

Ενδεικτικές λύσεις 3ης εξέτασης εργαστηρίων

Εργαστήριο Τρίτης 11:00-13:00

Ένας βάτραχος πέφτει σε πηγάδι βάθους D μέτρων και προσπαθεί να βγει έξω.

- Στο πρώτο του άλμα πηδάει H μέτρα.
- Καθένα από τα επόμενα άλματά του είναι N φορές το ύψος του προηγούμενου άλματος. Το N μπορεί να είναι και μικρότερο του 1, για παράδειγμα στην περίπτωση που ο βάτραχος κουράζεται μετά από κάθε άλμα.
- Δυστυχώς, μετά από κάθε του άλμα γλιστράει S μέτρα προς τα κάτω.

Ο βάτραχος βγαίνει από το πηγάδι όταν φτάσει ή ξεπεράσει το ύψος (βάθος) D . Αν, το πηγάδι έχει βάθος $D \leq H$ τότε ο βάτραχος βγαίνει έξω με το πρώτο του άλμα (θυμηθείτε ότι ο βάτραχος θα είχε βγει έξω προτού γλιστρίσει προς τα κάτω!).

Να γράψετε συνάρτηση που να υπολογίζει την έκβαση του παραπάνω "σεναρίου". Η συνάρτησή σας θα δέχεται ως είσοδο τα D , H , N και S , και θα επιστρέφει τον αριθμό των αλμάτων που χρειάστηκε ο βάτραχος για να απεμπλακεί από την περιπέτεια του. Ο βάτραχος μπορεί επίσης να μην καταφέρει να φτάσει στην κορυφή (ανάλογα με τα S και N) οπότε στην περίπτωση αυτή η συνάρτησή σας πρέπει να επιστρέφει -1.

Να γράψετε επίσης και μια συνάρτηση `main()` που θα καλεί την παραπάνω συνάρτηση για δεδομένα που θα δίνονται από τον χρήστη και στη συνέχεια να εμφανίζει το αποτέλεσμα.

```
def frog(D, H, N, S):
    progress = 0
    attempts = 0
    current_attempt = H
    while progress < D:
        attempts += 1
        progress += current_attempt
        if progress >= D:
            break
        if current_attempt <= S and N <= 1:
            return -1
        progress = max(progress-S, 0)
        current_attempt *= N
    return attempts
```

Εργαστήριο Πέμπτης 11:00-13:00

Η βάρκα ενός ψαρά, βυθίζεται S μέτρα από τη στεριά. Ο ψαράς προσπαθεί να βγει κολυμπώντας στην ακτή με μια σειρά προσπαθειών.

- Στην πρώτη προσπάθεια μετακινείται L μέτρα.
- Κάθε επόμενη προσπάθεια τον προωθεί N φορές την απόσταση της προηγούμενης προσπάθειας. Το N μπορεί να είναι και μικρότερο του 1, για παράδειγμα στην περίπτωση που ο ψαράς κουράζεται μετά από κάθε προσπάθεια.
- Δυστυχώς κολυμπάει ενάντια στο ρεύμα και μετά από κάθε προσπάθεια οπισθοχωρεί κατά C μέτρα λόγω ρεύματος.

Ο ψαράς βγαίνει στην ακτή όταν φτάσει ή ξεπεράσει την απόσταση S . Αν, η απόσταση από την ακτή $S \leq L$ τότε ο ψαράς βγαίνει έξω με την πρώτη του προσπάθεια (θυμηθείτε ότι ο ψαράς θα είχε βγει έξω προτού οπισθοχωρήσει λόγω του ρεύματος).

Να γράψετε συνάρτηση που να υπολογίζει την έκβαση του παραπάνω "σεναρίου". Η συνάρτησή σας θα δέχεται ως είσοδο τα S , L , N και C , και θα επιστρέφει τον αριθμό των προσπαθειών που χρειάστηκε ο ψαράς για να απεμπλακεί από την περιπέτεια του. Ο ψαράς μπορεί επίσης να μην καταφέρει να φτάσει στην ακτή (ανάλογα με τα C και N) οπότε στην περίπτωση αυτή η συνάρτησή σας πρέπει να επιστρέφει -1.

Να γράψετε επίσης και μια συνάρτηση `main()` που θα καλεί την παραπάνω συνάρτηση για δεδομένα που θα δίνονται από τον χρήστη και στη συνέχεια να εμφανίζει το αποτέλεσμα.

```
def fisherman(S, L, N, C):
    progress = 0
    attempts = 0
    current_attempt = L
    while progress < S:
        attempts += 1
        progress += current_attempt
        if progress >= S:
            break
        if current_attempt <= C and N <= 1:
            return -1
        progress -= C
        current_attempt *= N
    return attempts
```

Εργαστήριο Πέμπτης 13:00-15:00

Ένα μαϊμουδάκι προσπαθεί να αναρριχηθεί σε ένα δέντρο. Το ύψος του δέντρου είναι T μέτρα. Το μαϊμουδάκι ξεκινά από το κάτω μέρος του δέντρου και προσπαθεί να το αναρριχηθεί με μια σειρά προσπαθειών.

- Στην πρώτη προσπάθεια, το μαϊμουδάκι αναρριχάται C μέτρα.

- Σε κάθε επόμενη προσπάθεια, η απόσταση που αναρριχάται αυξάνεται κατά N (δηλαδή αναρριχάται N φορές την απόσταση της προηγούμενης προσπάθειας). Το N μπορεί να είναι και μικρότερο του 1, για παράδειγμα στην περίπτωση που το μαϊμουδάκι κουράζεται μετά από κάθε προσπάθεια.
- Ωστόσο, μετά από κάθε προσπάθεια, το μαϊμουδάκι ολισθαίνει κατά S μέτρα λόγω της λείας επιφάνειας του δέντρου.

Το μαϊμουδάκι συνεχίζει να αναρριχάται και να ολισθαίνει μέχρι να φτάσει ή να ξεπεράσει την κορυφή του δέντρου. Αν, το δέντρο έχει ύψος $T \leq C$ τότε το μαϊμουδάκι βγαίνει έξω με την πρώτη του προσπάθεια (θυμηθείτε ότι το μαϊμουδάκι θα είχε βγει έξω προτού γλιστρίσει προς τα κάτω!).

Να γράψετε συνάρτηση που να υπολογίζει την έκβαση του παραπάνω "σεναρίου". Η συνάρτησή σας θα δέχεται ως είσοδο τα T, C, N και S , και θα επιστρέφει τον αριθμό των προσπαθειών που χρειάστηκε το μαϊμουδάκι για να απεμπλακεί από την περιπέτεια του. Το μαϊμουδάκι μπορεί επίσης να μην καταφέρει να σκαρφλώσει στο δέντρο (ανάλογα με τα S και N) οπότε στην περίπτωση αυτή η συνάρτησή σας πρέπει να επιστρέφει -1 .

Να γράψετε επίσης και μια συνάρτηση `main()` που θα καλεί την παραπάνω συνάρτηση για δεδομένα που θα δίνονται από τον χρήστη και στη συνέχεια να εμφανίζει το αποτέλεσμα.

```
def monkey(T, C, N, S):
    progress = 0
    attempts = 0
    current_climb = C
    while progress < T:
        attempts += 1
        progress += current_climb
        if progress >= T:
            break
        if current_climb <= S and N <= 1:
            return -1
        progress = max(progress-S, 0)
        current_climb *= N
    return attempts
```