



Προγραμματισμός Ι – 4^η Σειρά Ασκήσεων

/* Προθεσμία υποβολής μέσω του eclass: Δευτέρα 13/11/23, 23:59:59 */

/* Επίδειξη στο εργαστήριο (ανάλογα με το Group που είστε γραμμένοι) */

****ΠΡΟΣΟΧΗ ΣΤΟΥΣ ΠΕΡΙΟΡΙΣΜΟΥΣ ΚΑΘΕ ΑΣΚΗΣΗΣ****

1. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει έναν θετικό ακέραιο και να εμφανίζει το πλήθος των bits του αριθμού αυτού με τιμή 1. Για παράδειγμα, αν ο χρήστης εισάγει τον αριθμό 30 (δυαδικό: 000000000000000000000000011110), το πρόγραμμα να εμφανίζει 4.

2. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει δύο αριθμούς που αντιστοιχούν σε μίλια (π.χ. a και b) και έναν τρίτο αριθμό (π.χ. step). Το πρόγραμμα να εμφανίζει σε μία στήλη τα μίλια και σε μία δεύτερη στήλη τα αντίστοιχα χιλιόμετρα ξεκινώντας από τον πρώτο αριθμό (π.χ. a) μέχρι τον δεύτερο αριθμό (π.χ. b, θεωρώντας ότι $a < b$) με βήμα τον τρίτο αριθμό (π.χ. step). Η μεταξύ τους σχέση θεωρήστε ότι είναι: 1 μίλι = 1.6 χιλιόμετρα.

Για παράδειγμα, αν ο χρήστης εισάγει ως a το 0, ως b το 1000 και ως step το 200, τότε η έξοδος να είναι:

```
Enter miles interval: 0 1000
```

```
Enter step: 200
```

MILES	KLM
0.00	0.00
200.00	320.00
400.00	640.00
600.00	960.00
800.00	1280.00
1000.00	1600.00

3. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει συνεχώς ακέραιους και να υπολογίζει το άθροισμά τους μέχρι αυτό να ξεπεράσει το 100. Στη συνέχεια, το πρόγραμμα να εμφανίζει το άθροισμά τους (την πρώτη φορά που θα είναι μεγαλύτερο του 100) και πόσους αριθμούς εισήγαγε ο χρήστης έως τότε (πόσοι δηλ. χρειάστηκαν για να ξεπεράσει το άθροισμά τους το 100).

4. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει έως 100 ακεραίους. Αν ο αριθμός που εισάγεται είναι μεγαλύτερος από τον τελευταίο που εισήχθηκε, η εισαγωγή των ακεραίων να τερματίζει και το πρόγραμμα να εμφανίζει πόσοι αριθμοί εισήχθησαν συνολικά.

5. Να γραφεί ένα πρόγραμμα το οποίο να πρέπει υποχρεωτικά να διαβάσει 10 θετικούς αριθμούς. Αν ο χρήστης εισάγει αρνητικό αριθμό, το πρόγραμμα να τον υποχρεώνει να εισάγει έναν νέο. Το πρόγραμμα να εμφανίζει πόσοι αρνητικοί αριθμοί εισήχθησαν πριν τερματίσει. Να χρησιμοποιήσετε έναν **for** βρόχο. Το μηδέν δεν προσμετράται ούτε στους θετικούς ούτε στους αρνητικούς αριθμούς.

6. Να γραφεί ένα πρόγραμμα το οποίο χωρίς τη χρήση πίνακα να διαβάζει 10 ακεραίους και να εμφανίζει πόσες φορές ο χρήστης εισήγαγε διαδοχικές τιμές. Για παράδειγμα, αν ο χρήστης εισάγει: -5, 10, 17, -31, -30, -29, 9, 75, 76, -4 το πρόγραμμα να εμφανίζει 3, λόγω των ζευγαριών: {-31, -30}, {-30, -29} και {75, 76}.

7. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει έναν ακέραιο που να αντιστοιχεί σε αριθμό γραμμών. Στην πρώτη γραμμή να εμφανίζονται τόσα '*' όσα και ο αριθμός των γραμμών και σε κάθε επόμενη ένα λιγότερο. Για παράδειγμα, αν ο χρήστης εισάγει την τιμή 5 το πρόγραμμα να εμφανίζει:

```
*****
*****
***
**
*
```

8. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει έναν θετικό ακέραιο (π.χ. N) και να υπολογίζει το άθροισμα της παράστασης: $\frac{5}{3} + \frac{25}{9} + \frac{125}{27} + \dots$

μέχρι αυτό να γίνει μεγαλύτερο από την τιμή του ακεραίου N. Το πρόγραμμα να εμφανίζει το τελευταίο αποδεκτό άθροισμα (το μεγαλύτερο άθροισμα κλασμάτων που όμως είναι μικρότερο ακόμα από N) και το πλήθος των κλασματικών όρων του. Το πρόγραμμα να υποχρεώνει τον χρήστη να εισάγει θετικό αριθμό, κάνοντας χρήση της εντολής **do-while**.

9. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει έναν ακέραιο (π.χ. N) και να εμφανίζει το αποτέλεσμα της παράστασης: $\frac{1}{1 \times 3} + \frac{1}{3 \times 5} + \frac{1}{5 \times 7} + \dots + \frac{1}{(N-2) \times N}$

Το πρόγραμμα να υποχρεώνει τον χρήστη να εισάγει έναν περιττό αριθμό μεγαλύτερο ή ίσο του 3, κάνοντας χρήση της εντολής **do-while**.

10. Να γραφεί ένα πρόγραμμα το οποίο (χωρίς τη χρήση πίνακα) να διαβάζει 100 πραγματικούς αριθμούς και να βρίσκει και να εμφανίζει τις δύο μεγαλύτερες διαφορετικές τιμές.

11. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει έναν ακέραιο και να εμφανίζει τα ψηφία του με λέξεις (Σημ. Αν ο αριθμός είναι αρνητικός να προηγείται η λέξη minus). Το πρόγραμμα να δέχεται ακεραίους μέχρι πέντε ψηφία, κάνοντας χρήση της εντολής **do-while**. Για παράδειγμα, αν ο χρήστης εισάγει το -17, το πρόγραμμα να εμφανίζει: minus one seven.

12. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει έναν θετικό ακέραιο (να μην δέχεται αρνητικούς αριθμούς κάνοντας χρήση της εντολής **do-while**) και να ελέγχει αν είναι αριθμός Armstrong ή όχι. Ένας αριθμός είναι αριθμός Armstrong αν είναι ίσος με το άθροισμα των ψηφίων του υψωμένα στον συνολικό αριθμό των ψηφίων του. Για παράδειγμα, ο τριψήφιος αριθμός 153 είναι αριθμός Armstrong, γιατί $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$.

13. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει συνεχώς πραγματικούς αριθμούς τύπου **double** και να αποθηκεύει σε έναν πίνακα 100 θέσεων τους αριθμούς με τιμή μεγαλύτερη από 5. Αν ο χρήστης εισάγει την τιμή -1, η εισαγωγή των αριθμών να τερματίζει και το πρόγραμμα να εμφανίζει την ελάχιστη τιμή που αποθηκεύτηκε στον πίνακα.

14. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει 500 πραγματικούς αριθμούς τύπου **double** και να τους αποθηκεύει σε έναν πίνακα. Στη συνέχεια, να ελέγχει αν ο πίνακας είναι «συμμετρικός». Θεωρήστε ως «συμμετρία» του πίνακα την ιδιότητα το πρώτο του στοιχείο να είναι ίσο με το τελευταίο, το δεύτερο με το προτελευταίο, κ.ο.κ.. Θα υπήρχε πρόβλημα στη λύση που δώσατε αν ο πίνακας ήταν 501 ακεραίων; Δικαιολογήστε την απάντησή σας (ως σχόλιο μέσα στον κώδικά που θα παραδώσετε).

15. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει συνεχώς ακεραίους και να τους αποθηκεύει σε έναν πίνακα 100 θέσεων μέχρι αυτός να «γεμίσει», με τον περιορισμό ένας εισαγόμενος αριθμός να αποθηκεύεται στον πίνακα μόνο αν είναι μικρότερος από τον αριθμό που εισήχθητε τελευταίος. Στο τέλος, το πρόγραμμα να εμφανίζει όλα τα στοιχεία του πίνακα.