

# Tema 3

Responsabili temă:	Alexandru Placinta, Mihaela Catrina, Catalin Olaru, Bianca Cauc, Ovidiu Dancila
Data publicării:	<b>25.04.2018</b>
Termenul de predare:	<b>15.05.2018 ora 23:55</b> Se acceptă teme trimise cu penalizare de 10 puncte / zi (din maxim 100 puncte) până la data de 18.05.2018 ora 23:55

Sătul de întrebările de matematică ale fratelui său mai mic, Gigi s-a hotărât să își ușureze munca și să scrie un program care să evalueze automat expresii matematice. Întrucât fratele lui Gigi este în clasa a 7-a, setul de operații pe care programul lui Gigi trebuie să le implementeze este unul restrâns. Cu toate acestea, deoarece se apropie 1 Mai și Gigi deja și-a planificat excursia la mare, acesta v-a rugat pe voi să îl ajutați și să scrieți 'măcar' partea de evaluare a expresiilor.

## 1. Introducere

O expresie aritmetică este alcătuită din operatori și operanzi (termenii asupra cărora se aplica operatorii). În cazul nostru, setul operatorilor va fi următorul:

- '-' - scădere
- '+' - adunare
- '\*' - înmulțire
- '/' - împărțire
- '^' - ridicare la putere
- 'sqrt' - radical

Se observă că primii 5 operatori sunt binari (au nevoie de 2 valori pentru a fi aplicați) iar cel de-al șaselea este unar (sqrt primește o singură valoare).

Pentru a asigura ordinea corectă a operațiilor, operatorii au priorități diferite (de la mic la mare):

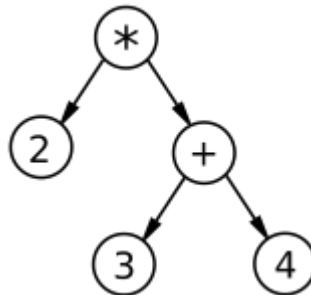
- '-' și '+' (cea mai mica prioritate)
- '\*' și '/'
- '^' și 'sqrt' (cea mai mare prioritate)

Această clasificare ne ajută la trecerea din forma infixată în forma prefixată.

## 2. Arbori de expresie

Pentru evaluarea unei expresii matematice, o abordare comună este cea a arborilor de expresii. Aceasta este o structură de date asemănătoare unui arbore (de cele mai multe ori, binar) în care nodurile sunt asociate operatorilor iar frunzele operanzilor.

De exemplu, expresiei  $2 * (3 + 4)$  îi corespunde următorul arbore de expresie:



În cazul expresiei de mai sus, se observă nevoia parantezelor pentru asigurarea ordinii dorite a operațiilor (în absența parantezelor arborele ar fi arătat diferit). Din acest motiv dorim să obținem o reprezentare mai abstractă a expresiilor, fără paranteze, care poate fi prelucrată mai ușor.

## 3. Forma prefixată

Forma prefixată (numită și forma poloneză) este o notăție matematică în care operatorii preced operanzii lor. Pentru expresia de mai sus, forma prefixată este următoarea:

Infix		Prefix
$2 * (3 + 4)$	=>	$* 2 + 3 4$
$2 * 3 + 4$	=>	$+ * 2 3 4$

După cum se observă, necesitatea parantezelor este eliminată de ordinea diferită a operatorilor.

## 4. Reguli de formare a expresiilor

O expresie poate fi reprezentată prin următoarele reguli:

**expr ::= variabilă | valoare**

**expr ::= expr + expr | expr \* expr | expr / expr | sqrt(expr) | expr ^ expr**

Acest lucru se interpretează astfel:

- O expresie este la bază o variabilă sau o valoare numerică
- O nouă expresie se poate construi prin adunarea a doua expresii, înmulțirea a doua expresii, împărțirea lor, etc.

## 5. Cerința

Dându-se o expresie aritmetică, construiți arborele de expresie aferent iar apoi evaluați rezultatul acesteia.

### a. Structura fișierelor de intrare

Structura fișierului de intrare este următoarea:

- Pe prima linie se găsește un număr întreg pozitiv care indică numărul de asignări de variabile (`nume_variabilă = valoare`)
- Pe următoarele N linii se găsesc asignări descrise mai sus
- Pe linia  $N + 2$  se găsește M ce reprezintă numărul de expresii care vor fi evaluate
- Pe următoarele M linii se găsește câte o expresie pe linie

### b. Structura fișierelor de ieșire

Fișierul de ieșire va consta în M linii, fiecare linie având rezultatul unei expresii din fișierul de intrare.

## 6. Restricții și precizări

- Toți operanzii pot fi reprezentați pe 32 de biți
- Numele variabilelor nu vor depăși 10 caractere
- Asignările sunt de forma '`nume_variabilă = valoare_numerică`' (nu se vor asigna alte expresii unei variabile)
- Valoarea din cadrul asignării variabilelor va fi întotdeauna un număr întreg valid
- În cazul în care '`sqrt`' se aplică unui număr negativ sau are loc o împărțire la 0, se va opri evaluarea expresiei și se va afișa '`Invalid expression`'
- În cazul în care se folosește o variabilă nedeclarată, se va opri evaluarea și se va afișa '`Variable <nume_variabilă> undeclared`'
- Temele care nu construiesc arborele și/sau se folosesc de alte metode de evaluare a expresiilor nu vor fi considerate
- Tema se va testa astfel: `./tema3 input_file output_file`

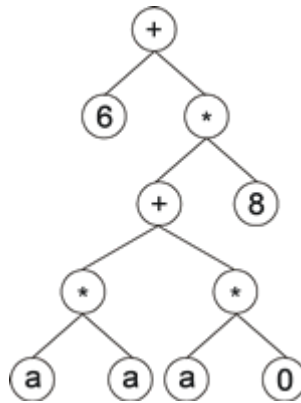
## 7. Arhiva

În arhiva pe care o veți încărca pe vmchecker se vor regăsi următoarele fișiere:

- Toate fișierele `.h` și `.c` în care ați lucrat
- Makefile cu reguli de build, clean și run
- README în care detaliați implementare voastră

## 8. Exemple

Pentru expresia '+ 6 \* + \* a a \* a 0 8', arborele de expresie va arăta astfel:



Exemplu de fișier de intrare, împreună cu rezultatul dorit

4 a = 1 b = 2 c = 3 aa = 4 5 + a - aa * b c sqrt + a + b c sqrt ^ b c - a d / a 0	-1 2 2 Variable d undeclared Invalid expression
---	---

0 3 + 4 -1 - 0 + 0 * 0 0 ^ 2 + 2 -1	3 0 2
---	-------------