

Clojure - Petit LP

Volem fer un petit intèrpret capaç d'avaluar expressions aritmètiques simples i executar 2 accions simples.

Expressions:

Les expressions seran combinacions de les comandes següents amb parentesis:

Comanda	Exemple	Explicació
:val	(:val 2) ↗ 2	Valor
:var	(:var "a") ↗ 2	Valor d'una taula de símbols
:add	(:add (:val 1) (:val 2)) ↗ 3	Suma
:sub	(:add (:val 4) (:val 2)) ↗ 2	Resta
:mul	(:add (:val 1) (:val 2)) ↗ 2	Multiplicació
:div	(:add (:val 2) (:val 1)) ↗ 2	Divisió

Accions:

Els programes seran llistes de les accions següents:

Comanda	Exemple	Explicació
:view	(:view (:val 2)) ↗ 2	Mostra un valor
:ass	(:ass "a" (:val 2))	Assignació a una taula de símbols

Un exemple de programa seria:

```
(:ass "a" (:val 2))
(:view (:div (:var "a") (:val 0))) ↗ >> Divisió per zero!
(:view (:add (:var "a") (:val 1))) ↗ >> 3

Taula de símbols final : {"a" 2}
```

Enunciat:

1. Definiu una funció `interpretar` que interpreti un programa com l'anterior i que sigui capaç de processar tot el que està especificat en les taules anteriors.

Joc de proves:

```
(interpretar '(:ass "a" (:val 2))
             (:view (:div (:var "a") (:val 0)))
             (:view (:add (:var "a") (:val 1)))))
```



```
>> Divisió per zero!
>> 3
>{"a" 2}
```

Regles del joc:

- Heu de fer el treball en parelles.
- Heu de lliurar un arxiu `clj` amb el nom i cognoms dels dos membres del grup en un comentari a la part de dalt de l'arxiu.
- Lliureu només un dels membres del grup.