

## Προγραμματιστική άσκηση:

### Η γλώσσα προγραμματισμού greek++

Η greek++ είναι μια μικρή γλώσσα προγραμματισμού. Παρόλο που οι προγραμματιστικές της ικανότητες είναι μικρές, η εκπαιδευτική αυτή γλώσσα περιέχει πλούσια στοιχεία και η κατασκευή του μεταγλωττιστή της έχει να παρουσιάσει αρκετό ενδιαφέρον, αφού περιέχονται σε αυτήν όλες σχεδόν οι εντολές που χρησιμοποιούνται από άλλες γλώσσες. Η greek++ περιέχει συναρτήσεις και διαδικασίες, μετάδοση παραμέτρων με αναφορά και τιμή, δομές απόφασης και βρόχων, αναδρομικές κλήσεις, κλπ.

Από την άλλη όμως πλευρά, η greek++ δεν υποστηρίζει βασικά προγραμματιστικά εργαλεία όπως πίνακες και συμβολοσειρές. Υποστηρίζει μόνο ακέραιους αριθμούς. Οι παραλήψεις αυτές έχουν γίνει ώστε να απλουστευτεί η διαδικασία κατασκευής του μεταγλωττιστή, μία απλούστευση όμως που έχει να κάνει κυρίως με τη μείωση των γραμμών του κώδικα και όχι με τη δυσκολία κατασκευής του.

#### Λεκτικές μονάδες

Το αλφάβητο της greek++ αποτελείται από τα μικρά και κεφαλαία γράμματα της ελληνικής και της αγγλικής αλφαβήτου («Α»,...,«Ζ», «α»,...,«ζ», «Α»,...,«Ω» και «α»,...,«ω»), τα αριθμητικά ψηφία («0»,...,«9»), τα σύμβολα των αριθμητικών πράξεων («+», «-», «\*», «/»), τους τελεστές συσχέτισης («<», «>», «=», «<=», «>=», «<>», το σύμβολο ανάθεσης «:=», τους διαχωριστές («;», «,»,«:») καθώς και τα σύμβολα ομαδοποίησης («(»,«)»,«[»,«]»,«{»,«}»), διαχωρισμού σχολίων («{»,«}») και του περάσματος παραμέτρων με αναφορά («%»).

Μερικές λέξεις είναι δεσμευμένες:

πρόγραμμα	δήλωση		
εάν	τότε	αλλιώς	εάν_τέλος
επανάλαβε	μέχρι	όσο	όσο_τέλος
για	έως	με_βήμα	για_τέλος
διάβασε	γράψε		
συνάρτηση	διαδικασία	διαπροσωπεία	
είσοδος	έξοδος		
αρχή_συνάρτησης		τέλος_συνάρτησης	
αρχή_διαδικασίας		τέλος_διαδικασίας	
αρχή_προγράμματος		τέλος_προγράμματος	
ή	και	εκτέλεσε	

Οι λέξεις αυτές δεν μπορούν να χρησιμοποιηθούν ως μεταβλητές.

Οι σταθερές της γλώσσας είναι είτε ακέραιες σταθερές που αποτελούνται από προαιρετικό πρόσημο.

Τα αναγνωριστικά της γλώσσας είναι συμβολοσειρές που αποτελούνται από γράμματα και ψηφία, αρχίζοντας όμως από γράμμα. Το μέγεθος των μεταβλητών δεν πρέπει να ξεπερνά τα τριάντα γράμματα ή αριθμούς. Οι λευκοί χαρακτήρες (tab, space, return) αγνοούνται και μπορούν να χρησιμοποιηθούν με οποιονδήποτε τρόπο χωρίς να επηρεάζεται η λειτουργία του μεταγλωττιστή, αρκεί βέβαια να μη βρίσκονται μέσα σε δεσμευμένες λέξεις, αναγνωριστικά, σταθερές, ή άλλες δεσμευμένες λέξεις. Το ίδιο ισχύει και για τα σχόλια, τα οποία πρέπει να βρίσκονται ανάμεσα σε αγκύλες.

## **Τύποι και δηλώσεις μεταβλητών**

Οι τύποι δεδομένων που υποστηρίζει η greek++ είναι οι ακέραιοι και οι πραγματικοί αριθμοί. Η δήλωση γίνεται με την εντολή δηλώσεις. Ακολουθούν τα ονόματα των αναγνωριστικών χωρισμένα με κόμματα, άνω και κάτω τελεία και στο τέλος ο τύπος της μεταβλητής. Μπορούμε να έχουμε πάνω από μία γραμμές δηλώσεων. Η παρακάτω δήλωση είναι νόμιμη:

```
δηλώσεις  
    i, j  
    k
```

## **Τελεστές και εκφράσεις**

Η προτεραιότητα των τελεστών από τη μεγαλύτερη στη μικρότερη είναι:

- (1) Μοναδικοί «όχι»
- (2) Πολλαπλασιαστικοί «\*», «/»
- (3) Μοναδικοί προσθετικοί «+», «-»
- (4) Δυαδικοί προσθετικοί «+», «-»
- (5) Σχεσιακοί «=», «<», «>», «<>», «<=», «>=»
- (6) Λογικοί «και», «ή»

## **Εντολές**

- Εκχώρηση

```
Id := expression
```

Χρησιμοποιείται για την ανάθεση της τιμής μίας μεταβλητής ή μίας σταθεράς, ή μίας έκφρασης σε μία μεταβλητή.

- Απόφασης

```
εάν condition τότε
    ...
[ αλλιώς
    ... ]
εάν_τέλος
```

Η εντολή απόφασης εκτιμάει εάν ισχύει η συνθήκη `condition` και εάν πράγματι ισχύει εκτελούνται οι εντολές που ακολουθούν. Το `αλλιώς` δεν αποτελεί υποχρεωτικό τμήμα της εντολής και γι' αυτό βρίσκεται σε αγκύλη. Οι εντολές που το ακολουθούν εκτελούνται εάν η συνθήκη `condition` δεν ισχύει.

- Επανάληψης

```
επανάλαβε
    ...
μέχρι condition
```

Η εντολή `επανάλαβε` εκτελεί τις εντολές που βρίσκονται ανάμεσα στο `επανάλαβε` και το `μέχρι` και τις επαναλαμβάνει έως ότου η συνθήκη `condition` σταματήσει να ισχύει.

```
όσο condition επανάλαβε
    ...
όσο_τέλος
```

Η εντολή `όσο` αποτιμά τη συνθήκη `condition` και εκτελεί τις εντολές που βρίσκονται ανάμεσα στο `όσο` και το `όσο_τέλος` και τις επαναλαμβάνει έως ότου η συνθήκη `condition` σταματήσει να ισχύει.

```
για id:=expr1 έως expr2 [ με_βήμα expr3 ] επανάλαβε
    ...
για_τέλος
```

Η `για` εκτελεί ότι υπάρχει ανάμεσα στο `για` και το `για_τέλος`, όσες φορές χρειαστεί, έτσι ώστε το `id` να ξεκινήσει από την τιμή `expr1`, να φτάσει στην τιμή `expr2` ενώ σε κάθε επανάληψη να μεταβάλλεται κατά `expr3`. Το `expr3` μπορεί να είναι είτε θετικό, είτε αρνητικό ή μηδέν. Στην πρώτη περίπτωση η `για` τερματίζεται αν το `id` γίνει μεγαλύτερο ίσο του `expr2`, ενώ στη δεύτερη αν γίνει μικρότερο ή ίσο του `expr2`. Αν το `expr3` είναι μηδέν, τότε έχουμε ατέρμονο βρόχο. Σημειώστε ότι αν το `id` μεταβληθεί μέσα στον βρόχο από τον προγραμματιστή, ο μεταγλωττιστής πρέπει να δημιουργηθεί μήνυμα σφάλματος.

- Εισόδου

διάβασε `id`

Διαβάζει μία μεταβλητή από το πληκτρολόγιο

- Εξόδου

γράψε `expression`

Παίρνει σαν όρισμα ένα `expression` το αποτιμά και το εμφανίζει στην οθόνη.

## Υποπρογράμματα

Η `greek++` υποστηρίζει συναρτήσεις και διαδικασίες

```
συνάρτηση id (formal_parameters)  
    formal_parameters  
    declarations  
    αρχή  
    sequence of statements  
    τέλος_συνάρτησης
```

```
διαδικασία id (formal_parameters)  
    formal_parameters  
    declarations  
    αρχή  
    sequence of statements  
    τέλος_διαδικασίας
```

Η «`formal_parameters`» είναι η λίστα των τυπικών παραμέτρων η οποία δηλώνεται μέσα στην παρένθεση, αλλά οι τύποι της καθώς και το εάν είναι με τιμή ή αναφορά δηλώνονται από κάτω.

## Μετάδοση παραμέτρων

Η `greek++` υποστηρίζει δύο τρόπους μετάδοσης παραμέτρων:

- (1) με σταθερή τιμή. Δηλώνεται με τη λεκτική μονάδα `in`. Αλλαγές στην τιμή της δεν επιστρέφονται στον καλόν πρόγραμμα
- (2) με αναφορά. Δηλώνεται με τη λεκτική μονάδα `inout`. Κάθε αλλαγή στη τιμή της μεταφέρεται και στο πρόγραμμα που κάλεσε τη συνάρτηση.

## Μορφή προγράμματος

```
πρόγραμμα id
  declarations
  subprograms
  αρχή_προγράμματος
    sequence of statements
  τέλος_προγράμματος
```

Η παραπάνω περιγραφή της γλώσσας συμπληρώνεται με τη γραμματική της στην οποία διασαφηνίζεται η ακριβής σύνταξη των εντολών και της μορφής του προγράμματος.

Ένα παράδειγμα προγράμματος είναι το ακόλουθο:

```
πρόγραμμα τεστ
  δήλωση α, β
  δήλωση γ

  συνάρτηση αύξηση(α, β)
    διαπροσωπεία
    είσοδος α
    έξοδος β
    αρχή_συνάρτησης
      β := α + 1 ;
      αύξηση := α + 1    { δεν μπαίνει ερωτηματικό
                          είναι τέλος block }
    τέλος_συνάρτησης

  διαδικασία τύπωσε_συν_1(χ)
    διαπροσωπεία
    είσοδος χ
    αρχή_διαδικασίας
      γράψε χ+1
    τέλος_διαδικασίας

  αρχή_προγράμματος
    α := 1 ;
    β := 2 + α * α / (2 - α - (2*α));
    γ := αύξηση(α, %β);
```

```
για α:=1 έως 8 με_βήμα 2 επανάλαβε
    εκτέλεσε τύπωσε_συν_1(α)
για_τέλος;
β := 1;
όσο β<10 επανάλαβε
    εάν β<>22 ή [β>=23 και β<=24] τότε
        β := β+1
    εάν_τέλος { όχι ερωτηματικό, είναι τέλος block }
όσο_τέλος; { θέλει ερωτηματικό
    χωρίζει εκτελέσιμες εντολές }
διάβασε β;
επανάλαβε
    β := β + 1
μέχρι β<-100
τέλος_προγράμματος
```