτα μέλους. Κάθε πελάτης που καταχωρείται ως μέλος λαμβάνει έναν **επταψήφιο αριθμό μητρώου** τα δύο πρώτα ψηφία αφορούν την περιοχή (Καστοριά, Ηράκλειο, Θεσσαλονίκη κλπ) υπάρχουν 60 τέτοιες περιοχές (10-69), και τα πέντε επόμενα αφορούν στον πελάτη (π.χ. 5200067 αναφέρεται στον Σινάτκα που βρίσκεται στην Καστοριά περιοχή 52 και είναι ο 67 πελάτης της περιοχής).

Η εταιρία ευελπιστεί να χειρίζεται περίπου 100000 πελάτες και διάλεξε ο προγραμματιστής της να διαχειρίζεται τις πληροφορίες μέσω **πίνακα κατακερματισμού** ο οποίος αντιμετωπίζει τις πιθανές συγκρούσεις με τη μέθοδο του **διπλού κατακερματισμού**.

Για το λόγο αυτό επέλεξε να χρησιμοποιεί πίνακα **διάστασης 200003** και ως συνάρτηση **δεύτερου κατακερματισμού** την συνάρτηση που δίνει **βήμα=30011-K%30011** Κ το κλειδί ο επταψήφιος αριθμός μητρώου του μέλους.

Κατασκευάστε πρόγραμμα σε C το οποίο θα δημιουργεί με τη χρήση της τυχαίας συνάρτησης rand()τους 100000 τυχαίους αριθμούς μητρώου των πελατών και παράλληλα να καταχωρεί για τον καθένα τη συνολική κατανάλωση σε ευρώ που έχει κάνει μέχρι τώρα.

Για να καταχωρήσετε την κατανάλωση κάθε πελάτη δημιουργείτε τυχαίο δεκαδικό αριθμό από 500.0 μέχρι 15000.0€. Για την αρχικοποίηση της σειράς των τυχαίων να χρησιμοποιήσετε την συνάρτηση srand(AEM) όπου AEM ο αριθμός μητρώου σας.

Θα ζητείται ο αριθμός μητρώου του πελάτη, ο οποίος θα πληκτρολογείται και το πρόγραμμα θα τον εντοπίσει και θα εμφανίζει τη συνολική κατανάλωση που έχει κάνει. Αν δεν τον εντοπίσει, τότε να τον καταχωρεί με αρχικό λογαριασμό τα 537.5€ Να εμφανίζονται στην οθόνη όλοι οι αριθμοί μητρώου των πελατών και τα ποσά κατανάλωσής τους με τους οποίους ο αριθμός μητρώου που πληκτρολογήσατε έρχεται **σε σύγκρουση**.

Μπορείτε με την διαχείριση που επέλεξε ο προγραμματιστής να βρείτε εύκολα τη συνολική κατανάλωση που έκαναν οι πελάτες σε συγκεκριμένη περιοχή π.χ. στην περιοχή της Καστοριάς 52;

Υπόδειξη: Τους τυχαίους αριθμούς μητρώου μπορείτε να τους δημιουργήσετε με την έκφραση $N=100000*(10+\text{rand()} \% 60) + 1000*(\text{rand()} \% 100) + \text{rand()} \% 1000$. Τους λογαριασμούς δε με $B=500.0 + (\text{rand()} \% 14500)/10.0$

14. Σε ένα εφημερεύον νοσοκομείο προσέρχονται στα επείγοντα περιστατικά ασθενείς (**Patients**). Η σειρά με την οποία θα εξυπηρετηθούν εξαρτάται από το περιστατικό. Για αυτό κάθε ασθε-

νής κατά την είσοδό του στο νοσοκομείο λαμβάνει ως σειρά προτεραιότητας (**Priority**) έναν ακέραιο αριθμό από το 1 μέχρι το 100. Το 100 σημαίνει εξαιρετικά επείγον και είναι το περιστατικό που θα αντιμετωπιστεί πρώτο αμέσως μετά από αυτό που εξυπηρετείται αυτή τη στιγμή. Το περιστατικό με αριθμό προτεραιότητας 1 μπορεί να περιμένει μέχρι να αντιμετωπιστούν όλα τα πιο επείγοντα. Στην απίθανη περίπτωση που τύχει να περιμένουν ασθενείς με την ίδια προτεραιότητα, εξυπηρετείται πρώτος αυτός που έχει μικρότερο **PId** αριθμό προσέλευσης.

Όταν λοιπόν ξεκινά η εφημερία του νοσοκομείου προσέρχονται τα 20 αρχικά περιστατικά και λαμβάνουν αμέσως ο καθένας τους το δικό του PId που αυξάνει με τη σειρά κατά 1 και με τυχαίο τρόπο τον αριθμό προτεραιότητας Priority. Αρχικά λοιπόν, την χρονική στιγμή 00:00, θεωρείστε ότι περίμεναν την έναρξη της εφημερίας 20 ασθενείς που έλαβαν αμέσως σειρά προτεραιότητας, με τυχαίο τρόπο, και περίμεναν στην αίθουσα αναμονής την εξυπηρέτησή τους.

Υποθέστε ότι ο χρόνος αντιμετώπισης κάθε περιστατικού ακολουθεί τη συνάρτηση **CareTime**=100 + 10 x Priority δευτερόλεπτα.

Κάθε περιστατικό αντιμετωπίζεται άμεσα, αμέσως μετά την έξοδο του προηγούμενου ασθενή, και επίσης σε περίπτωση που στην αίθουσα αναμονής δεν περιμένει ασθενής, οι γιατροί αναλαμβάνουν να αντιμετωπίσουν κάθε νέο προσερχόμενο περιστατικό άσχετα του αριθμού προτεραιότητας που λαμβάνει. Με άλλα λόγια δεν υπάρχει αδρανής χρόνος για τους γιατρούς σε περίπτωση που περιμένει ασθενής.

Ωστόσο κατά τη διάρκεια της εφημερίας συνεχίζουν να προσέρχονται νέα περιστατικά με ρυθμό **ένας ασθενής κάθε 15 λεπτά**, δηλαδή **ArrivalTime**=00:15, 00:30..... 10:45, 11:00, οπότε και αυτός λαμβάνει αμέσως, με τυχαίο τρόπο, σειρά προτεραιότητας και προστίθεται στους ήδη υπάρχοντες.

Το νοσοκομείο δέχεται νέα περιστατικά μέχρι και 11 ώρες μετά την έναρξη της εφημερίας, ενώ συνεχίζει να εξυπηρετεί τα περιστατικά που περιμένουν, μέχρι να τελειώσουν.

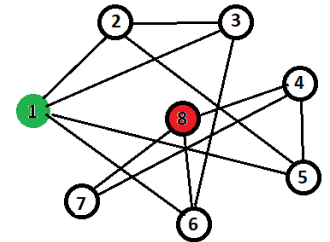
Να μοντελοποιηθεί το πρόβλημα. Κάθε ασθενής λαμβάνει έναν αύξοντα ακέραιο αριθμό ως ταυτότητα (PId) και την προτεραιότητά του (Priority). Για την εξυπηρέτησή του συστήματος δημιουργείται **σωρός μεγίστου**. Από τον οποίο καλείται κάθε φορά για εξυπηρέτηση η ρίζα του. Θα χρειασθεί λοιπόν να προγραμματιστούν οι κατάλληλες συναρτήσεις. Δηλαδή συνάρτηση δημιουργίας του σωρού, συνάρτηση καταχώρησης κόμβου, συνάρτηση αφαίρεσης κόμβου κλπ.

Το πρόγραμμα θα αρχικοποιεί την τυχαιότητα με τον AEM σας, θα εμφανίζει στην οθόνη τη σειρά εξυπηρέτησης των ασθενών, εμφανίζοντας την ταυτότητά του (PId), την ώρα που προσήλθε (ArrivalTime), την προτεραιότητα που έλαβε (Priority) και την ώρα που εξυπηρετήθηκε (ServiceTime).

Στο τέλος θα εμφανίζει συνολικά στοιχεία. Θα εμφανίζει το συνολικό αριθμό των περιστατικών που αντιμετώπισε, το μέσο χρόνο αντιμετώπισης κάθε περιστατικού και το συνολικό πιθανό χρόνο αδράνειας δηλαδή πόσο χρόνο κατά τη διάρκεια της εφημερίας (12 ώρες) δεν είχαν να αντιμετωπίσουν κανένα περιστατικό.

15. Να δημιουργήσετε πρόγραμμα σε C το οποίο να δημιουργεί απλό συνεκτικό¹ γράφο με N κορυφές των οποίων το όνομα θα είναι διαδοχικά 1,2 ..N-1, N. Στη συνέχεια με τυχαίο τρόπο να δημιουργήσετε τις διασυνδέσεις μεταξύ των κορυφών. Δηλαδή η πρώτη κορυφή θα επιλέγεται τυχαία με πόσες από τις υπόλοιπες N-1 κορυφές θα συνδεθεί άμεσα και στη συνέχεια, πάλι τυχαία, με ποιες από αυτές. Μετά για τη δεύτερη κορυφή το ίδιο, θα επιλέγεται τυχαία με πόσες από τις υπόλοιπες N-2 θα συνδεθεί άμεσα και στη συνέχεια με ποιες από αυτές και ούτω καθεξής.

Το πρόγραμμα να χειρίζεται τον γράφο με πίνακα γειτνίασης. Να δημιουργήσετε συνάρτηση που να προσδιορίζει την διαδρομή που ξεκινά από την κορυφή 1 και καταλήγει στην κορυφή N και διέρχεται από τον ελάχιστο αριθμό κορυφών του γράφου.



Για την αρχικοποίηση της συνάρτησης τυχαιότητας θα χρησιμοποιήσετε τον AEM σας, δηλαδή `srand(AEM);`

Το πρόγραμμα θα το τρέξετε για τυχαίο αριθμό κορυφών N, που κυμαίνονται από 6 μέχρι 10. Ο τυχαίος αυτός αριθμός N προκύπτει από τον μηχανισμό τυχαιότητας που ήδη έχει αρχικοποιηθεί.

Το πρόγραμμα θα εμφανίζει στην οθόνη τον πίνακα γειτνίασης του γράφου σας και την ή τις διαδρομές με την μορφή π.χ. 1->3->6->8 για N=8. Μπορεί οι διαδρομές αυτές να είναι περισσότερες από μία, τότε εμφανίζει όλες τις ισοδύναμες.

Για παράδειγμα στον γράφο του σχήματος έ-χουμε 8 συνολικά κορυφές οι οποίες διασυνδέονται μεταξύ τους με τυχαίο τρόπο. Από την κορυφή 1, όπως φαίνεται για να μεταβεί κά-ποιος στην 8 πρέπει να περάσει από ενδιάμεσους κόμβους. Υπάρχουν πολλές διαδρομές π.χ. η 1->2->3->6->8, 1->3->6->8, 1->6->8 η 1->2->5->4->8 και άλλες. Από όλες αυτές η διαδρομή που προτιμάται είναι η 1->6->8.

Υπόδειξη : Για να βρείτε τις διαδρομές αυτές μπορείτε να σκεφτείτε ως εξής: Στον πίνακα γειτνίασης ελέγχετε αν στην γραμμή που αντιστοιχεί η κορυφή 1 υπάρχει η κορυφή 8. Αν υπάρχει σημαίνει ότι υπάρχει απευθείας σύνδεση μεταξύ των κορυφών και δεν υπάρχει άλλη συντομότερη. Αν δεν υπάρχει η γραμμή 8 τότε μπορείτε να πάτε στην γραμμή που αντιστοιχεί στην κορυφή 8 και να εξετάσετε αν υπάρχει κοινή κορυφή η κορυφές που υπάρχουν με την γραμμή της 1. Αν υπάρχουν αυτές είναι οι ενδιάμεσες κορυφές για τη σύνδεση της 1 με την 8. Αν είναι μόνο μία τότε μία είναι και η ζητούμενη διαδρομή, με ενδιάμεση αυτή τη μία κορυφή, αν είναι περισσότερες τότε τόσες είναι και οι ισοδύναμες. Αν πάλι δεν υπάρχει καμία τότε πρέπει να προχωρήσει κανείς ένα ακόμα βήμα. Τι να κάνει δηλαδή; Να εξετάσει αν υπάρχουν κορυφές του γράφου της γραμμής 1 του πίνακα γειτνίασης που συνδέονται άμεσα με κορυφές του γράφου που υπάρχουν στην γραμμή της κορυφής 8. Αν υπάρχουν τότε η διαδρομές που προκύπτουν έχουν τη μορφή 1->κορυφή της γραμμής 1 -> κορυφή της γραμμής 8->8 π.χ.

¹ Συνεκτικός γράφος λέγεται ο γράφος του οποίου κάθε ζεύγος κορυφών συνδέεται με μια τουλάχιστον διαδρομή

στο σχήμα 1->3->6->8. Στην απίθανη ατυχή περίπτωση που δεν υπάρχει και τώρα διαδρομή συνεχίζει κανείς με παρόμοιο τρόπο.