(Flex & Bison – Αριθμομηχανή)

Σκοπός:
Να υλοποιήσετε έναν απλό αριθμομηχανή χρησιμοποιώντας τα εργαλεία **Flex** (lexer) και **Bison** (parser).
Το πρόγραμμα πρέπει να μπορεί να αναλύει και να υπολογίζει απλές αριθμητικές εκφράσεις που περιλαμβάνουν:

- ακέραιους αριθμούς,
- τον τελεστή πρόσθεσης +,
- τον τελεστή πολλαπλασιασμού *,
- παρενθέσεις ( ).

Το πρόγραμμα πρέπει να είναι σε θέση να διαβάζει είσοδο:

- **από αρχείο** (μέσω stdin redirection ή ορίσματος γραμμής εντολών).

Απαιτήσεις

1. Flex (calc.l)

Υλοποιήστε lexer που:

1. Αναγνωρίζει ακέραιους αριθμούς (π.χ. 123) και επιστρέφει το token NUMBER.
2. Διαβάζει και αγνοεί whitespace (space, tab, newline).
3. Επιστρέφει τα tokens +, *, (και) όταν τα συναντήσει.
4. Αγνοεί οποιονδήποτε άλλο άσχετο χαρακτήρα.

Ο lexer πρέπει να τοποθετεί την αριθμητική τιμή των αριθμών στο yylval.

---

2. Bison (calc.y)

Υλοποιήστε parser που:

1. Ορίζει τη γραμματική (BNF) για αριθμητικές εκφράσεις:
    - expr + expr
    - expr * expr
    - (expr)
    - NUMBER

2. Υπολογίζει την τιμή κάθε έκφρασης χρησιμοποιώντας semantic actions.
3. Τυπώνει το αποτέλεσμα κάθε έκφρασης με τη μορφή:
4. = <αποτέλεσμα>

---

3. Υποστήριξη εισόδου από αρχείο

Αν ο χρήστης δώσει όνομα αρχείου στη γραμμή εντολών, π.χ.:

./calc input.txt

τότε το πρόγραμμά σας πρέπει να διαβάζει από το αρχείο αυτό.

Παράδειγμα εισόδου

3+4*2

7+1*5

(2+3)*4

10+2*3

Παράδειγμα εξόδου

= 11

= 12

= 20

= 16

Παραδοτέα

1. Τα αρχεία **calc.l** και **calc.y**.
2. Τον κώδικα μεταγλώττισης:
3. bison -d calc.y
4. flex calc.l
5. gcc calc.tab.c lex.yy.c -o calc
6. Ένα μικρό αρχείο δοκιμών (input.txt).
7. Ένα screenshot από επιτυχή εκτέλεση.