

Μικροϋπολογιστές

5η Ομάδα Ασκήσεων

Παπαρρηγόπουλος Θοδωρής el18040 paparrigopoulosthodoris@gmail.com

Φιλιππόπουλος Ορφέας el18082 orfeasfil2000@gmail.com

Εκτός από τα αρχεία των ασκήσεων επισυνάπτεται και ένα αρχείο MACROS.asm το οποίο περιέχει κάποιες χρήσιμες μακροεντολές που χρησιμοποιούνται στα περισσότερα προγράμματα.

Άσκηση 1η

Αποθηκεύουμε τους αριθμούς 128-1 σε έναν πίνακα χρησιμοποιώντας επανάληψη. Στη συνέχεια προσθέτουμε τους περιττούς αριθμούς ξεκινώντας από το 127 με βήμα 2. Αποθηκεύουμε το αποτέλεσμα και μετά το διαιρούμε με το 64 προκειμένου να βρούμε το μέσο όρο. Για το μέγιστο και το ελάχιστο αποθηκεύουμε στο MIN το 128 και στο MAX το 0 και κάθε φορά που βρίσκουμε μεγαλύτερο από το MAX ή μικρότερο από το MIN, αποθηκεύουμε τις νέες αυτές τιμές στα MAX και MIN αντίστοιχα.

Άσκηση 2

Αρχικά χρησιμοποιούμε τη ρουτίνα DEC_KEYB η οποία αγνοεί όλους τους χαρακτήρες πέρα από τους 0-9 επειδή το πρόγραμμα ζητάει από το χρήστη να διαβάσει δυο διψήφιους δεκαδικούς αριθμούς. Αφού το πρόγραμμα διαβάσει το ψηφίο υψηλότερης αξίας, το πολλαπλασιάζει με το 10 και στη συνέχεια προσθέτει το δεύτερο ψηφίο. Έτσι τυπώνουμε τους αριθμούς Z και W.

Στη συνέχεια χρησιμοποιούμε τη ρουτίνα PRINT_HEX. Αν ο αριθμός είναι μεγαλύτερος ή όχι από το 9, τότε προσθέτει το 37H ή το 30H απομονώνοντας τα 4 MSBs και στη συνέχεια τα 4 LSBs. Για την αφαίρεση ελέγχουμε αν ο W είναι μεγαλύτερος του Z. Αν ισχύει αυτό τότε πρέπει πρώτα να τυπώσουμε το αποτέλεσμα και μετά να αφαιρέσουμε τον Z από τον W.

Άσκηση 3

Οι ρουτίνες PRINT_DEC, PRINT_OCT, PRINT_BIN πραγματοποιούν συνεχόμενες διαιρέσεις με 10, 8, 2, αποθηκεύουν το υπόλοιπο κάθε φορά στη στοίβα ώστε να μπορούμε να απομονώνουμε τα ψηφία και να τα κάνουμε pop από τη στοίβα.

Αρχικά διαβάζουμε 3 HEX ψηφία χρησιμοποιώντας την ρουτίνα HEX_KEYB και κρατώντας το 1^ο ψηφίο στο BH. Το 2^ο ψηφίο το κάνουμε τέσσερις φορές SAL στο BL και προσθέτουμε το 3^ο ψηφίο στον BL. Ελέγχουμε συνέχεια αν διαβάζουμε T για να διακόψουμε τη λειτουργία. Τέλος, τυπώνουμε τα ψηφία σε δεκαεξαδική μορφή και καλούμε τις ρουτίνες που φτιάξαμε για να τυπώσουμε και τις άλλες μορφές.

Άσκηση 4

Το πρόγραμμα περιμένει το διάβασμα 20 χαρακτήρων ή το πάτημα του ENTER. Το πρόγραμμα υλοποιείται με loop που τρέχει για 20 επαναλήψεις αλλά σταματάει ενδιάμεσα αν πατηθεί το enter. Αν διαβάσει το χαρακτήρα "=" τερματίζει. Στη συνέχεια ελέγχει για γράμματα. Αν βρούμε γράμμα το

τυπώνουμε αφού πρώτα έχουμε αφαιρέσει το 20H για να γίνει κεφαλαίο. Στη συνέχεια τυπώνει μια παύλα και μετά ψάχνει για αριθμούς. Αν βρούμε αριθμό, τον τυπώνουμε. Ο έλεγχος γίνεται με τη σειρά που διαβάζουμε τους χαρακτήρες οπότε τυπώνονται και αυτοί με τη σωστή σειρά.

Άσκηση 5

Από το δεξί διάγραμμα, έχουμε ότι $y = a \cdot x$. Για $y = 4095$ και $x = 4$ έχουμε ότι $\text{volts} = 4/4095$. Από το αριστερό διάγραμμα για την πρώτη περιοχή, έχουμε $y = a \cdot x$ όπου $y = 2$ και $T=400$. Οπότε $T = 200 \cdot \text{volts} = 800/4095 = 100/511$. Αντίστοιχα, για την δεύτερη περιοχή έχουμε $y = a \cdot x + b$, $y = 3$ και $x = 1200$. Άρα $T = 800 \cdot \text{volts} - 1200 = 800/1024 - 1200$. Οπότε για την πρώτη περιοχή: $\text{volts} \leq 2 \Rightarrow \text{bits} \leq 2048$ και για την δεύτερη περιοχή: $2 < \text{volts} \leq 3 \Rightarrow 2048 < \text{bits} \leq 3071$ και $\text{error bits} > 3071$. Το πρόγραμμα ξεκινάει τυπώνοντας το μήνυμα START και περιμένει να διαβάσει το "Y" ώστε να ξεκινήσει. Διαβάζει τον τριψήφιο από το πληκτρολόγιο. Ελέγχει παράλληλα αν έχει διαβάσει N ώστε να τερματίσει. Ελέγχει σε ποια περιοχή βρίσκεται το input και αν είναι μεγαλύτερο του 3071 τυπώνει error. Αφού βρούμε την περιοχή που ανήκει το input, υπολογίζουμε το αποτέλεσμα με τα T-bits. Τέλος, για να έχουμε ακρίβεια δεκαδικού ψηφίου πολλαπλασιάζουμε τον αριθμό των bits με το 10 και το τελευταίο ψηφίο που τυπώνουμε είναι μετά το κόμμα.