



Problema 7

28 / 30 Novembre 2023

Descrizione

Gli esercizi qui proposti hanno come oggetto sequenze *ordinate* di numeri interi, rappresentate attraverso le liste — liste ordinate come argomento, relativamente ai punti 1, 2 e 3, oppure da costruire nel caso del punto 4. Per operare sulle liste utilizza esclusivamente la costante `null` e le primitive di base: `null?`, `car`, `cdr`, `cons`.

1. Innanzitutto, definisci in Scheme una procedura `belong?` che, dati un intero x e una lista ordinata di interi S , verifica se x è un elemento di S .

Esempi

```
(belong? 18 '()) → false
(belong? 18 '(5 7 10 18 23)) → true
(belong? 18 '(5 7 10 12 23)) → false
```

2. Successivamente, definisci in Scheme una procedura `position` che, dati un intero x e una lista ordinata di interi senza ripetizioni S , restituisce la posizione (indice) di x in S . A tal fine, assumi che x sia un elemento di S .

Esempi

```
(position 7 '(7 8 24 35 41)) → 0
(position 35 '(7 8 24 35 41)) → 3
(position 41 '(7 8 24 35 41)) → 4
```

3. Definisci quindi una procedura `sorted-ins` che, dati un intero x e una lista ordinata di interi senza ripetizioni S , restituisce la lista *ordinata e senza ripetizioni* che contiene x e tutti gli elementi di S .

Esempi

```
(sorted-ins 24 '()) → '(24)
(sorted-ins 5 '(7 8 24 35 41)) → '(5 7 8 24 35 41)
(sorted-ins 24 '(7 8 24 35 41)) → '(7 8 24 35 41)
(sorted-ins 27 '(7 8 24 35 41)) → '(7 8 24 27 35 41)
```

4. Infine, applica `sorted-ins` per definire una procedura `sorted-list` che, data una lista di interi S , restituisce la lista *ordinata e senza ripetizioni* S' che contiene tutti e soli gli interi di S .

Esempi

```
(sorted-list '(35 8 41 24 7)) → '(7 8 24 35 41)
(sorted-list '(35 24 8 41 24 7)) → '(7 8 24 35 41)
```