

## Εκφώνηση

Υλοποιήστε την εξαγωγή τύπων για τον απλό λ-λογισμό, όπως περιγράφεται στην [εκφώνηση](#) της άσκησης.

Μπορείτε να χρησιμοποιήσετε οποιαδήποτε από τις γλώσσες που υποστηρίζονται. Για δική σας διευκόλυνση, σας προτείνουμε να χρησιμοποιήσετε μια συναρτησιακή γλώσσα.

---

## Δεδομένα εισόδου

Η πρώτη γραμμή της εισόδου θα περιέχει το πλήθος  $N$  των λ-όρων που θα εισαχθούν. Κάθε μία από τις επόμενες  $N$  γραμμή θα περιέχει ένα λ-όρο.

Η ακριβής γραμματική για τους λ-όρους είναι αυτή που δίνεται στην εκφώνηση της άσκησης. Το γράμμα "λ" στην αφαίρεση θα συμβολίζεται με το χαρακτήρα "\" (backslash). Οι μεταβλητές θα αρχίζουν με γράμμα και θα αποτελούνται από γράμματα και ψηφία. Αν σας εξυπηρετεί, θεωρήστε ότι θα υπάρχει ακριβώς ένα κενό διάστημα μετά την τελεία στην αφαίρεση και μεταξύ των δύο όρων στην εφαρμογή. Παρενθέσεις θα χρησιμοποιούνται ακριβώς στα σημεία που ορίζει η γραμματική.

Είναι πιθανό να χρησιμοποιείται το ίδιο όνομα μεταβλητής σε πολλές αφαιρέσεις.

---

## Δεδομένα εξόδου

Η έξοδος του προγράμματός σας θα αποτελείται από  $N$  γραμμές. Κάθε μία θα περιέχει τον τύπο του αντίστοιχου λ-όρου, αν ο αλγόριθμος εξαγωγής τύπων επιτυγχάνει, διαφορετικά το μήνυμα "type error".

Οι τύποι πρέπει να εμφανίζονται σε κανονική μορφή, ως εξής. Ως μεταβλητές τύπων θα χρησιμοποιούνται οι συμβολοσειρές @0, @1, @2, κ.ο.κ. Το βέλος των συναρτήσεων θα γράφεται ως "->" με ένα κενό διάστημα εκατέρωθεν. Παρενθέσεις θα χρησιμοποιούνται μόνο όπου είναι απαραίτητες. Το υπόδειγμα κώδικα Haskell που δίνεται στο Moodle εμφανίζει τους τύπους ακριβώς με αυτήν τη μορφή.

Επιπλέον, η χρήση των μεταβλητών πρέπει να γίνεται με τέτοιο τρόπο ώστε ο τύπος που εμφανίζεται να είναι "λεξικογραφικά" μικρότερος από κάθε άλλον ισοδύναμο τύπο, που μπορεί να προκύψει με μετονομασία των μεταβλητών. Για παράδειγμα, ο τύπος:

```
β -> (β -> α) -> α
```

πρέπει να εμφανίζεται ως

```
@0 -> (@0 -> @1) -> @1
```

και όχι με κάποια από τις παρακάτω ισοδύναμες μορφές:

```
@1 -> (@1 -> @0) -> @0  
@7 -> (@7 -> @42) -> @42
```

**Σημείωση:** το "λεξικογραφικά" παραπάνω δε είναι απόλυτα σωστό, γιατί σύμφωνα με τη λεξικογραφική διάταξη των συμβολοσειρών ισχύει π.χ. "@2" > "@10". Αντίθετα, εδώ πρέπει να προτιμάται η μεταβλητή @2 έναντι της @10. Εν ολίγοις, οι μεταβλητές που θα εμφανίζονται σε έναν τύπο, κατά σειρά εμφάνισης, πρέπει να είναι οι @0, @1, @2, κ.λπ.

---

## Περιορισμοί

Οι περιορισμοί χρόνου και μνήμης δεν έχουν σκοπό να σας αναγκάσουν να βρείτε ιδιαίτερα αποδοτικές υλοποιήσεις για την εξαγωγή τύπων. Όμως, μην το παρακάνετε! Θα υπάρχουν testcases με πάνω από 100 όρους. Επίσης, θα υπάρχουν μεγάλοι όροι που θα έχουν μέχρι 100 εμφανίσεις μεταβλητών --- στον όρο "(λ. x)" μετράμε (αυθαίρετα) δύο εμφανίσεις μεταβλητών.

- Όριο χρόνου εκτέλεσης: 120 sec.
  - Όριο μνήμης: 128 MB.
- 

## Παράδειγμα εισόδου

```
5
(λx. x)
(λx. (λy. y))
(λx. (λy. x))
(λx. (x x))
(λx. (λy. (λz. ((y z) x))))
```

---

## Παράδειγμα εξόδου

```
@0 -> @0
@0 -> @1 -> @1
@0 -> @1 -> @0
type error
@0 -> (@1 -> @0 -> @2) -> @1 -> @2
```