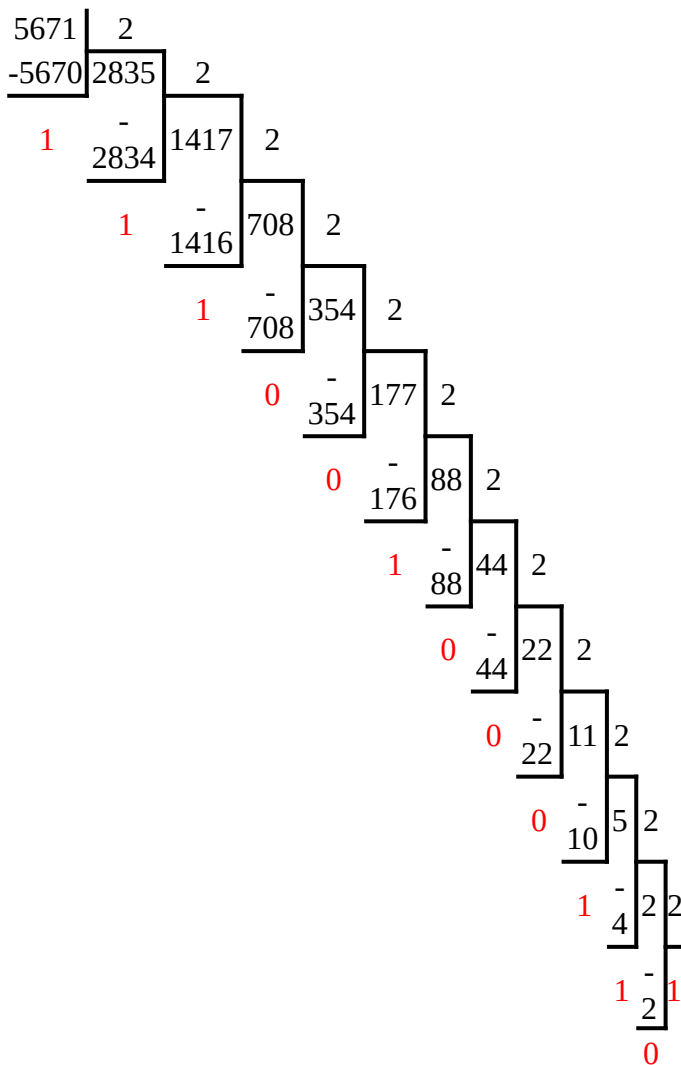


5. Να μετατρέψετε το δεκαδικό αριθμό 5671 σε δυαδικό



67. Τι γνωρίζετε για την κωδικοποίηση UNICODE;

Το Unicode είναι μια κωδικοποίηση γραμματοσειράς που αντιστοιχεί τους χαρακτήρες ως ακέραιους αριθμούς κώδικα μηχανής. Αυτό το πρότυπο είναι παρόμοιο με το ASCII (American Standard Code for Information Interchange) αλλά διαφέρει σε κάποια σημεία και ως προς τον αριθμό των χαρακτήρων που μπορεί να αναπαραστήσει. Το Unicode χρησιμοποιεί δύο bytes (16 bits) για να αναπαραστήσει τον κάθε χαρακτήρα αντί του ενός byte (8 bits) που χρησιμοποιεί το ASCII. Με αυτόν τον τρόπο το Unicode μπορεί να αντιπροσωπεύσει περισσότερους από 65.000 χαρακτήρες σε αντίθεση με τους 256 που μπορεί να χειριστεί το ASCII, καθιστώντας το ικανό να περιλαμβάνει σχεδόν όλα τα συστήματα γραφής που είναι σε χρήση σήμερα. Εκτός από χαρακτήρες όλων των γλωσσών του κόσμου το Unicode περιλαμβάνει και άλλα ιστορικά και εξαφανισμένα αλφάβητα όπως τα Αιγυπτιακά κυρίως για ακαδημαϊκούς σκοπούς αλλά και σύμβολα για την μουσική και τα μαθηματικά. Τα τελευταία χρόνια επίσης περιλαμβάνει και εικονίδια όπως τα emoji.

66. Τι γνωρίζετε για την κωδικοποίηση ASCII;

Ο κώδικας ASCII (American Standard Code for Information Interchange, Αμερικανικός Πρότυπος Κώδικας για Ανταλλαγή Πληροφοριών, είναι ένα κωδικοποιημένο σύνολο χαρακτήρων του λατινικού αλφάβητου όπως αυτό χρησιμοποιείται σήμερα στην Αγγλική γλώσσα και σε άλλες δυτικοευρωπαϊκές γλώσσες. Χρησιμοποιείται για αναπαράσταση κειμένου στους υπολογιστές, σε συσκευές τηλεπικοινωνίας, καθώς και σε άλλες συσκευές που δουλεύουν με κείμενο. Οι περισσότερες σύγχρονες κωδικοποιήσεις χαρακτήρων βασίζονται στον ASCII, αν και υποστηρίζουν πολύ περισσότερους χαρακτήρες.

Ο κώδικας ASCII απεικονίζει μία ομάδα 128 χαρακτήρων σε δυαδικούς αριθμούς μήκους 8 bit. Οι αντίστοιχοι ακέραιοι αριθμοί, στους οποίους απεικονίζονται οι χαρακτήρες είναι από το 0 ως τον 127. Το όγδοο bit στον κώδικα ASCII μπορεί να έχει μόνιμα την τιμή 1 ή 0, οπότε ονομάζεται *stick bit* (σταθερό bit). Επίσης μπορεί να χρησιμοποιηθεί για την ανίχνευση λαθών, οπότε ονομάζεται *parity bit* (bit ομοτιμίας). Πρώτη η IBM χρησιμοποίησε το όγδοο bit στον κώδικα ASCII για να απεικονίσει χαρακτήρες σε χώρες με διαφορετικό αλφάβητο. Η χρήση του τελευταίου bit για τους χαρακτήρες άλλων χωρών επιτρέπει την απεικόνιση 128 επιπλέον διαφορετικών χαρακτήρων.

9. Να μετατραπούν στο δεκαδικό σύστημα οι δυαδικοί αριθμοί:

α. 00111010

$$0 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 0 \cdot 128 + 0 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 0 + 0 + 32 + 16 + 8 + 0 + 2 + 0 = 58_{10}$$

β. 11001110

$$1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 1 \cdot 128 + 1 \cdot 64 + 0 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 128 + 64 + 0 + 0 + 8 + 4 + 2 + 0 = 206_{10}$$

γ. 1111100000001111

$$1 \cdot 2^{15} + 1 \cdot 2^{14} + 1 \cdot 2^{13} + 1 \cdot 2^{12} + 1 \cdot 2^{11} + 0 \cdot 2^{10} + 0 \cdot 2^9 + 0 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 32768 + 1 \cdot 16384 + 1 \cdot 8192 + 1 \cdot 4096 + 1 \cdot 2048 + 0 \cdot 1024 + 0 \cdot 512 + 0 \cdot 256 + 0 \cdot 128 + 0 \cdot 64 + 0 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 32768 + 16384 + 8192 + 4096 + 2048 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 8 + 4 + 2 + 1 = 63503_{10}$$

δ. 1001100110011001

$$1 \cdot 2^{15} + 0 \cdot 2^{14} + 0 \cdot 2^{13} + 1 \cdot 2^{12} + 1 \cdot 2^{11} + 0 \cdot 2^{10} + 0 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 =$$

$$1 \cdot 32768 + 0 \cdot 16384 + 0 \cdot 8192 + 1 \cdot 4096 + 1 \cdot 2048 + 0 \cdot 1024 + 0 \cdot 512 + 1 \cdot 256 + 1 \cdot 128 + 0 \cdot 64 + 0 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 =$$

$$32768 + 0 + 0 + 4096 + 2048 + 0 + 0 + 256 + 128 + 0 + 0 + 16 + 8 + 0 + 0 + 1 = 39321_{10}$$

15. Να μετατραπούν σε δεκαεξαδικούς και οκταδικούς οι δυαδικοί αριθμοί:

α. 0101

$$0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 0 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 0 + 4 + 0 + 1 = 5_{10}$$

$$0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 0 \cdot 32 + 0 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 0 + 0 + 0 + 4 + 0 + 1 = 5_{10}$$

β. 1101

$$001101_2 = 001 \ 101 = 001_{(=1)} \ 101_{(=5)} = 15_8$$

$$11012 = 1101 = 1101(=D) = D16$$

γ. 10111100

$$1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 1 \cdot 128 + 0 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 =$$

$$128 + 0 + 32 + 16 + 8 + 4 + 0 + 0 = 188_{10}$$

$$0 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 0 \cdot 256 + 1 \cdot 128 + 0 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 =$$

$$0 + 128 + 0 + 32 + 16 + 8 + 4 + 0 + 0 = 188_{10}$$

δ. 100011001

$$1 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 256 + 0 \cdot 128 + 0 \cdot 64 + 0 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 =$$

$$256 + 0 + 0 + 0 + 16 + 8 + 0 + 0 + 1 = 281_{10}$$

$$0 \cdot 2^{11} + 0 \cdot 2^{10} + 0 \cdot 2^9 + 1 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 =$$

$$0 \cdot 2048 + 0 \cdot 1024 + 0 \cdot 512 + 1 \cdot 256 + 0 \cdot 128 + 0 \cdot 64 + 0 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 =$$

$$0 + 0 + 0 + 256 + 0 + 0 + 0 + 16 + 8 + 0 + 0 + 1 = 281_{10}$$

ε. 11110111

$$1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 128 + 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 =$$

$$128 + 64 + 32 + 16 + 0 + 4 + 2 + 1 = 247_{10}$$

$$0 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 0 \cdot 256 + 1 \cdot 128 + 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 =$$

$$0 + 128 + 64 + 32 + 16 + 0 + 4 + 2 + 1 = 247_{10}$$

16. Να μετατραπούν στο δεκαδικό σύστημα οι δυαδικοί αριθμοί:

α. 01111010

$$0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 0 \cdot 128 + 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 =$$

$$0 + 64 + 32 + 16 + 8 + 0 + 2 + 0 = 122_{10}$$

β. 11001110

$$1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 1 \cdot 128 + 1 \cdot 64 + 0 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 =$$

$$128 + 64 + 0 + 0 + 8 + 4 + 2 + 0 = 206_{10}$$

γ. 1111100000001111

$$1 \cdot 2^{15} + 1 \cdot 2^{14} + 1 \cdot 2^{13} + 1 \cdot 2^{12} + 1 \cdot 2^{11} + 0 \cdot 2^{10} + 0 \cdot 2^9 + 0 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$$

$$1 \cdot 32768 + 1 \cdot 16384 + 1 \cdot 8192 + 1 \cdot 4096 + 1 \cdot 2048 + 0 \cdot 1024 + 0 \cdot 512 + 0 \cdot 256 + 0 \cdot 128 + 0 \cdot 64 + 0 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 =$$

$$32768 + 16384 + 8192 + 4096 + 2048 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 8 + 4 + 2 + 1 = 63503_{10}$$

δ. 1001100110011001

$$1 \cdot 2^{15} + 0 \cdot 2^{14} + 0 \cdot 2^{13} + 1 \cdot 2^{12} + 1 \cdot 2^{11} + 0 \cdot 2^{10} + 0 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 =$$

$$1 \cdot 32768 + 0 \cdot 16384 + 0 \cdot 8192 + 1 \cdot 4096 + 1 \cdot 2048 + 0 \cdot 1024 + 0 \cdot 512 + 1 \cdot 256 + 1 \cdot 128 + 0 \cdot 64 + 0 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 =$$

$$32768 + 0 + 0 + 4096 + 2048 + 0 + 0 + 256 + 128 + 0 + 0 + 16 + 8 + 0 + 0 + 1 = 39321_{10}$$

ε. 1100110011001100

$$1 \cdot 2^{15} + 1 \cdot 2^{14} + 0 \cdot 2^{13} + 0 \cdot 2^{12} + 1 \cdot 2^{11} + 1 \cdot 2^{10} + 0 \cdot 2^9 + 0 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 =$$

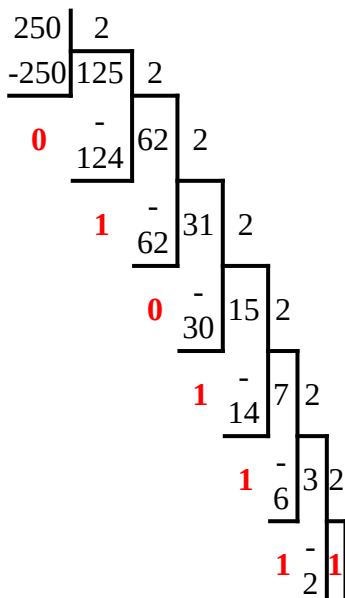
$$1 \cdot 32768 + 1 \cdot 16384 + 0 \cdot 8192 + 0 \cdot 4096 + 1 \cdot 2048 + 1 \cdot 1024 + 0 \cdot 512 + 0 \cdot 256 + 1 \cdot 128 + 1 \cdot 64 + 0 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 =$$

$$32768 + 16384 + 0 + 0 + 2048 + 1024 + 0 + 0 + 128 + 64 + 0 + 0 + 8 + 4 + 0 + 0 = 52428_{10}$$

17. Να μετατραπούν σε δυαδικούς οι δεκαεξαδικοί αριθμοί:

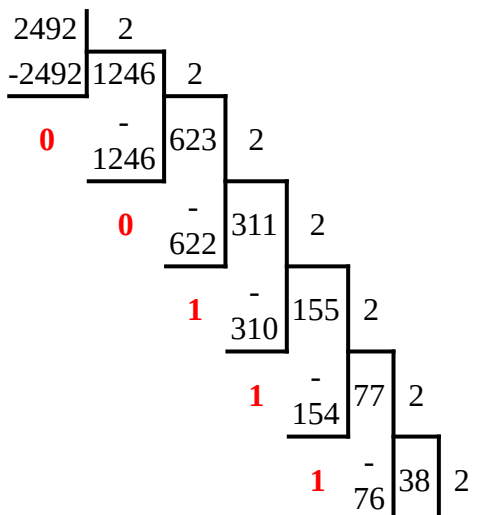
α. FA

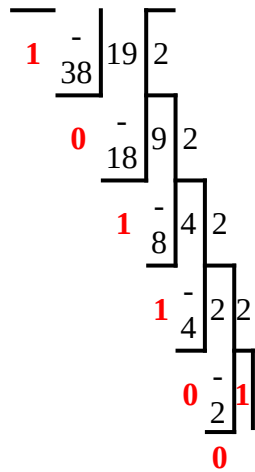
$$15 \cdot 16^1 + 10 \cdot 16^0 = 15 \cdot 16 + 10 \cdot 1 = 240 + 10 = 250_{10}$$



β. 9BC

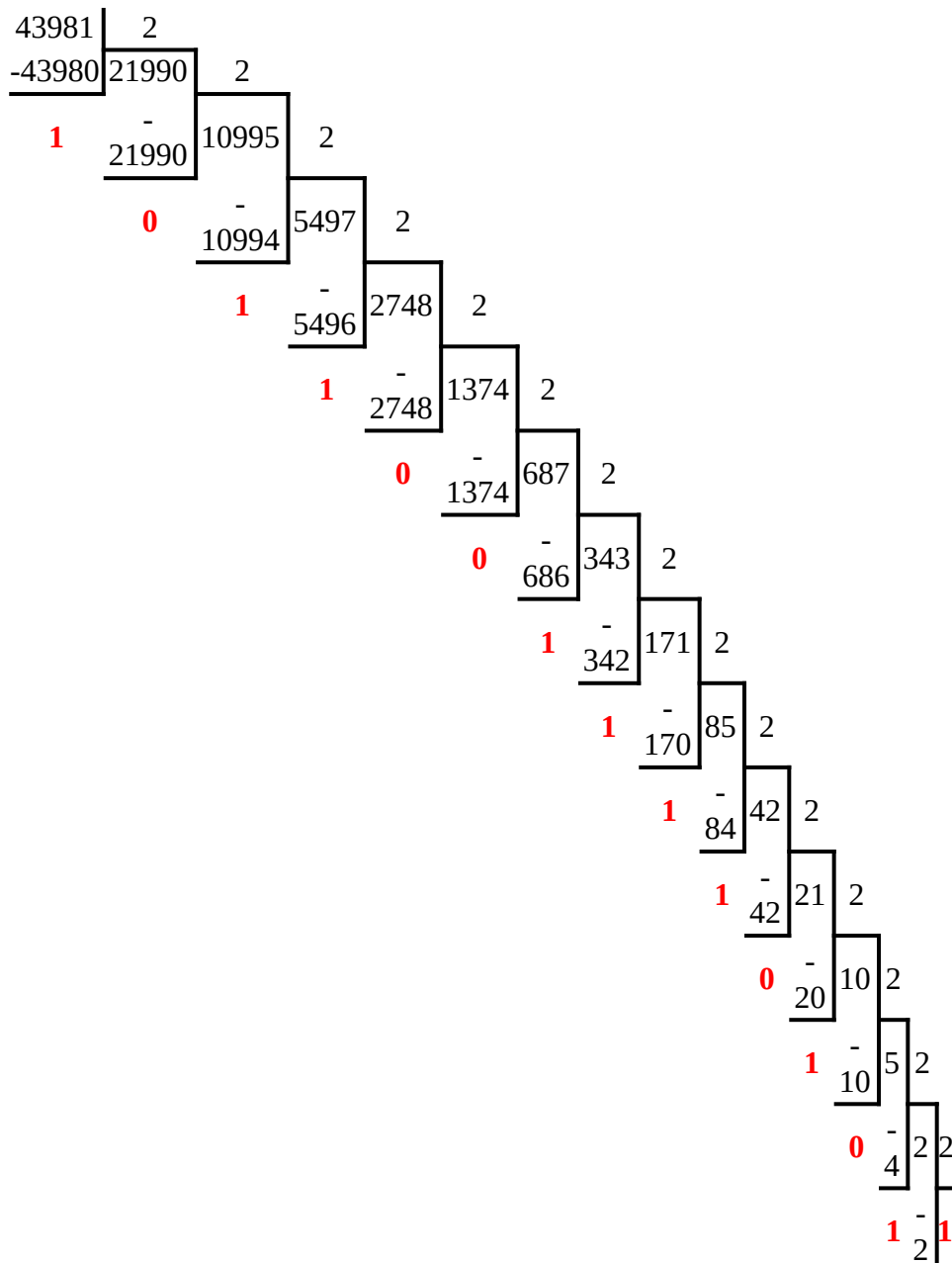
$$9 \cdot 16^2 + 11 \cdot 16^1 + 12 \cdot 16^0 = 9 \cdot 256 + 11 \cdot 16 + 12 \cdot 1 = 2304 + 176 + 12 = 2492_{10}$$





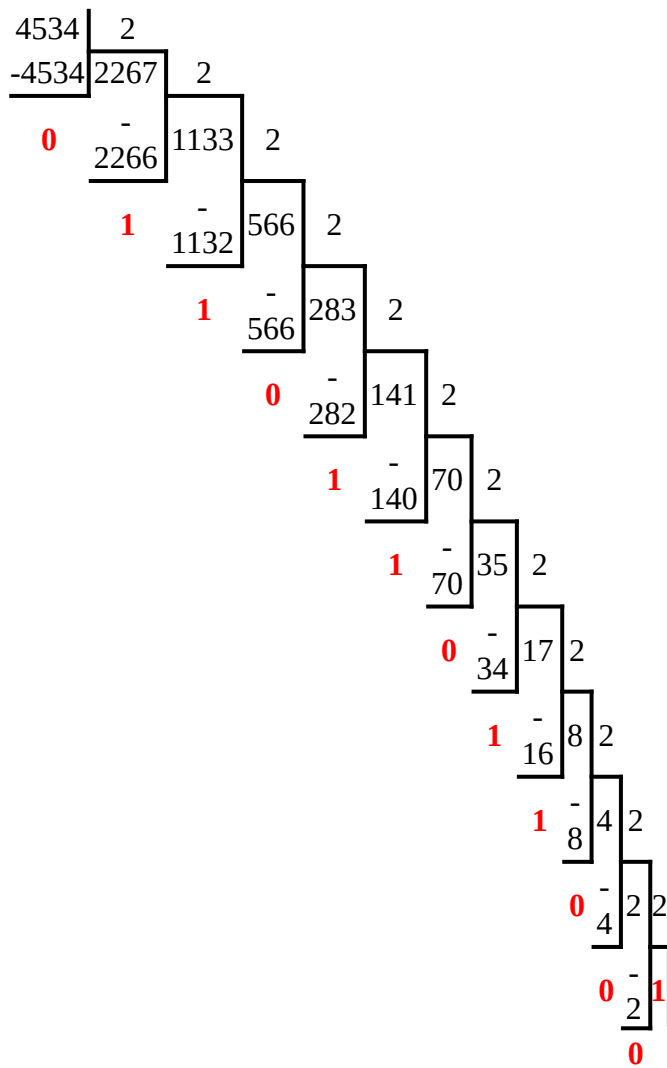
γ .ABCD

$$10 \cdot 16^3 + 11 \cdot 16^2 + 12 \cdot 16^1 + 13 \cdot 16^0 = 10 \cdot 4096 + 11 \cdot 256 + 12 \cdot 16 + 13 \cdot 1 = 40960 + 2816 + 192 + 13 = 43981_{10}$$



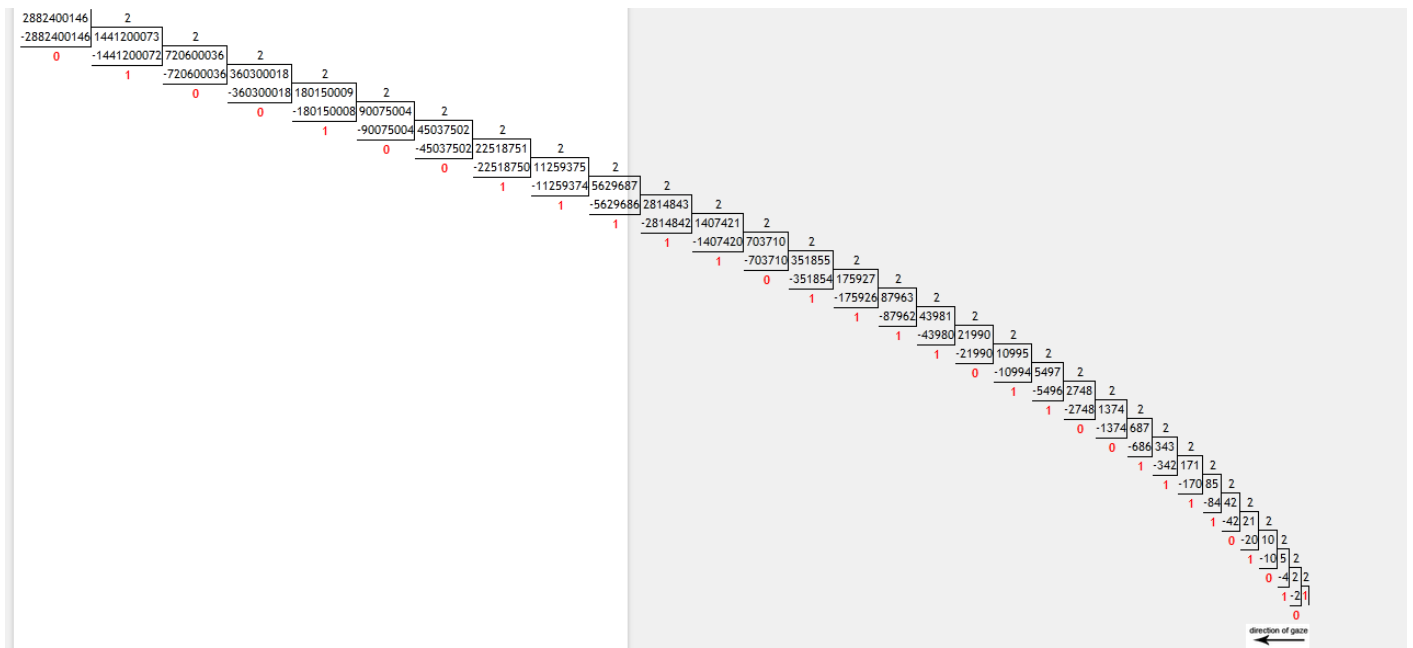
0

8.4534



ε. ABCDEF92

$$\begin{aligned} &10 \cdot 16^7 + 11 \cdot 16^6 + 12 \cdot 16^5 + 13 \cdot 16^4 + 14 \cdot 16^3 + 15 \cdot 16^2 + 9 \cdot 16^1 + 2 \cdot 16^0 = \\ &10 \cdot 268435456 + 11 \cdot 16777216 + 12 \cdot 1048576 + 13 \cdot 65536 + 14 \cdot 4096 + 15 \cdot 256 + 9 \cdot 16 + 2 \cdot 1 = \\ &2684354560 + 184549376 + 12582912 + 851968 + 57344 + 3840 + 144 + 2 = 2882400146_{10} \end{aligned}$$



47. Να δημιουργηθεί πρόγραμμα, (συμβολική γλώσσα Assembly) που να υπολογίζει αν ο αριθμός που θα αποθηκευτεί στον B είναι άρτιος ή περιττός. Σε περίπτωση που ο αριθμός είναι περιττός να τερματίζεται το πρόγραμμα, σε διαφορετική περίπτωση να ξεκινάει πάλι από την αρχή.

153. Τι τιμές έχουν τα bytes AEH και 83H αν αυτά παριστάνουν απροσημάστους αριθμούς του δυαδικού συστήματος και τι αν παριστάνουν προσημασμένους αριθμούς στην παράσταση συμπληρώματος ως προς 2 και γιατί;

Για απροσημάστους αριθμούς του δυαδικού συστήματος γράφω το κάθε ψηφίο του δεκαεξαδικού με την αντίστοιχη τετράδα στο δυαδικό και έχουμε:

$$(AE)_{16} = (1010 \ 1110)_2 = (10101110)_2 = (174)_{10}$$

$$(83)_{16} = (1000 \ 0011)_2 = (10000011)_2 = (131)_{10}$$

Για προσημασμένους αριθμούς στην παράσταση συμπληρώματος ως προς 2 :

$$(AE)_{16} = (1010 \ 1110)_{\text{συμπλήρωμα ως προς 2}} = (10101110)_{\text{συμπλήρωμα ως προς 2}}$$

$$(83)_{16} = (1000 \ 0011)_{\text{συμπλήρωμα ως προς 2}} = (10000011)_{\text{συμπλήρωμα ως προς 2}}$$

Είναι και οι δύο αρνητικοί αφού το πρώτο τους bit είναι 1 και το μέτρο τους προκύπτει εξ'ορισμού, εάν αντικαταστήσουμε το 0 με 1 και το 1 με 0 και στη συνέχεια προσθέσουμε 1.

$$(AE)_{16} = (10101110)_{\text{συμπλήρωμα ως προς 2}} = -(0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1)_{10} = -(82)_{10}$$

$$(83)_{16} = (10000011)_{\text{συμπλήρωμα ως προς 2}} = -(0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 1)_{10} = -(125)_{10}$$

46. Σας δίνονται οι παρακάτω εντολές σε assembly:

LDA str -> Το περιεχόμενο της θέσης μνήμης str θα μεταφερθεί στον καταχωρητή A.

STA str -> Το περιεχόμενο του καταχωρητή A μεταφέρεται στη θέση μνήμης str.

LDB str -> Το περιεχόμενο της θέσης μνήμης str αντιγράφεται στον καταχωρητή B.

STB str -> Το περιεχόμενο του καταχωρητή B αντιγράφεται στη θέση μνήμης str.

ADD -> Προσθέτει τα περιεχόμενα των καταχωρητών A και B και τα αποθηκεύει στον καταχωρητή A

Να γράψετε κώδικα χρησιμοποιώντας MONAXA τις παραπάνω εντολές και να υπολογίζει την παρακάτω έκφραση (ακολουθώντας τη σειρά των παρενθέσεων). $2x + y + 2(u+v)$

46. Σας δίνονται οι παρακάτω εντολές σε assembly:

LDA str -> Το περιεχόμενο της θέσης μνήμης str θα μεταφερθεί στον καταχωρητή A.

STA str -> Το περιεχόμενο του καταχωρητή A μεταφέρεται στη θέση μνήμης str.

LDB str -> Το περιεχόμενο της θέσης μνήμης str αντιγράφεται στον καταχωρητή B.

STB str -> Το περιεχόμενο του καταχωρητή B αντιγράφεται στη θέση μνήμης str.

ADD -> Προσθέτει τα περιεχόμενα των καταχωρητών A και B και τα αποθηκεύει στον καταχωρητή A.

Να γράψετε κώδικα χρησιμοποιώντας MONAXA τις παραπάνω εντολές και να υπολογίζει την παρακάτω έκφραση (ακολουθώντας τη σειρά των παρενθέσεων).

$2x + y + 2(u+v)$

LDA #x
LDB #x
ADD AB
LDB #y
ADD AB
STA str
LDA #u
LDB #v
ADD AB
ADD AB
LDB #u
ADD AB
LDB str
ADD AB

Απάντηση για το πρόγραμμα στην Assembly που υπολογίζει ζυγούς / μονούς αριθμούς.

```
;Author:quickgrid
;Site: http://quickgrid.blogspot.com
;Code: Even odd check using loop in 8086 assembly explained code
.model small
.stack 100h
.data
    msg1 db ': odd$'
    msg2 db ': even$'
.code
main proc
    mov ax, @data
    mov ds, ax
    ;when loop called it uses cx as counter
    mov cx, 9
    ;loop label
    loop:
        ;input a number
        mov ah, 1
        int 21h
        ;divide any number in al by bl or 2 here
        mov bl, 2
        div bl
        cmp ah, 1
        je even
        mov ah, 9
        lea dx, msg2
        int 21h
        jmp skip
    even:
        mov ah, 9
        lea dx, msg1
        int 21h
    skip:
        ;prints a newline
        mov ah, 2
        mov dl, 0dh
        int 21h
        mov dl, 0ah
        int 21h
    loop loop
main endp
end main
```