

Ασκήσεις 6 - Τελεστές bit-προς-bit

Άσκηση 6.1

Να γραφεί πρόγραμμα που να διαβάζει έναν χαρακτήρα και θα δείχνει: τον χαρακτήρα, τον κωδικό του (ASCII) στο δεκαδικό, στο δεκαεξαδικό και στο δυαδικό σύστημα. Π.χ.

Πάτησε πλήκτρο : A

'A' = 65 = 0x41 = 01000001

Οδηγίες: Η έξοδος ενός αριθμού στην δεκαεξαδική του μορφή μπορεί να γίνει με τον τροποποιητή hex. Για την έξοδο σε δυαδική μορφή να δημιουργήσετε μια συνάρτηση που θα παίρνει ένα δεδομένο τύπου char (δηλ. έναν ακέραιο ενός byte) και θα επιστρέφει μια συμβολοσειρά των οκτώ χαρακτήρων η οποία θα περιέχει την δυαδική μορφή του αριθμού αυτού. Η συμβολοσειρά αυτή μπορεί, στην συνέχεια, να τυπωθεί σε οποιοδήποτε σημείο του προγράμματος είναι επιθυμητό. Η συμβολοσειρά μπορεί να είναι μια παράμετρος της συνάρτησης ή να επιστρέφεται με return από τη συνάρτηση.

Οδηγίες για τη συνάρτηση: Θα πρέπει να απομονώνεται κάθε φορά ένα bit του αριθμού και, ανάλογα με την τιμή του, θα τίθεται στο αντίστοιχο στοιχείο της συμβολοσειράς η τιμή '1' ή η τιμή '0'.

Π.χ. για να απομονωθεί το b_7 της μεταβλητής c και να περάσει, μόνο αυτό, σε μια μεταβλητή z, πρέπει να γίνει bitwise and με τη μάσκα 10000000_2 , δηλ. με το 128 ή ισοδύναμα το 0x80.

$z = c \& 128$; Αν το z είναι 0 τότε $s[0] = '0'$ αλλιώς $s[0] = '1'$;

Μετά πρέπει να γίνει έλεγχος του επόμενου bit, του b_6 . Η μάσκα τώρα είναι 01000000_2 , δηλ. το 64 ή ισοδύναμα το 0x40

$z = c \& 64$; Αν το z είναι 0 τότε $s[1] = '0'$ αλλιώς $s[1] = '1'$;

μετά η μάσκα θα πρέπει γίνει 32 (= 00100000_2) για να ελεγχθεί το bit b^5 ,

$z = c \& 32$; Αν το z είναι 0 τότε $s[2] = '0'$ αλλιώς $s[2] = '1'$;

κ.ο.κ. μέχρι να ελεγχθεί το λιγότερο σημαντικό bit, όπου η μάσκα είναι 00000001_2 , δηλ. το 1

$z = c \& 1$; Αν το z είναι 0 τότε $s[7] = '0'$ αλλιώς $s[7] = '1'$;

Συνεπώς θα χρειαστεί ένας βρόχος όπου η αρχική τιμή της μάσκας θα είναι 128 και σε κάθε επανάληψη θα ολισθαίνει προς τα δεξιά κατά μια θέση όσο είναι μεγαλύτερη του μηδενός και κάθε φορά θα γίνεται η πράξη and και ο έλεγχος.

Αν πρόκειται η συμβολοσειρά να χρησιμοποιηθεί σαν πίνακας όπως παραπάνω τότε θα πρέπει να δεσμευτεί κατάλληλος χώρος δηλ. 8 bytes. Εναλλακτικά αφού είναι αντικείμενο της κλάσης string, μπορεί να ξεκινήσουμε από ένα αρχικά κενό string και σε κάθε επανάληψη να προστίθεται το '1' ή '0' ανάλογα με το αποτέλεσμα του ελέγχου, δηλ.

string s = "" ; // κενό string

$z = c \& 128$; Αν το z είναι 0 τότε $s = s + '0'$ αλλιώς $s = s + '1'$;

$z = c \& 64$; Αν το z είναι 0 τότε $s = s + '0'$ αλλιώς $s = s + '1'$;

κ.ο.κ.

Β' μέρος: Για τους χαρακτήρες από 'θ' μέχρι 'z' να εμφανίζετε τον εκάστοτε χαρακτήρα, τον κωδικό του (ASCII) στο δεκαδικό, στο δεκαεξαδικό και στο δυαδικό σύστημα όπως παραπάνω.

θ = 48 = 0x30 = 00110000

1 = 49 = 0x31 = 00110001

2 = 50 = 0x32 = 00110010

..

Γ' μέρος: Να εισάγετε έναν χαρακτήρα, να τον εμφανίσετε στις τέσσερις μορφές όπως παραπάνω (δηλ. χαρακτήρα και κωδικό ASCII σε δεκαδικό, δεκαεξαδικό και δυαδικό) να θέσετε το bit-5 ίσο με 1 και να τον εμφανίσετε ξανά στις τέσσερις μορφές όπως παραπάνω για να διαπιστώσετε την αλλαγή. Δοκιμάστε να εισάγετε έναν κεφαλαίο χαρακτήρα και δείτε το αποτέλεσμα.