

I. ΥΠΟΛΟΓΙΣΜΟΣ ΒΑΘΜΟΥ ΜΑΘΗΜΑΤΟΣ ΜΥΥ502

Σε αυτή την άσκηση θα δημιουργήσετε ένα πρόγραμμα το οποίο θα υπολογίζει τον τελικό βαθμό του μαθήματος ενός φοιτητή για το μάθημα του Προγραμματισμού Συστημάτων. Ο υπολογισμός θα γίνεται με βάση:

- α) Τον αριθμό απουσιών του φοιτητή στα εργαστήρια
- β) Τους βαθμούς στις προόδους 1 και 2
- γ) Τον βαθμό της τελικής εξέτασης

Από εσάς ζητείται να γραφτεί ένα πρόγραμμα που θα αποτελείται από μια συνάρτηση `main()`. Το πρόγραμμα θα ζητά από τον χρήστη τις παραπάνω πληροφορίες και στο τέλος θα εκτυπώνει τον τελικό βαθμό του φοιτητή με βάση τον εξής αλγόριθμο:

Αλγόριθμος Υπολογισμού Βαθμού

Δεδομένα: Αριθμός απουσιών A, Βαθμός 1ης προόδου PR1, Βαθμός 2ης προόδου PR2, Βαθμός τελικής εξέτασης TE.

1. Υπολογισμός βαθμού παρουσίας: $A = \{\text{καμιά απουσία:10, μία απουσία:5, } \geq 2 \text{ απουσίες:0}\}$
2. Υπολογισμός βαθμού εργαστηρίων: $BE = A(20\%) + PR1(40\%) + PR2(40\%)$
Αν $BE < 4.5$ τότε
Εκτύπωση "Αποτυχία στα εργαστήρια" και τερματισμός
Τέλος
3. Αν $TE < 4.5$ τότε
Εκτύπωση "Αποτυχία στην τελική εξέταση" και τερματισμός
Τέλος
4. Υπολογισμός τελικού βαθμού μαθήματος: $GR = BE(50\%) + TE(50\%)$
5. Εκτύπωση "Επιτυχία στο ΜΥΥ502 με βαθμό GR"

II. ΜΕΤΑΤΡΟΠΕΣ ΧΑΡΑΚΤΗΡΩΝ

Θα πρέπει να φτιάξετε ένα πρόγραμμα το οποίο θα διαβάζει ότι χαρακτήρες δίνει ο χρήστης από το πληκτρολόγιο και ακολουθεί τους παρακάτω κανόνες, μέχρι να δοθεί ο χαρακτήρας '\$' ή να συμπληρωθούν 30 χαρακτήρες συνολικά.

- Αν ο χαρακτήρας είναι κεφαλαίο γράμμα (από 'A' έως 'Z'), να τον μετατρέψει στον αντίστοιχο μικρό και να τον τυπώνει.
- Αν ο χαρακτήρας είναι μικρό γράμμα (από 'a' έως 'z'), να τυπώνει τη θέση του γράμματος στο Αγγλικό αλφάβητο (το 'a' είναι το 1, το 'b' είναι το 2, ..., το 'z' είναι το 26).
- Αν ο χαρακτήρας είναι ψηφίο (από '0' έως '9'), να τοποθετείται σε έναν πίνακα ακεραίων `nums`, ο οποίος αποθηκεύει την αξία όλων των ψηφίων που δίνονται (δηλαδή, αν π.χ. πληκτρολογηθεί ο χαρακτήρας '2', θα πρέπει να αποθηκευτεί ο ακέραιος 2).
- Οποιοσδήποτε άλλος χαρακτήρας απλώς εκτυπώνεται ως έχει.

Στο τέλος του προγράμματος θα πρέπει να εκτυπώνετε (με `printf()`) το περιεχόμενο του πίνακα `nums`.

ΠΡΟΣΟΧΗ: Προσπαθήστε να υλοποιήσετε την άσκηση χωρίς να ανατρέξετε στον πίνακα ASCII.

III. SUDOKU - ΈΛΕΓΧΟΣ ΕΓΚΥΡΟΤΗΤΑΣ ΛΥΣΗΣ

Το Sudoku είναι ένα παιχνίδι λογικής με αριθμούς όπου δίνεται ένας πίνακας 9x9 και πρέπει να συμπληρωθούν

τα κελιά με τέτοιο τρόπο ώστε κάθε στήλη, κάθε γραμμή και κάθε υποπίνακας-κουτάκι 3x3 να περιέχουν όλα τα ψηφία από το 1 μέχρι το 9. Για να είναι δυνατή η εύρεση μοναδικής λύσης υπάρχουν κελιά προσυμπληρωμένα.

Παράδειγμα προβλήματος Sudoku:

| | | | | | | | | |
|---|---|---|---|---|---|--|---|---|
| 5 | 3 | | | 7 | | | | |
| 6 | | | 1 | 9 | 5 | | | |
| | 9 | 8 | | | | | 6 | |
| 8 | | | | 6 | | | | 3 |
| 4 | | | 8 | | 3 | | | 1 |
| 7 | | | | 2 | | | | 6 |
| | 6 | | | | | | 2 | 8 |
| | | | 4 | 1 | 9 | | | 5 |
| | | | | 8 | | | 7 | 9 |

και της λύσης του:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 4 | 6 | 7 | 8 | 9 | 1 | 2 |
| 6 | 7 | 2 | 1 | 9 | 5 | 3 | 4 | 8 |
| 1 | 9 | 8 | 3 | 4 | 2 | 5 | 6 | 7 |
| 8 | 5 | 9 | 7 | 6 | 1 | 4 | 2 | 3 |
| 4 | 2 | 6 | 8 | 5 | 3 | 7 | 9 | 1 |
| 7 | 1 | 3 | 9 | 2 | 4 | 8 | 5 | 6 |
| 9 | 6 | 1 | 5 | 3 | 7 | 2 | 8 | 4 |
| 2 | 8 | 7 | 4 | 1 | 9 | 6 | 3 | 5 |
| 3 | 4 | 5 | 2 | 8 | 6 | 1 | 7 | 9 |

Σας ζητείται ένα πρόγραμμα το οποίο θα ελέγχει αν ένας πλήρως συμπληρωμένος πίνακας 9x9 αποτελεί έγκυρη λύση Sudoku. Ο έλεγχος θα γίνεται από μία συνάρτηση `check_solution()`, η οποία καλείται από την `main()`. Η συνάρτηση θα δέχεται τον πίνακα 9x9 ως όρισμα.

Δοκιμάστε το πρόγραμμά σας με λύσεις που θα βρείτε στο internet και είτε τις εισάγει ο χρήστης (με `scanf`) είτε τις ορίζετε εσείς ως αρχικοποιημένους πίνακες μέσα στο πρόγραμμά σας.

IV. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ: ΕΛΕΓΧΟΣ ΕΓΚΥΡΟΤΗΤΑΣ DIAGONAL SUDOKU

Υπάρχει μια παραλλαγή του Sudoku, το Diagonal Sudoku, όπου ισχύουν οι κανόνες του απλού Sudoku, όμως υπάρχει ένας επιπλέον περιορισμός: και στις δύο διαγώνιους του πίνακα 9x9 πρέπει να εμφανίζονται από μια φορά τα ψηφία 1 έως 9. Παράδειγμα:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 8 | 1 | 4 | 2 | 7 | 6 | 9 | 3 |
| 3 | 7 | 4 | 5 | 9 | 6 | 8 | 1 | 2 |
| 9 | 6 | 2 | 1 | 3 | 8 | 4 | 7 | 5 |
| 6 | 2 | 9 | 3 | 8 | 5 | 7 | 4 | 1 |
| 1 | 5 | 7 | 9 | 6 | 4 | 3 | 2 | 8 |
| 8 | 4 | 3 | 2 | 7 | 1 | 5 | 6 | 9 |
| 4 | 1 | 8 | 7 | 5 | 2 | 9 | 3 | 6 |
| 2 | 9 | 5 | 6 | 4 | 3 | 1 | 8 | 7 |
| 7 | 3 | 6 | 8 | 1 | 9 | 2 | 5 | 4 |

Τροποποιήστε τη συνάρτηση `check_solution()` ώστε να δουλεύει και για Diagonal Sudoku.

V. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ: ΣΥΝΑΡΤΗΣΗ ΕΚΤΥΠΩΣΗΣ ΑΡΙΘΜΟΥ ΣΕ ΔΥΑΔΙΚΗ ΜΟΡΦΗ

Η C μας παρέχει τη δυνατότητα να εκτυπώσουμε έναν αριθμό σε δεκαδική, δεκαεξαδική και οκταδική μορφή (μέσω των `%d`, `%x` και `%o` της `printf()`). Στην άσκηση αυτή ζητείται να προσθέσετε τη δυνατότητα εκτύπωσης αριθμού σε δυαδική μορφή, υλοποιώντας μία συνάρτηση `void printbin(unsigned int num)` η οποία δέχεται ως όρισμα έναν θετικό αριθμό και τυπώνει τα bits από τα οποία αποτελείται. Θεωρήστε ότι οι αριθμοί θα είναι πάντα θετικοί.

Για να το κάνετε αυτό, χρησιμοποιήστε τη γνωστή μέθοδο μετατροπής ενός αριθμού από το δεκαδικό στο δυαδικό σύστημα (διαδοχικές διαιρέσεις με το 2). Ορίστε μέσα στη συνάρτηση έναν πίνακα με 32 θέσεις μιας και οι `int` έχουν μέγεθος 32 bit.

Η `main()` θα πρέπει να περιέχει έναν βρόχο ο οποίος συνεχώς διαβάζει έναν αριθμό από το πληκτρολόγιο και τον τυπώνει καλώντας την `printbin()`, μέχρι να δοθεί ο αριθμός 502.

ΠΑΡΑΤΗΡΗΣΗ: επειδή οι ακέραιοι αριθμοί αποθηκεύονται σε αναπαράσταση συμπληρώματος ως προς 2, το πιο σημαντικό bit χρησιμοποιείται ως πρόσημο και επομένως μόνο τα 31 από τα 32 bits αποθηκεύουν το μέτρο του αριθμού. Για να εκμεταλλευτείτε και τα 32 bits, προτείνεται ο αριθμός που θα διαβάσετε καθώς και η παράμετρος της `printbin` να είναι `unsigned int` αντί για απλό `int`.