

ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2018-19)

Εργασία 2

Γράψτε ένα πρόγραμμα C (έστω ότι το πηγαίο αρχείο του ονομάζεται `calculate.c`) το οποίο να διαβάζει από την είσοδο αριθμητικές παραστάσεις μεταξύ ακεραίων, μία σε κάθε γραμμή, μέχρι το τέλος της εισόδου, και να εκτυπώνει, για κάθε παράσταση, το αριθμητικό αποτέλεσμά της.

Στις παραστάσεις επιτρέπεται να υπάρχει “λευκό διάστημα” (white space), δηλαδή κενοί χαρακτήρες ή χαρακτήρες στηλογνώμονα (πλήκτρο `tab`), όχι όμως στο εσωτερικό των αριθμών (τα ψηφία τους πρέπει να είναι συνεχόμενα). Δεν υπάρχει απαίτηση να αναγνωρίζετε και να υπολογίζετε παραστάσεις που περιλαμβάνουν αριθμούς, ή που για τον υπολογισμό τους απαιτούνται ενδιάμεσα αποτελέσματα, που δεν μπορούν να φυλαχθούν σε τύπο ακεραίων `int`. Αν υπάρχει κάποιο συντακτικό λάθος στην παράσταση, θα πρέπει το πρόγραμμά σας να το αντιλαμβάνεται, να εκτυπώνει μήνυμα λάθους, και να συνεχίζει με την επόμενη παράσταση.

Προτείνεται να αναπτύξετε το πρόγραμμά σας σταδιακά, με διαρκώς αυξανόμενες απαιτήσεις, ώστε να είναι ευκολότερη η αποσφαλμάτωσή του. Για το σκοπό αυτό, ακολουθήστε τα βήματα που παρουσιάζονται στη συνέχεια (τα ποσοστά στις παρενθέσεις είναι η βαρύτητα του κάθε βήματος στην τελική αξιολόγηση του προγράμματος που θα παραδώσετε).

Βήμα Α (20%)

Θα πρέπει να αναγνωρίζονται παραστάσεις που περιλαμβάνουν προσθέσεις και αφαιρέσεις μεταξύ αριθμών (χωρίς πρόσημο). Οι πράξεις εκτελούνται από τα αριστερά προς τα δεξιά. Φυσικά, το αποτέλεσμα μπορεί να είναι και αρνητικός αριθμός. Ακόμα και ένας απλός ακεραίος αριθμός θεωρείται ως παράσταση, που έχει σαν τιμή τον ίδιο τον αριθμό. Μία ενδεικτική εκτέλεση φαίνεται στη συνέχεια.

```
$ ./calculate
32+541
Result 1: 573
125 - 38 - 142 + 5
Result 2: -50
125 * 32
Result 3: Error!
37-0
Result 4: 37
125a + 3
Result 5: Error!
+ -
Result 6: Error!
678
Result 7:
678
^D
```

Βήμα Β (30%)

Οι όροι που αναγνωρίζονται στην αριθμητική παράσταση με τις προσθαφαιρέσεις να μπορούν να είναι, εκτός από αριθμοί, και γινόμενα ακεραίων. Υπενθυμίζεται ότι, στα μαθηματικά, σε παραστάσεις που περιλαμβάνουν προσθέσεις, αφαιρέσεις και πολλαπλασιασμούς, οι πολλαπλασιασμοί προηγούνται των προσθέσεων και των αφαιρέσεων, ενώ οι προσθέσεις και αφαιρέσεις εκτελούνται κατά σειρά, από τα αριστερά προς τα δεξιά. Κάποια ενδεικτική εκτέλεση είναι η εξής:

```
$ ./calculate
32 - 5*7
Result 1: -3
3*6*2*4 + 2*2*10 -7*8
Result 2: 128
^D
```

Βήμα Γ (10%)

Να είναι δυνατόν να συμμετέχουν στις παραστάσεις και αριθμοί με πρόσημο (+ ή -). Ενδεικτική εκτέλεση:

```
$ ./calculate
32 + -8
Result 1: 24
-25 + +7 - -9
Result 2: -9
-25 * -4 + 6 * -3 * 5
Result 3: 10
^D
```

Βήμα Δ (30%)

Να είναι δυνατόν να συμμετέχουν στην αριθμητική παράσταση και επί μέρους παραστάσεις μέσα σε παρενθέσεις (σε οποιοδήποτε βάθος). Ενδεικτική εκτέλεση:

```
$ ./calculate
5 * (3 + 7)
Result 1: 50
((-3 + 8) * ((7 + 2) * (5 - -6 - 8) + 3 * -5)) + (7 * (2 - 8))
Result 2: 18
^D
```

Βήμα Ε (10%)

Σε περίπτωση λάθους εισόδου, να εκτυπώνεται ακριβέστερο διαγνωστικό μήνυμα για το είδος του λάθους. Για τις περιπτώσεις λαθών που θα αναγνωρίζει το πρόγραμμά σας, έχετε σχετική ελευθερία. Μπορεί να είναι περισσότερο (αλλά όχι λιγότερο) ακριβή από αυτά που φαίνονται στην ενδεικτική εκτέλεση που ακολουθεί.

```
$ ./calculate
35 27 - 4
Result 1: Unexpected character
32 * * 7
Result 2: Unexpected character
23 + 7 -
Result 3: Unexpected end of input
5 * (3 + 4
Result 4: Missing closing parenthesis
^D
```

Ουσιαστικά, αυτό που ζητείται στην άσκηση αυτή είναι η υλοποίηση ενός μικρού τμήματος του προγράμματος `bc`, που συνήθως είναι εγκατεστημένο στα συστήματα Linux.

Για λεπτομέρειες επάνω στη σύνταξη των επιτρεπτών αριθμητικών παραστάσεων που δεν καθορίζονται σαφώς στην εκφώνηση, μπορείτε είτε να δείτε πώς χειρίζεται τη συγκεκριμένη περίπτωση που σας προβληματίζει το πρόγραμμα `bc` και να υιοθετήσετε την προσέγγισή του, είτε να πάρετε κάποια απόφαση και να την ακολουθήσετε, τεκμηριώνοντάς την με σαφήνεια στα σχόλια του προγράμματός σας.

Για διευκόλυνσή σας, παρατίθενται στη συνέχεια οι κανόνες (σε μορφή BNF - Backus-Naur Form) που καθορίζουν πότε μία αριθμητική παράσταση (`<expression>`) είναι συντακτικά αποδεκτή. Το σύμβολο `|` υπονοεί διάζευξη. Υπενθυμίζεται ότι μέσα στις αριθμητικές παραστάσεις επιτρέπεται το λευκό διάστημα (κάτι που δεν περιγράφεται στους παρακάτω κανόνες), όχι όμως μεταξύ των ψηφίων ενός αριθμού.

```
<expression> ::= <term> | <expression> '+' <term> | <expression> '-' <term>
<term>       ::= <factor> | <factor> '*' <term>
<factor>     ::= <snumber> | '(' <expression> ')'
<snumber>    ::= <unumber> | '+' <unumber> | '-' <unumber>
<unumber>    ::= <digit> | <digit> <unumber>
<digit>      ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
```

Συνοπτικά, η ερμηνεία των παραπάνω κανόνων είναι η εξής. Μία αριθμητική παράσταση είναι είτε ένας όρος, είτε μία ακολουθία με προσθαιρέσεις δύο ή περισσότερων όρων. Ένας όρος είναι είτε ένας παράγοντας είτε ένα γινόμενο δύο ή περισσότερων παραγόντων. Ένας παράγοντας είναι είτε ένας αριθμός, είτε μία αριθμητική παράσταση κλεισμένη μέσα σε παρενθέσεις. Ένας αριθμός είναι είτε χωρίς πρόσημο, είτε με πρόσημο `+` ή `-`. Ένας αριθμός χωρίς πρόσημο είναι μία ακολουθία από ένα ή περισσότερα ψηφία (0, 1, ..., 9). Ουσιαστικά, αυτό που ορίζουν οι προηγούμενοι κανόνες είναι μία *γραμματική* (grammar) για τις συντακτικά αποδεκτές αριθμητικές εκφράσεις.

Η παράδοση της άσκησης αυτής συνίσταται στην υποβολή του πηγαίου αρχείου `calculate.c` μέσω του `eclass`.

Σημείωση: Στην άσκηση αυτή **απαγορεύεται αυστηρά η χρήση πινάκων (συμπεριλαμβανομένων και των συμβολοσειρών)**. Επίσης, απαγορεύεται η χρήση συναρτήσεων της βιβλιοθήκης εισόδου-εξόδου της C που διαχειρίζονται αρχεία.