
Interpret de mini-llenguatge de programació

X29499_ca

PRELIMINARS:

En aquest exercici assumim que heu resolt alguns exercicis anteriors relacionats amb un llenguatge de programació molt bàsic, que queda exemplificat com segueix:

```
x=3
y=3+x
z=2*(x+y)
Print(z-y+x)
```

L'execució del programa anterior escriuria 15 per la sortida estandard. Amb anterioritat, heu creat els següents fitxers:

- Fitxer `evaluate.cpp`, que implementa una funció `evaluate`, la qual rep un `map<string, int>` que guarda els valors de les variables fins al moment, i un arbre binari que representa una expressió del llenguatge, i retorna l'avaluació de l'arbre.
- Fitxer `execute.cpp`, que implementa una funció `execute`, la qual rep un `map<string, int>` que guarda els valors de les variables fins al moment, i un arbre binari que representa una instrucció del llenguatge (assignació o escriure), i o bé modifica l'estat d'una variable (si és una assignació) o bé escriu per la sortida (si és una escriptura).
- Fitxer `scanner.cpp`, que implementa una funció `scanner`, la qual rep un `string` amb una instrucció del programa, i retorna la llista d'elements atòmics del llenguatge que apareixen en aquest string.

Ara us oferim dos fitxers més, anomenats `parser.hpp` i `parser.cpp`, que declaren i defineixen una funció `parser`, la qual rep una llista d'elements atòmics del llenguatge que representen una instrucció, i retorna un arbre binari que representa la mateixa instrucció.

EXERCICI:

Usant les funcions dels fitxers anteriors, haureu de crear un fitxer `program.cpp`, amb una funció `main`, que implementa un interpret del llenguatge.

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile`, `program.cpp`, `BinaryTree.hpp`, `evaluate.hpp`, `execute.hpp`, `scanner.hpp`, `parser.hpp`, `parser.cpp`, `utils.hpp`, `utils.cpp`. Us falta afegir els fitxers `evaluate.cpp`, `execute.cpp` i `scanner.cpp` que teniu fets d'un exercici anterior, i crear el fitxer `program.cpp` i implementar-hi el programa principal. Quan pugueu la vostra solució al jutge, només cal que pugueu un tar construït així:

```
tar cf solution.tar program.cpp execute.cpp evaluate.cpp scanner.cpp
```

Entrada

L'entrada té una seqüència línies, on cadascuna és una instrucció del llenguatge.

Sortida

El programa dona com a sortida el que seria la sortida resultant d'executar la seqüència d'instruccions donada d'entrada.

Exemple d'entrada 1

```
a=3*2+1
Print (a-(7-a)-(2*a))
a=a+6-3
Print (a-(3*a))
b=a*a-(a*2)
Print (4*a)
Print (b-(2-4))
Print (3*b)
b=7-4-(a*4)
Print (a+3-b+(7-3))
```

Exemple de sortida 1

```
-7
-20
40
82
240
54
```

Exemple d'entrada 2

```
a=8*2+ 7
b=a+9-2
Print (b- 5-(b-2) )
Print ((a- a)*(7-a))
Print (2*( a+3+(7-2)))
a=(b+a)*5+(7-3- 9)
c=(4+9)*(b* 3)
c=c- c
a=8*c
c=(9+7) *(6+c) -c
Print (8*8)
Print (3*(b*1))
b=c+a
Print ( 9*b-5-( c- 1))
Print ( 9*c+2)
Print (b*(3*4))
c=4*b
b=(1+c)* (a-4)-( 2+6)
Print (5+c)
Print ((a-4) *(1* a-a*c))
```

Exemple de sortida 2

```
-3
0
62
64
90
764
866
1152
389
0
```

Exemple d'entrada 3

```
a=36*87+ 50
Print ((63+68)* 3-(a+12) )
Print (6-(47-a-(65+a- a)*85))
a=a-2
a=a-7-72*30+ (16- (a-46) ) -(42-(35+
b=77+a-(13* 73-(43-( a+ 20-( 12-68)))
Print (b- 43+(22+5)+(a-85*43)-9- (34*91-
a=a- a-a+(b-b+(b+a)) - ( a+87)+(53-
Print (94-b-(b-63)-( b-74- (19-37)-(a+
Print ( 73-b+98- (35-b-a-(17-a)) )
a=(62-52)* 7-(5+ b-(37+57))
Print (87-a-(24-1+a)-(77*20-(82-a) -(3-32))
b=a-12+( 43+ 40)+( 95-(61- 93))
Print (52+40- (b-8-(a+44-(b-a)-87*53)) )
a=62*35+(b+34) -(98-30-(a-86) )
Print ( 45+21-b+(87-a+(83-b-44+ (82-
a=22- (83*12+b)- (77- 74- 55) -
Print ( 6-a)
Print (a+59- (11*91-(b-40)))
Print (61-71)
```

Exemple de sortida 3

```
-2801
8666
-9699
5791
253-(60-33-29)-(36-19+( a+5))+29* (a- a)*(45-a-(
5082
474863 a-(64+ 54-97*89))) )
-71968-58) )+ (a-42+b+ (18-(38-b)+(b-79-(a+ 19) +
b-740)-(96-50- (b- 61+ ( b+a))) ) +( a+15+
7746
-10
- (58-26-(b-27)- (99+63- 81+a)) +(91*92-(b- 81)+(b
- ( 13-56+(b- 43)- b)+((72+ (43- 13+( b-b)) )*27+
```

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T21:07:18.588Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>