

Συμπληρωματικές Ασκήσεις 2

Επιμέλεια : Αλέξανδρος Αργυρόπουλος

Επικοινωνία: alex.argiropoulos2004@gmail.com

1. Να φτιάξετε πρόγραμμα που θα αποτελείται από τις εξής συναρτήσεις :

$$\alpha) f(x) = (\ln x)^2 \quad \beta) g(x) = x^3 + 4x^2 + 17 \quad \gamma) (f \circ g)(x) \quad \delta) (g \circ f)(x)$$

ε) `main()` : Η οποία θα ζητάει από τον χρήστη έναν αριθμό και θα χρησιμοποιεί κατάλληλα τις άλλες συναρτήσεις ώστε να εκτυπώνει τις τιμές αυτών , εφόσον ο αριθμός ανήκει στο πεδίο ορισμού .

2. Να φτιαχτεί πρόγραμμα που θα διαβάζει τις πλευρές δύο τριγώνων . Έπειτα θα εμφανίζει την Περίμετρο και το Εμβαδόν τους . Επίσης θα ελέγχει αν τα τρίγωνα είναι όμοια και τότε θα εμφανίζει το λόγο ομοιότητάς τους . Σχόλια: 1) Στο πρόγραμμα να περιέχεται μία συνάρτηση `main()` για τις εισόδους και τις εξόδους. 2) Να μην χρησιμοποιηθούν λίστες .

3. Χρησιμοποιώντας το `module random` να φτιάξετε ένα παιχνίδι τύχης με τους εξής κανόνες :

α) Μία παρτίδα αποτελείται από $N = 10$ γύρους .

β) Ο χρήστης επιλέγει έναν οποιοδήποτε μη μηδενικό ακέραιο x . Έπειτα ακολουθεί τυχαία επιλογή ακεραίου y από το $(-x, x)$ ή $(x, -x)$.

γ) Ανάλογα με την τιμή του y έχουμε τις εξής περιπτώσεις :

→ 0 : Θα πρέπει να δοθεί νέα τιμή του y .

→ Θετικός : Το σκορ του γύρου υπολογίζεται ως $S = \sum_{i=1}^y i$.

→ Αρνητικός : Το σκορ του γύρου υπολογίζεται ως $S = y \cdot (x + y)^3$.

δ) Το σκορ ενός γύρου , επηρεάζεται από το πλήθος των εμφανίσεων της τιμής 0 (πλ) για το y : Παρουσιάζεται ως συντελεστής $\frac{1}{\pi\lambda}$. Δηλαδή : $S_{\text{τελικό}} = \frac{1}{\pi\lambda} \cdot S$, εφόσον $\pi\lambda \neq 0$ αλλιώς δεν λαμβάνεται υπόψη.

ε) Το σκορ του κάθε γύρου αθροίζεται στο τελικό σκορ (ΤΣ) . Ο παίκτης "κερδίζει" αν $T\S > 0$, ενώ "χάνει" για $T\S < 0$. Στο μηδέν έχουμε "ισοπαλία" .

στ) Το κέρδος του παίκτη σε κάθε παρτίδα υπολογίζεται ως εξής : (Ορίζουμε ως pl_+, pl_- : το πλήθος εμφανίσεων θετικών και αρνητικών τιμών του y αντίστοιχα) .

→ $T\S > 0$: $K = T\S \cdot pl_+ - 2^{pl_-}$

→ $T\S < 0$: $K = 2^{pl_+} + T\S \cdot pl_-$

→ $T\S = 0$: $K = 0$

Το πρόγραμμα θα εκτυπώνει το ΤΣ , αν ο χρήστης νίκησε , έχασε ή βρέθηκε σε ισοπαλία , τα pl_+, pl_- και το κέρδος K (αν $K < 0$ μπορεί να αναφέρεται ως χρέος) . Σχόλιο: Το γεγονός ότι ένας παίκτης "κερδίζει" δεν εξασφαλίζει θετικό κέρδος . Αντίστοιχα όταν χάνει δεν πρόκειται αναγκαστικά χρέος .

4. **Πρόβλημα** (Δημιουργία module λογικών συναρτήσεων) : Να φτιάξετε ένα αρχείο `logic_functions` στο οποίο θα εισάγετε μία συνάρτηση για καθεμία από τις εξής περιπτώσεις :

- ▷ `x_in(a, b, x)` : Ελέγχει την ορθότητα της πρότασης $x \in (a, b)$.
- ▷ `is_prime(n)` : Ελέγχει αν ο φυσικός αριθμός n είναι πρώτος .
- ▷ `is_right_tr(a,b,c)` : Ελέγχει αν τρίγωνο με πλευρές a, b, c είναι ορθογώνιο .
- ▷ `is_perfect_sq(n)` : Ελέγχει αν ο αριθμός είναι το τετράγωνο ενός φυσικού αριθμού (χωρίς τη χρήση οποιασδήποτε προγραμμαστικής μορφής της τετραγωνικής ρίζας) .
- ▷ `is_eq_points(p1, p2)` : Ελέγχει αν τα δύο σημεία p_1, p_2 του επιπέδου (ως επέκταση του χώρου ή καλύτερα να αντιμετωπιστεί το επίπεδο ως ειδική περίπτωση του χώρου) ταυτίζονται .
- ▷ `is_p_symmetric(p1, p2)` , $p \in \{x, y, O, id\}$ (4 συναρτήσεις) : Ελέγχουν αν το σημείο p_2 είναι το συμμετρικό του p_1 ως προς τον άξονα $x'x, y'y$, την αρχή των αξόνων και την ευθεία $y = x$ αντίστοιχα .

5. **Εφαρμογές** του Προβλήματος 4 :

- i) *Διαστήματα* : Έστω δύο ανοικτά διαστήματα $I_1, I_2 \subseteq \mathbb{R}$. Να φτιαχτεί πρόγραμμα που ελέγχει :
α) $x \in I_1 \cap I_2$, β) $x \in I_1 \cup I_2$, γ) $x \in I_1 \triangle I_2$, δ) $I_1 \cap I_2 \neq \emptyset$ ε*) $x \in I_1^c$ (συμπλήρωμα) .
Υπόδειξη: Για το ε) να χρησιμοποιηθούν τα $\pm\infty$ από το module `math` ($\pm \text{math.inf}$) .
- ii) Να φτιαχτεί πρόγραμμα (δομημένο με συναρτήσεις , μια εξ'αυτών να είναι τύπου `main()`) στο οποίο θα γίνεται είσοδος τριών μηκών και θα γίνεται έλεγχος για το αν μπορούν ή όχι να αποτελέσουν ορθογώνιο τρίγωνο , ενημερώνοντας καταλλήλως το χρήστη. Έπειτα στην περίπτωση που είναι θα εκτυπώνει ποια πλευρά είναι η υποτείνουσα , θα υπολογίζει το εμβαδόν του τριγώνου και στην περίπτωση που η υποτείνουσα είναι τέλειο τετράγωνο θα εμφανίζει επίσης το ύψος που αντιστοιχεί στην υποτείνουσα .

Σχόλιο: Μία τρίτη άμεση εφαρμογή του module `logic_functions` περιέχεται στο Φυλλάδιο 3 (Άσκηση 7) .