

## 11.11 Άσκηση 11 – Fractions (Πράξεις με κλάσματα<sup>1</sup>)

### 11.11.1 Η έκφραση

Θεωρήστε εκφράσεις της παρακάτω μορφής

<τελεστής> <αριθμητής 1<sup>ου</sup> κλάσματος>/<παρονομαστής 1<sup>ου</sup> κλάσματος> <αριθμητής 2<sup>ου</sup> κλάσματος>/<παρονομαστής 2<sup>ου</sup> κλάσματος>

Όπου ο τελεστής μπορεί να είναι + - \* /

Παραδείγματα εκφράσεων

+ 1/5 3/4

\* 3/6 4/12

### 11.11.2 Το πρόγραμμα

Αναπτύξτε ένα πρόγραμμα που θα δίνει τη δυνατότητα στο χρήστη να εισάγει εκφράσεις της παραπάνω μορφής και το σύστημα να υπολογίζει την τιμή της έκφρασης και να την εμφανίζει στην κύρια έξοδο. Η τιμή της έκφρασης θα υπολογίζεται με εφαρμογή του τελεστή (operator) στους δύο τελεστέους (operands) που έπονται του τελεστή. Το αποτέλεσμα να δίνεται και σε μορφή απλοποιημένου κλάσματος.

Υιοθετήστε την αυξητική ανάπτυξη. Αξιοποιήστε την αφαιρετικότητα στις διεργασίες.

### 11.11.3 Η συνάρτηση getExpression

Για να διαβάσετε την έκφραση από την κύρια είσοδο θα χρησιμοποιήσετε τη συνάρτηση `getExpression()` η οποία βρίσκεται στην βιβλιοθήκη `libi2p.dll`.

Η συνάρτηση δίνεται σε δύο εκδόσεις: `getExpressionV1()` και `getExpressionV2()`.

Η συνάρτηση και στις 2 εκδόσεις της :

- ζητάει από τον χρήστη να εισάγει από την βασική είσοδο μια έκφραση όπως αυτή ορίζεται παραπάνω, και,
- διαβάζει από την βασική είσοδο τα στοιχεία της έκφρασης που εισάγει ο χρήστης.

Στα στοιχεία αυτά θα πρέπει με κάποιο τρόπο να αποκτήσει πρόσβαση η συνάρτηση που κάλεσε την `getExpression` (δηλαδή η `main` για την άσκηση μας). Στο σημείο αυτό είναι η διαφορά των 2 εκδόσεων της `getExpression`. Η μία αξιοποιεί τους δείκτες (pointers) και η άλλη τις δομές (struct).

<sup>1</sup> Η άσκηση βασίζεται σε ανάλογη άσκηση που δίνεται στους φοιτητές της σχολής Ηλεκτρολόγων του ΕΜΠ στο μάθημα Εισαγωγή στον Προγραμματισμό.

### 11.11.4 Η `getExpression` με δείκτες (pointers)

Η `getExpressionV1()` διαβάζει από την `stdin` τα στοιχεία που αποτελούν μια έκφραση προθεματικού τελεστή (prefix notation - παράγραφος 4.2.1) με κλάσματα, δηλαδή τον τελεστή και τους αριθμητές και παρονομαστές των κλασμάτων. Παράδειγμα έκφρασης: `+ 2/13 5/13`

Η **συνάρτηση δέχεται** ως ορίσματα δείκτες σε μεταβλητές στις οποίες θα αποδώσει τις τιμές που θα διαβάσει από την κύρια είσοδο. **Επιστρέφει** δε:

- `false` αν στην θέση του τελεστή δοθεί ο χαρακτήρας `q` ή `Q`,
- `true` σε κάθε άλλη περίπτωση

Η συνάρτηση `getExpressionV1` έχει το παρακάτω function prototype

```
bool getExpressionV1(char *operatorPtr, int *op1np,
                   int *op1dp, int *op2np, int *op2dp);
```

όπου

`operatorPtr` : operator Pointer (Δείκτης στην μεταβλητή `operator`)

`op1nPtr` : operand1 numerator Pointer (Δείκτης στην μεταβλητή `operand1 numerator`)

`op1dPtr` : operand1 denominator Pointer (Δείκτης στην μεταβλητή `operand1 denominator`)

`op2nPtr` : operand2 numerator Pointer

`op2dPtr` : operand2 denominator Pointer

**Για να κατανοήσετε τη χρήση δεικτών ως ορίσματα συνάρτησης** ανατρέξτε στις διαφάνειες 17-21 του αρχείου [I2P\\_Functions.pdf](#).

Παράδειγμα χρήσης της `getExpressionV1()` μπορείτε να βρείτε στο αρχείο [i2pLibrary.pdf](#).

### 11.11.5 Η `getExpression` με δομές (struct)

Η 2 έκδοση της `readExperssion` αποθηκεύει τα στοιχεία της έκφρασης που διαβάζει από τον χρήστη σε μία δική της μεταβλητή τύπου `Expression` και στη συνέχεια επιστρέφει αυτή την μεταβλητή στην συνάρτηση που την κάλεσε. Η συνάρτηση έχει το παρακάτω function prototype:

```
Expression getExpressionV2(void);
```

Αυτό σημαίνει πως η συνάρτηση που θα την καλέσει να πρέπει να έχει ορίσει μια μεταβλητή τύπου `Expression` και να καλέσει την `getExpressionV2` για να της αποδώσει τιμή. Οι παρακάτω προτάσεις κάνουν αυτό ακριβώς. Η πρώτη δηλώνει τη μεταβλητή `exp` ως τύπου `Expression` και η 2η καλεί την `getExpression` για να της αποδώσει τιμή.

```
Expression exp;
```

```
exp = getExpressionV2();
```

Με τον τρόπο αυτό η συνάρτηση που κάλεσε την `getExpressionV2` έχει πρόσβαση στα στοιχεία της έκφρασης καθώς αυτά βρίσκονται στην δική της μεταβλητή (`exp` - local variable).

Αν στην θέση του τελεστή δοθεί ο χαρακτήρας q η Q, η συνάρτηση επιστρέφει έχοντας τοποθετήσει τον χαρακτήρα στην θέση του τελεστή (exp.optr). Αυτό μπορείτε να το ελέγξετε για να τερματίσετε το πρόγραμμα σας.

Παράδειγμα χρήσης της `getExpressionV2()` μπορείτε να βρείτε στο αρχείο [i2pLibrary.pdf](#).

### **11.11.6 Αξιοποίηση της βιβλιοθήκης `libi2p.dll`**

Οδηγίες για την αξιοποίηση της βιβλιοθήκης `libi2p.dll` θα βρείτε στο αρχείο [i2pLibrary.pdf](#)