

### 3. gyakorlat

#### Veremmel megoldható feladatok:

A következő feladatokban a beolvasáshoz az előadásjegyzetben is használt `read(&x : T):  $\mathbb{B}$`  függvényt használjuk. Ez a függvény beolvassa x-be a következő értéket a bemenetről, és ha ez sikeres, akkor igaz, ha pedig végére értünk az inputnak, akkor hamis értéket ad vissza.

**1. feladat:** Adott egy zárójelekből álló, legfeljebb  $n$  hosszú karaktersorozat a bemeneten. Olvassuk be, és döntsük el róla, hogy helyes zárójelezést határoz-e meg! (Vagyis, hogy párba állíthatók-e a zárójelek úgy, hogy minden nyitó zárójelnek van egy olyan csukó zárójel a párja, amely később következik a sorozatban.)

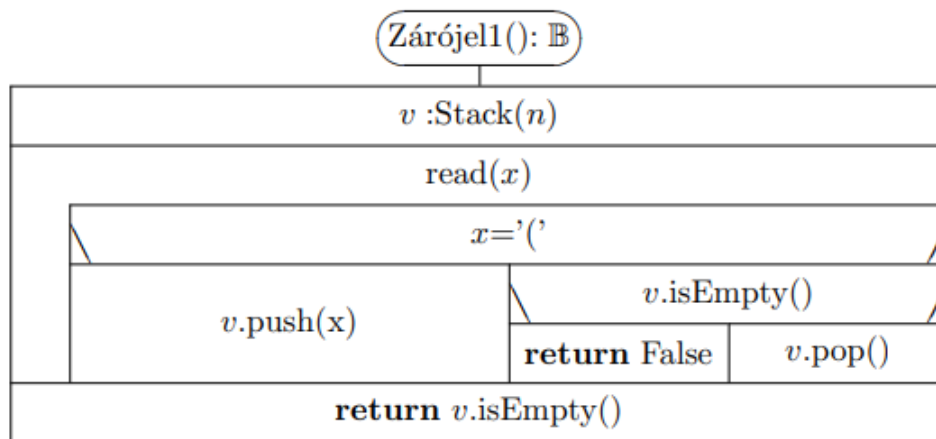
**2. feladat:** Adott egy zárójelekből álló, legfeljebb  $n$  hosszú karaktersorozat a bemeneten. Olvassuk be, és döntsük el róla, hogy helyes zárójelezést határoz-e meg! Írassuk ki az összetartozó zárójelpárok indexeit!

**3. feladat:** A bemenetről karakterenként beolvasunk egy (legfeljebb  $n$  karakterből álló) névsort, ahol a nevek egymástól vesszővel vannak elválasztva. Írjuk ki a neveket fordított sorrendben! Például: András,Béla,Csaba  $\rightarrow$  Csaba,Béla,András

Megoldások:

- Adott egy zárójelekből álló, legfeljebb  $n$  hosszú karaktersorozat a bemeneten. Olvassuk be, és döntsük el róla, hogy helyes zárójelezést határoz-e meg! (Vagyis, hogy párba állíthatók-e a zárójelek úgy, hogy minden nyitó zárójelnek van egy olyan csukó zárójel a párja, amely később következik a sorozatban.)

**Megoldás verem segítségével:**



Fejléc helyesebben: `Zárójel1(n:N):  $\mathbb{B}$`

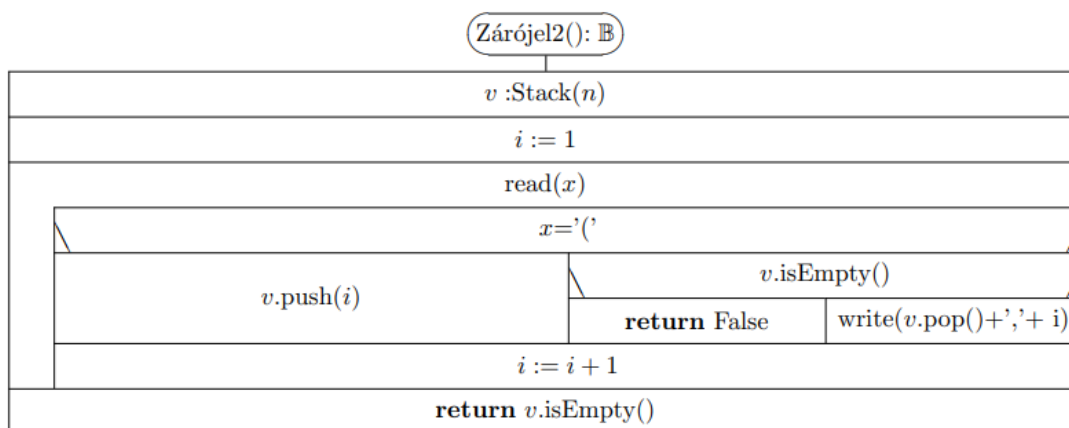
### 3. gyakorlat

#### Megjegyzések:

- Vegyük észre, hogy a fenti módszerhez valójában nincs is szükség verem használatára. Elég lenne egyetlen számláló változóval nyilvántartani azt, hogy eddig mennyivel több nyitó zárójellel találkoztunk, mint csukó zárójellel.
- Módosítsuk úgy a feladatot úgy, hogy tényleg szükség legyen a veremre: Ezúttal írassuk ki az összetartozó zárójelpárok indexeit!

- Adott egy zárójelekből álló, legfeljebb  $n$  hosszú karaktersorozat a bemeneten. Olvassuk be, és döntsük el róla, hogy helyes zárójelezést határoz-e meg! Írassuk ki az összetartozó zárójelpárok indexeit!

#### Megoldás:

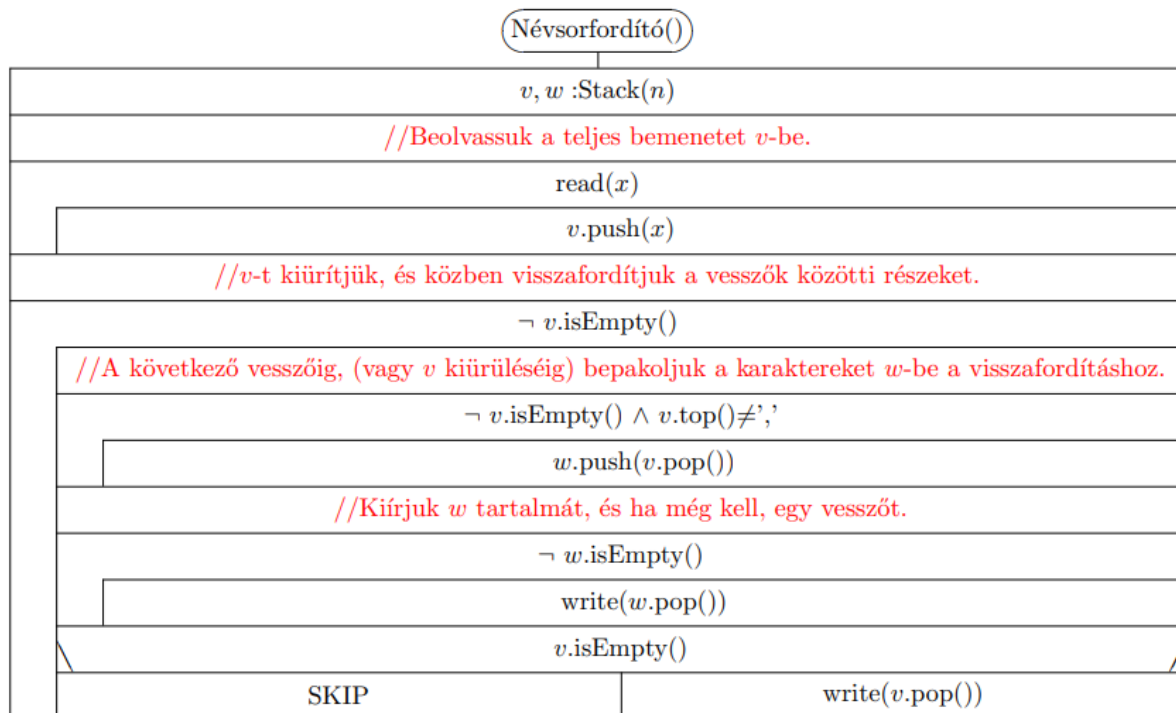


Fejléc helyesebben: Zárójel2(n:N): B

- A bemenetről karakterenként beolvassuk egy (legfeljebb  $n$  karakterből álló) névsort, ahol a nevek egymástól vesszővel vannak elválasztva. Írjuk ki a neveket fordított sorrendben! Például: András,Béla,Csaba  $\rightarrow$  Csaba,Béla,András

#### Megoldás:

### 3. gyakorlat



Fejléc helyesebben: Névsorfordító( $n:\mathbb{N}$ )

### Házi feladat

Tükrözött-e a szöveg? Adott egy legfeljebb  $n$  hosszú, betűkből és '#' szimbólumokból álló sorozat. A sorozatot tükrözöttnek nevezzük, ha felbontható olyan páratlan hosszú, palindrom karaktersorozatokról álló részekre, amelyeknek középső karaktere az egyetlen bennük szereplő '#'. Döntsük el a bemenetről olvasott szövegről, hogy tükrözött-e!

Példák:

- Tükrözött: #, #####, abc#cba, ##a#aabc#cba
- Nem tükrözött: abc, abc#cb, abc#cbaa#aa, ab#bac###c

### 3. gyakorlat

## Lengyel forma<sup>1</sup>

Lengyel forma: egy aritmetikai kifejezés postfix alakja. Jellemzői:

- Nincsenek benne zárójelek, a kiértékelés mégis egyértelmű, és könnyen elvégezhető,
- Operandusok sorrendje nem változik, az infix kifejezéshez képest,
- Operátorok sorrendje: az elvégzésük sorrendjében szerepelnek,
- Minden operátort közvetlen megelőznek az operandusai. Az operandus lehet változó, konstans, de lehet postfix kifejezés is.

infix kifejezés	lengyel forma (postfix alak)	Megjegyzés
$a+b$	$ab+$	műveleti jel az operandusai mögött áll
$a+b*c$	$abc*+$	műveletek rangsorának hatása: $\text{prec}(*) > \text{prec}(+)$
$a*b+c$	$ab*c+$	műveletek rangsorának hatása: $\text{prec}(*) > \text{prec}(+)$
$a*(b+c)$	$abc+*$	zárójelezés felülbírálhatja a műveletek rangsorát
$a/b*c$	$ab/c*$	azonos rangú műveletek általában balról jobbra sorrendben végzendők el
$a^b^c$	$abc^^$	a fenti szabály alól akad néhány kivétel, például az egymást követő hatványozás sorrendje jobbról balra értendő

#### Feladat:

A)  $(a+b) * (c-d) / f^{(g-h)} + j-l-i$  kifejezés lengyel formára hozása.

#### Megoldás:

$$a b + c d - * f g h - ^ / j + l - i -$$

B) Értékadó operátor hatása, Hova illik az értékadó operátor?

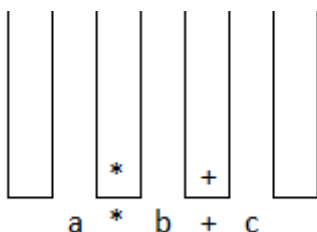
$$x = (a+b) * (c-d) / f^{(g-h)} + j-l-i$$

#### Megoldás:

$$x \quad a b + c d - * f g h - ^ / j + l - i - =$$

## Lengyel formára hozás verem segítségével – bemutatás példákon keresztül

### 1. $a * b + c$

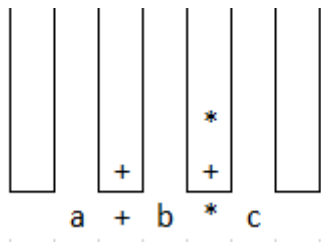


**Kimenet:**  $a b * c +$

### 2. $a + b * c$

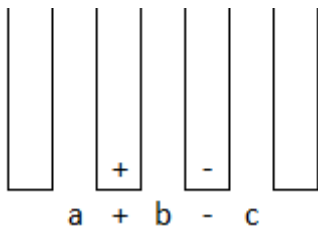
<sup>1</sup> Az eredeti prefix jelölési formát, **Jan Łukasiewicz** lengyel matematikus javasolta 1920-ban, később az ausztrál filozófus, **Charles Leonard Hamblin** javasolta a postfix alakot (1950), melyet emiatt „fordított lengyel formának” is szokás nevezni. (forrás: wikipedia)

### 3. gyakorlat



**Kimenet:** a b c \* +

#### 3. a + b - c



**Kimenet:** a b + c -

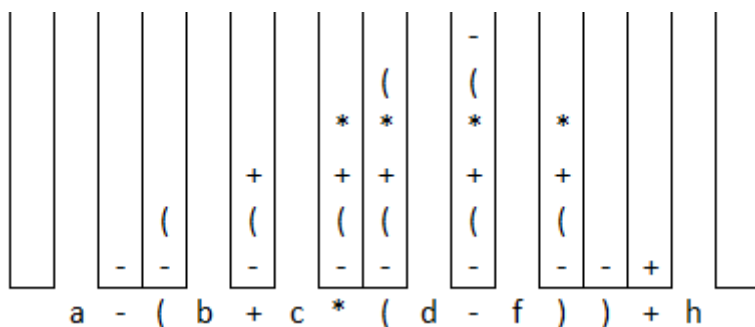
#### Precedencia hatása:

- Minden beolvasott műveleti jel bekerül a verembe, hogy „megvárja”, míg az operandusai kiíródnak, de előtte a veremben várakozó műveleti jelek vizsgálata történik:
- ha azonos rangú a beolvasott és a verem tetején lévő műveleti jel, kiírjuk a veremben lévő (balról jobbra sorrend esetén) – 3. példa,
- ha a veremben magasabb prioritású művelet szerepel, mint ami bekerülne, kiírjuk – 1. példa,
- ha a verem tetején alacsonyabb rangú van, mint az olvasott, akkor bekerül a verembe – 2. példa.

#### Feladat:

$a - (b + c * (d - f)) + h$  kifejezés lengyel formára hozása verem segítségével. A verem tartalmát folyamatosan tartjuk nyilván!

#### Megoldás:



**Kimenet:** a b c d f - \* + - h +

Kicsit bonyolultabb kifejezés gyakorlásra:

$$x = a + (-b^c + d * e) / ((f + g) * h / -k) - p * z$$

### 3. gyakorlat

#### Algoritmus:

Bemenet: egy helyesen zárójelezett kifejezés elemenként. (Az operandusokat, mint szimbólumokat tesszük a verembe. Feltesszük, hogy a kifejezés csak bináris operátorokat tartalmaz.)

LengyelForma(n)				
V: Stack(n)				
Read ( x )				
Operandus(x)	x = ' ( '	x = ' ) '	Operator(x)	
Write(x)	V.push ( x )	V.top ≠ '('	BalJobbOperator(x)	
		Write(V.pop())	$!V.IsEmpty() \wedge V.top() \neq '('$ $\wedge pr(x) \leq pr(V.top())$	$!V.IsEmpty() \wedge V.top() \neq '('$ $\wedge pr(x) < pr(V.top())$
		V.pop()	Write(V.pop())	Write(V.pop())
		V.push(x)	V.push(x)	
$! V.IsEmpty()$				
Write(V.pop())				

#### Megemlíthető:

- Vannak jobbról balra sorrendű operátorok, ezek feldolgozása hogyan történik?  
 $y = x = a \wedge b \wedge c$
- Gondoljuk meg, hogy az egy operandusú operátorok (pl. negatív előjel  $-a^b$ , vagy  $++i*x$ ) hogyan illeszthetők be a lengyel formába?
- Egyszerű függvények bevonása. pl:  $x = z * \sin(y/w)^2$
- Javasoljuk a hallgatóknak, hogy keressék meg az interneten a C++ nyelv operátorait és precedenciájukat, például: <http://www.cplusplus.com/doc/tutorial/operators/>

### Lengyel forma kiértékelése

#### Feladat:

Hozzuk lengyel formára a következő kifejezést, majd verem segítségével értékeljük ki:

$(a + b) * c - d$

**Lengyel forma: a b + c \* d -**

Tegyük fel, hogy a változók az alábbi értékkel rendelkeznek, számítsuk ki a kifejezés értékét a lengyel formából!

a = 2, b = 4, c = 3, d = 1

#### Kiértékelés:

	4		3		1		
2	2	6	6	18	18	17	
2	4	+	3	*	1	-	

### 3. gyakorlat

#### Az algoritmus:

Bemenet: egy lengyel formájú kifejezés elemenként. (Az operandusok értékét tesszük a verembe. Feltesszük, hogy a kifejezés csak bináris operátorokat tartalmaz, de értékadó operátort nem.)

lengyel_kiertekeles(n)	
V:Stack(n)	
read(x)	
<div><div>Operator(x)</div><div>jobb:=V.pop()</div><div>bal:=V.pop()</div><div>V.push(bal ⊗ jobb)</div></div>	<div><div>Operandus(x)</div><div>V.push(x)</div></div>
write(V.pop())	

Megjegyzések az Operator(x) ághoz:

- bal  $\otimes$  jobb jelölés: elvégezzük az x műveletet,
- feltettük, hogy az operátorok két operandusúak, de ez könnyen kiterjeszthető egy operandusú műveletekre
- az algoritmus kiszámíthatja az értéket, vagy fordító program esetén generálhatja a kiszámítás kódját.

#### Javasolt házi feladatok lengyel forma témakörhöz:

- (1) Teljesen és helyesen zárójelezett kifejezésből hogyan állítható elő a lengyel forma.
- (2) Teljesen és helyesen zárójelezett kifejezésből hogyan értékelhető ki verem segítségével a kifejezés.
- (3) Lengyel formából hogyan állíthatjuk elő a teljesen zárójelezett alakot.