

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΔΠΜΣ – ΡΟΗ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ  
Η γλώσσα προγραμματισμού C

*ΕΡΓΑΣΤΗΡΙΟ 3: Πίνακες, βρόχοι, συναρτήσεις, αρχεία*

Μάιος 2019

Το σημερινό εργαστήριο έχει ως θέμα τους μονοδιάστατους αλλά και τους πολυδιάστατους πίνακες της γλώσσας C. Η C δεν υποστηρίζει άμεσα κάποια δομή πολυδιάστατων πινάκων, αλλά έμμεσα, επιτρέποντας τα στοιχεία των μονοδιάστατων πινάκων να είναι άλλοι μονοδιάστατοι πίνακες, κοκ. Θα χρησιμοποιήσετε φωλιασμένους βρόχους για την επεξεργασία τέτοιων πινάκων, ενώ θα δείτε για πρώτη φορά συναρτήσεις και αρχεία.

Τις παρακάτω ασκήσεις θα τις ολοκληρώσετε κατά προτίμηση μέχρι τη μεθεπόμενη εβδομάδα. Η παράδοσή τους θα γίνει όπως έγινε με τα προηγούμενα εργαστήρια. Υπενθυμίζεται η στοίχιση και ο σχολιασμός που πρέπει να έχουν τα προγράμματά σας για να είναι κατανοητά.

*Άσκηση 1: «Σχεδίαση γεωμετρικών σχημάτων σε δισδιάστατο πίνακα»*

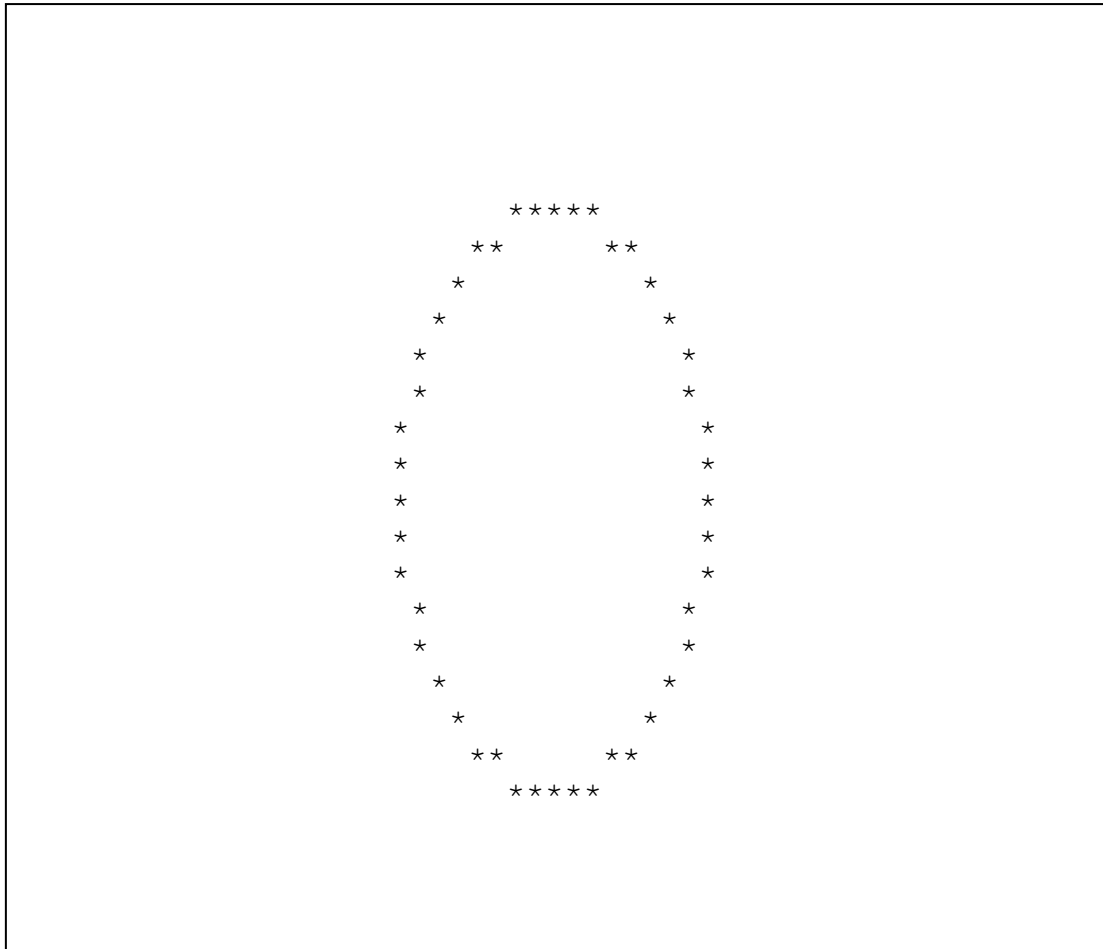
Δημιουργήστε την καινούργια εργασία *lab3*, καθώς και ένα καινούργιο αρχείο προγράμματος *ONOMA\_lab3.1.c*, όπου αντί για *ONOMA* θα βάλετε το ονοματεπώνυμό σας. Συνδέστε το με την εργασία *lab3*.

Στην άσκηση αυτή θα γράψετε ένα πρόγραμμα σχεδίασης απλών γεωμετρικών σχημάτων σε δισδιάστατο πίνακα. Πιο συγκεκριμένα, θα σχεδιάσετε τρίγωνα, τετράγωνα, παραλληλόγραμμα, τραπέζια και κύκλους, διαφόρων μεγεθών. Κάθε σχήμα θα πρέπει να εκτυπώνεται μέσα από κατάλληλη συνάρτηση, η οποία θα δέχεται ως παραμέτρους τα απαραίτητα στοιχεία που απαιτούνται για κάθε σχήμα, και θα επιστρέφει 0 για επιτυχημένη και 1 για αποτυχημένη εκτύπωση. Η εκτύπωση θα γίνεται με κάποιον χαρακτήρα – πχ. '\*'.  
Δηλώστε ένα δισδιάστατο πίνακα χαρακτήρων μεγέθους 60×60. Ξεκινήστε το πρόγραμμα με ερώτηση προς το χρήστη για το σχήμα που θέλει να εκτυπωθεί. Αφού διαβαστεί η απάντηση, θα πρέπει κατάλληλη εντολή *switch* να διαχειρίζεται κάθε σχήμα αναλυτικότερα, και στη συνέχεια να το εκτυπώνει. Όλα τα μεγέθη των σχημάτων θα πρέπει να είναι σε πλήθος χαρακτήρων που θα εκτυπωθούν.

Δηλώστε ένα δισδιάστατο πίνακα χαρακτήρων μεγέθους 60×60. Ξεκινήστε το πρόγραμμα με ερώτηση προς το χρήστη για το σχήμα που θέλει να εκτυπωθεί. Αφού διαβαστεί η απάντηση, θα πρέπει κατάλληλη εντολή *switch* να διαχειρίζεται κάθε σχήμα αναλυτικότερα, και στη συνέχεια να το εκτυπώνει. Όλα τα μεγέθη των σχημάτων θα πρέπει να είναι σε πλήθος χαρακτήρων που θα εκτυπωθούν.

Για τρίγωνο – όπου η μία ακμή λαμβάνεται ως βάση και τοποθετείται οριζόντια στον πίνακα, θα πρέπει να ζητείται εύρος βάσης, ύψος, καθώς και η σχετική απόσταση της επάνω κορυφής από την κάθετο στη βάση στην κάτω αριστερή κορυφή του τριγώνου. Για παράδειγμα, ένα όρθιο ισοσκελές τρίγωνο εύρους βάσης 16 και οποιουδήποτε ύψους, θα έχει σχετική απόσταση 8 χαρακτήρων, ενώ ένα ορθογώνιο τρίγωνο με την ορθή γωνία κάτω αριστερά και οποιουδήποτε μεγέθους θα έχει σχετική απόσταση 0





Όσο αφορά τον αλγόριθμο υλοποίησης, σε κάθε περίπτωση, θα καλείται η κατάλληλη συνάρτηση για την εκτύπωση του σχήματος, με παραμέτρους τα πιο πάνω στοιχεία. Η συνάρτηση θα πρέπει αρχικά να βρίσκει κάποιο σημείο κοντά στο κέντρο του πίνακα όπου θα γίνει η εκτύπωση, έτσι ώστε το σχήμα να μη βγει εκτός του πίνακα. Αν το σχήμα δε χωράει στον πίνακα, η συνάρτηση επιστρέφει 1. Στη συνέχεια, η συνάρτηση θα πρέπει να βρίσκει τα σημεία του πίνακα που θα πρέπει να σημειωθούν με το χαρακτηριστήρα σχεδίασης, θεωρώντας τον πίνακα σα σύστημα καρτεσιανών συντεταγμένων  $(x, y)$ . Για όλες τις περιπτώσεις εκτός του κύκλου, η συνάρτηση θα βρίσκει τις συντεταγμένες των κορυφών του σχήματος, και με βάση τις εξισώσεις των ακμών, θα βρίσκει εκείνα τα σημεία του πίνακα που πέφτουν πιο κοντά στις ακμές, προχωρώντας επαναληπτικά από κάθε κορυφή μέχρι την επόμενη. Ειδικά για κύκλο, θα πρέπει να ξεκινάει με τις συντεταγμένες του κέντρου, να βρίσκει τις συντεταγμένες των τεσσάρων σημείων πάνω, κάτω, δεξιά και αριστερά του κέντρου, και ξεκινώντας από το καθένα να προχωράει μέχρι το επόμενο επιλέγοντας τα σημεία που είναι πιο κοντά στη καμπύλη με βάση την εξίσωση του κύκλου.

Για τις ακμές μπορείτε να χρησιμοποιήσετε την ακόλουθη εξίσωση ευθείας που περνάει από δύο σημεία  $(x_1, y_1)$  και  $(x_2, y_2)$ :

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

και για τον κύκλο την εξίσωση κύκλου με κέντρο το σημείο  $(x_0, y_0)$  και ακτίνα  $r$ :

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

Σε όποια από τα σχήματα μπορείτε, να εκμεταλλευτείτε τη συμμετρία που έχουν, ώστε να ελαχιστοποιήσετε τους υπολογισμούς.

Για κάθε σχήμα που θα ολοκληρώνετε, εκτυπώστε το αποτέλεσμα στην οθόνη, καθώς και σε ένα αρχείο με κατάλληλο όνομα, ανοίγοντάς το με τη συνάρτηση *fopen()*, με δεύτερη παράμετρο "w" για εγγραφή, γράφοντας σε αυτό με τη συνάρτηση *fprintf()*, η οποία λειτουργεί όπως η *printf()*, με μια πρόσθετη πρώτη παράμετρο με το δείκτη στο αρχείο, και στο τέλος κλείνοντάς το με τη συνάρτηση *fclose()*.

Μεταφράστε και δοκιμάστε τον κώδικά σας για κάθε σχήμα του οποίου τον κώδικα ολοκληρώνετε.

Όταν γράψετε τον κώδικα για όλα τα σχήματα, αποθηκεύστε και υποβάλετε το αρχείο του προγράμματος, και προχωρήστε στην επόμενη άσκηση.

### *Άσκηση 2: «Κωδικοποίηση πραγματικών αριθμών»*

Σε κάθε υπολογιστή, όλες οι πληροφορίες που επεξεργάζεται κωδικοποιούνται σε δυαδική μορφή. Έτσι, οι μη προσημασμένοι ακέραιοι κωδικοποιούνται στην αντίστοιχη δυαδική τους αναπαράσταση. Για παράδειγμα, ο 6 κωδικοποιείται στη μορφή 110, ο 20 στη μορφή 10100, ο 1022 στη μορφή 111111110. Πριν από καθεμία από τις πιο πάνω μορφές, υπονοούνται τόσα μηδενικά όσα συμπληρώνουν το μέγεθος λέξης του επεξεργαστή, που συνήθως είναι 32 ή 64. Οι αρνητικοί ακέραιοι κωδικοποιούνται σε κάποια συμπληρωματική μορφή, που δε θα μας απασχολήσει εδώ.

Οι πραγματικοί αριθμοί συνήθως κωδικοποιούνται σε επιστημονική μορφή, δηλαδή σε μορφή με συντελεστή και εκθέτη. Πχ. στο δεκαδικό σύστημα μια τέτοια μορφή είναι η "2,3e-5", που υπονοεί το  $2,3 \times 10^{-5}$ . Επομένως, έχουμε έναν αριθμό δύο πεδίων, ένα πεδίο συντελεστή και ένα πεδίο εκθέτη, τα οποία συμπυκνώνονται στην ίδια λέξη πληροφορίας. Στην πράξη, υπάρχουν κάποιοι κανόνες σύμφωνα με τους οποίους ορίζεται η κωδικοποίηση πραγματικών αριθμών, όπως για παράδειγμα το πλήθος ψηφίων κάθε πεδίου, ο τρόπος κωδικοποίησης του προσήμου, η βάση του εκθέτη, κοκ. Ένας από αυτούς τους κανόνες αφορά την *κανονική* μορφή των αριθμών, και ορίζει πόσα ακριβώς *ακέραια* ψηφία έχει ο συντελεστής.

Στην άσκηση αυτή σας ζητείται να υλοποιήσετε τον κώδικα μετατροπής ενός πραγματικού αριθμού που διαβάζεται από το πληκτρολόγιο σε δεκαδική μορφή, στην ακόλουθη μορφή μεγέθους 32 δυαδικών ψηφίων (bits), από τα αριστερά προς τα δεξιά:

- 1 bit προσήμου συντελεστή (0 για μη αρνητικός, 1 για αρνητικός),
- 20 bits κανονικοποιημένου μέτρου συντελεστή, όπου όλα τα ψηφία είναι ακέραια, και το λιγότερο σημαντικό ψηφίο είναι μη μηδενικό, εκτός αν όλα τα ψηφία είναι μηδενικά,
- 1 bit προσήμου εκθέτη (0 για μη αρνητικός, 1 για αρνητικός), και
- 10 bits ακέραιου μέτρου εκθέτη.

Για απλούστευση, η κωδικοποίηση που ζητείται έχει ως βάση εκθέτη το 10, οπότε στην ουσία το μόνο που πρέπει να κάνετε είναι να διαβάσετε ξεχωριστά το συντελεστή και τον εκθέτη, να φέρετε τον αριθμό στην κανονική του μορφή (όλα τα ψηφία ακέραια και το λιγότερο σημαντικό ψηφίο μη μηδενικό *στη δεκαδική μορφή*), και να κωδικοποιήσετε τον τελικό συντελεστή και τον τελικό εκθέτη σε δυαδική μορφή, στα κατάλληλα δύο πεδία από τα 32 bits του αριθμού, λαμβάνοντας υπόψη και το πρόσημο του συντελεστή και του εκθέτη.

Για παράδειγμα, αν η είσοδος είναι η παραπάνω "2,3e-5", τότε για να βρείτε την κανονικοποιημένη μορφή θα κάνετε την πράξη:

$$2,3 * 10^{-5} = (23 * 10^{-1}) * 10^{-5} = 23 * 10^{-6}$$

οπότε στην κωδικοποιημένη δυαδική μορφή θα έχετε 1 ψηφίο '0' για το πρόσημο του συντελεστή, 20 ψηφία '000000000000000010111' για το συντελεστή (ο αριθμός 23), 1 ψηφίο '1' για το πρόσημο του εκθέτη, και 10 ψηφία '0000000110' για τον εκθέτη (ο αριθμός 6). Η τελική ζητούμενη μορφή του αριθμού αυτού είναι:

000000000000000001011110000000110

Αν τώρα η είσοδος είναι "230e-5", τότε για να βρείτε την κανονικοποιημένη μορφή θα κάνετε την πράξη:

$$230 * 10^{-5} = (23 * 10^1) * 10^{-5} = 23 * 10^{-4}$$

οπότε στην κωδικοποιημένη δυαδική μορφή θα έχετε 1 ψηφίο '0' για το πρόσημο του συντελεστή, 20 ψηφία '000000000000000010111' για το συντελεστή (ο αριθμός 23), 1 ψηφίο '1' για το πρόσημο του εκθέτη, και 10 ψηφία '0000000100' για τον εκθέτη (ο αριθμός 4). Η τελική ζητούμενη μορφή του αριθμού αυτού είναι:

000000000000000001011110000000100

Τα βήματα που θα πρέπει να ακολουθήσετε για να καταλήξετε στην κωδικοποιημένη δυαδική μορφή ενός πραγματικού αριθμού είναι επομένως τα εξής:

1. Ανάγνωση του αριθμού ως συμβολοσειρά.
2. Έλεγχος του προσήμου του συντελεστή.
3. Έλεγχος των ψηφίων του συντελεστή, ώστε μόλις βρείτε υποδιαστολή, να αρχίσετε να μετράτε τα κλασματικά ψηφία, και παράλληλα να μετράτε τα μηδενικά ψηφία που ακολουθούν το τελευταίο μη μηδενικό.
4. Μόλις βρείτε το χαρακτήρα 'e' ή 'E', έχει ολοκληρωθεί ο συντελεστής, οπότε αφού αφαιρέσετε τα τελικά μηδενικά ψηφία, ο συντελεστής μπορεί να μετατραπεί από συμβολοσειρά σε αριθμό.
5. Έλεγχος του προσήμου του εκθέτη.
6. Έλεγχος των ψηφίων του εκθέτη, ώστε μόλις φτάσετε στο τέλος της συμβολοσειράς, να μπορείτε να μετατρέψετε τον εκθέτη σε αριθμό.
7. Αφαίρεση από τον εκθέτη του πλήθους κλασματικών ψηφίων της εισόδου και πρόσθεση του πλήθους των μηδενικών ψηφίων που αφαιρέσατε, για την κανονικοποίηση του αριθμού.
8. Σύνθεση του αποτελέσματος με τη βοήθεια αριστερής ολίσθησης και λογικών πράξεων επιπέδου bit.

Δημιουργήστε λοιπόν ένα καινούργιο αρχείο προγράμματος και ονομάστε το *ONOMA\_lab3.2.c*, με το ονοματεπώνυμό σας ως ONOMA. Συνδέστε το με την εργασία *lab3* και αποσυνδέστε από αυτήν το αρχείο προγράμματος της προηγούμενης άσκησης.

Προχωρήστε με βάση τα παραπάνω, και αν θέλετε ακολουθήστε τις οδηγίες που ακολουθούν:

Εισάγετε τη δήλωση ενός καθολικού πίνακα χαρακτήρων μεγέθους 61 – έστω *str*, υποθέτοντας ότι το μέγιστο μήκος συμβολοσειράς πραγματικού αριθμού που θα σας δοθεί είναι 60 χαρακτήρες. Δηλώστε έναν μη προσημασμένο ακέραιο – έστω *number* – στον οποίο θα αποθηκεύσετε το αποτέλεσμα της κωδικοποίησης. Ο μη προσημασμένος ακέραιος τύπου `unsigned int` έχει μέγεθος 32 bits και χωράει ακριβώς αυτό που θέλουμε.

Δηλώστε επίσης 6 ακόμα ακέραιους, όπου θα αποθηκεύσετε το πρόσημο του συντελεστή, την τιμή του συντελεστή, το πλήθος των κλασματικών ψηφίων, το πλήθος των τελικών μηδενικών ψηφίων, το πρόσημο του εκθέτη και την τιμή του εκθέτη.

Ξεκινήστε τον κώδικα της συνάρτησης *main()* ζητώντας από το χρήστη να πληκτρολογήσει έναν πραγματικό αριθμό. Διαβάστε τον αριθμό ως συμβολοσειρά, με τη βοήθεια της συνάρτησης *scanf()* και μορφότυπο *%s*.

Ελέγξτε το πρόσημο του αριθμού. Αν δεν υπάρχει χαρακτήρας '+' ή '-' στην αρχή της συμβολοσειράς, τότε το πεδίο προσήμου συντελεστή θα είναι 0, αλλιώς το αντίστοιχο του χαρακτήρα προσήμου (0 για '+' και 1 για '-').

Χρησιμοποιήστε κατάλληλα την εντολή επανάληψης *while*, ώστε να διαβάσετε ένα-ένα τα ψηφία του συντελεστή, μέχρι την υποδιαστολή, ή μέχρι το χαρακτήρα εκθέτη 'e' ή 'E' αν δεν υπάρχει υποδιαστολή, ή μέχρι το τέλος της συμβολοσειράς αν δεν υπάρχει τίποτα από τα δύο. Αν βρείτε υποδιαστολή, συνεχίστε μετρώντας τα κλασματικά ψηφία που ακολουθούν. Παράλληλα, μετρήστε τα τελικά μηδενικά ψηφία του συντελεστή. (Μπορείτε να ξεκινάτε τη μέτρηση μόλις εμφανιστεί 0 και να την ακυρώνετε αν εμφανιστεί μη μηδενικό ψηφίο.)

Η μετατροπή του συντελεστή από συμβολοσειρά σε μη προσημασμένο αριθμό μπορεί να γίνει μέσα από την ίδια εντολή *while*, πολλαπλασιάζοντας τον αριθμό της προηγούμενης επανάληψης – αρχικά 0 – με το 10 και προσθέτοντας την τιμή του νέου ψηφίου, για όλα τα ψηφία, ακέραια και κλασματικά. Έτσι για παράδειγμα η συμβολοσειρά "2,3" θα γίνει ο αριθμός 23 με την πράξη:

$$((0) * 10 + 2) * 10 + 3$$

Αν υπάρχουν τελικά μηδενικά ψηφία στο συντελεστή, μπορείτε στο τέλος να διαιρέσετε με την αντίστοιχη δύναμη του 10.

Αποθηκεύστε το αποτέλεσμα – αλλά και τα ενδιάμεσα αθροίσματα – στην αντίστοιχη μεταβλητή της τιμής του συντελεστή που δηλώσατε πιο πάνω. Η κωδικοποίηση με την οποία αποθηκεύονται οι ακέραιοι είναι ήδη δυαδική και δε χρειάζεται να νοιαστείτε γι' αυτό! Θα πρέπει όμως να ελέγξετε αν ο συντελεστής χωράει σε 20 δυαδικά ψηφία, κι αυτό θα συμβαίνει αν η τιμή του είναι μικρότερη από το  $2^{20}$ . Αν αυτό δε συμβαίνει, θα πρέπει να εκτυπώσετε κατάλληλο μήνυμα και να τερματίσετε την εκτέλεση του προγράμματος.

Προσθέστε κώδικα εκτύπωσης του προσήμου και της τιμής – στο δεκαδικό σύστημα – του συντελεστή που υπολογίσατε.

Μεταφράστε και δοκιμάστε τον κώδικά σας.

Αν ακολουθεί εκθέτης, συνεχίστε με τον εκθέτη. Παρακάμψτε το χαρακτήρα 'e' ή 'E', και ελέγξτε το πρόσημο του εκθέτη, δίνοντας κατάλληλη τιμή στην αντίστοιχη μεταβλητή. Μπορείτε να μετατρέψετε τη συμβολοσειρά του εκθέτη σε μη προσημασμένο αριθμό, όπως κάνατε με το συντελεστή.

Αφού βρείτε την τιμή της συμβολοσειράς του εκθέτη, υπολογίστε τον τελικό εκθέτη, αφαιρώντας το πλήθος των κλασματικών ψηφίων του αρχικού συντελεστή, και προσθέτοντας το πλήθος των τελικών μηδενικών ψηφίων του αρχικού συντελεστή. Προσέξτε ότι αν ο εκθέτης έχει αρνητικό πρόσημο, η αφαίρεση και η πρόσθεση θα υλοποιηθούν αντίστροφα! Στο τέλος θα πρέπει να ελέγξετε αν προκύψει αρνητικό αποτέλεσμα, οπότε θα διορθώσετε κατάλληλα τη μεταβλητή προσήμου του εκθέτη και θα πάρετε την αντίθετη τιμή ως τελική τιμή εκθέτη.

Όπως και με το συντελεστή, η τιμή που παράγεται αποθηκεύεται αυτόματα ως δυαδική, και το μόνο που πρέπει να κάνετε ακόμα, είναι να ελέγξετε αν η τιμή χωράει σε 10 δυαδικά ψηφία, ώστε αν δε χωράει, να τερματίσετε την εκτέλεση του προγράμματος με κατάλληλο μήνυμα.

Προσθέστε κώδικα εκτύπωσης του προσήμου και της τιμής – στο δεκαδικό σύστημα – του εκθέτη.

Μεταφράστε και δοκιμάστε τον κώδικά σας.

Τώρα έχει ολοκληρωθεί ο βρόχος ελέγχου των ψηφίων της συμβολοσειράς *str*. Για την τελική σύνθεση του αποτελέσματος, θα χρησιμοποιήσετε τη μεταβλητή *number*, αρχικοποιώντας την με το πρόσημο του συντελεστή. Για να βάλετε στη συνέχεια την τιμή του συντελεστή, θα πρέπει να αφήσετε 20 θέσεις, κι επομένως θα κάνετε αριστερή ολίσθηση της μεταβλητής κατά 20 bits, χρησιμοποιώντας τον τελεστή '<<' της C, ως εξής:

```
number <<= 20
```

Για να συμπληρώσετε την τιμή του συντελεστή – έχοντας ελέγξει ότι δεν ξεπερνάει το μέγεθος των 20 δυαδικών ψηφίων, θα κάνετε μια απλή λογική πράξη με τον τελεστή '|' μεταξύ του *number* και της μεταβλητής που περιέχει την τιμή του συντελεστή.

Συνεχίζοντας όμοια, θα ενσωματώσετε το πρόσημο του εκθέτη με ολίσθηση κατά 1 θέση και νέο λογικό '|', και την τιμή του εκθέτη με ολίσθηση κατά 10 θέσεις και το τελευταίο λογικό '|'.

Για να εκτυπώσετε το αποτέλεσμα σε δυαδική μορφή θα πρέπει να γράψετε ένα βρόχο εντολής *for*, που θα εκτελεί ακριβώς 32 επαναλήψεις, ελέγχοντας και εκτυπώνοντας ένα-ένα τα δυαδικά ψηφία της κωδικοποίησης από αριστερά προς τα δεξιά. Μια καλή ιδέα είναι να χρησιμοποιήσετε μια τοπική ακέραια μεταβλητή τύπου *unsigned int* ως μάσκα, αρχικοποιημένη σε  $2^{31}$ , και σε κάθε επανάληψη να εκτελείτε λογικό '&' με τη μεταβλητή *number*, για να ελέγχετε αν το ψηφίο είναι 0 ή 1, και στη συνέχεια να εκτυπώνετε το χαρακτήρα '0' ή '1', αντίστοιχα, και να ολισθαίνετε τη μάσκα μια θέση δεξιά.

Μεταφράστε και δοκιμάστε τον κώδικά σας.

Δημιουργήστε τέλος έναν εξωτερικό βρόχο εντολής *while*, ο οποίος να επαναλαμβάνει την όλη διαδικασία, για κάθε νέο αριθμό που δίνεται στην είσοδο. Ο βρόχος μπορεί να έχει τη μορφή «*while (1)*» και να τερματίζεται με εντολή *break* αν δίνεται κενή συμβολοσειρά (μόνο Enter). Προσέξτε τις μεταβλητές που θέλουν αρχικοποίηση, ώστε να τους δίνετε αρχική τιμή μέσα στο βρόχο.

Αποθηκεύστε και υποβάλετε το αρχείο του προγράμματος.