

Prove varie con uso di liste

Provo a realizzare un documento wxm con esempi di blocchi, test, cicli iterativi, liste ed operazioni sulle liste...
Ecco subito un blocco di tre istruzioni:

```
(%i1) (a:7, b:3, d:5);
(%o1) 5
```

```
(%i2) b:1;
(%o2) 1
```

I blocchi di istruzione sono delimitati da parentesi tonde e ogni istruzione è separata da un'altra da una virgola. L'istruzione `if...then...else` va contenuta in un blocco dove le parole chiave `if`, `then` ed `else` hanno lo stesso ruolo di separatore delle virgole.

```
(%i3) (if (d:8,b>=3) then (a:1,c:3) else (a:-1,c:-3),a>7);
(%o3) -1>7
```

Notare che l'ultima espressione del blocco iniziante con `if` viene valutata considerando come operatori relazionali `=`, `>`, `>=`, `<`, `<=` mentre l'ultima espressione del blocco, come si vede, NON viene valutata.

```
(%i4) d;
(%o4) 8
```

I cicli iterativi sono costituiti da una sequenza di quattro blocchi separati dalle parole chiave: `for from thru do`. Dopo il `for` si specificava la variabile contatore del ciclo. La variabile viene incrementata di una unità a partire dal valore specificato dopo `from` e fino al valore indicato dopo `thru` (incluso). Il blocco iterato è quanto specificato dopo il `do`.

```
(%i5) (a:0, for k from 1.5 thru 3.5 do (a:a+k), a);
(%o5) 7.5
```

Posso avanzare di una quantità diversa da uno usando la parola chiave `step`:

```
(%i6) (s:0, for k step 2 from 1 thru 6 do (s:s+k), s);
(%o6) 9
```

Definisco esplicitamente la lista `aa`: (notare che faccio un esempio di comando istruzione ossia tutto ciò che sta tra `/*` e `*/`).

```
(%i7) aa: [1,2,3,4,5] /* aa è una lista */ ;
(%o7) [1,2,3,4,5]
```

[/ Voglio sapere quanti valori ho definito fino a questo punto:

```
(%i8) values;  
(%o8) [a, b, d, c, s, aa]
```

[/ Per scoprire la lunghezza di una lista uso la funzione length(lista):

```
(%i9) length(aa);  
(%o9) 5
```

[/ Ecco due modi di usare la funzione makelist con quattro o tre argomenti. Con quattro makelist(espressione,variabile,inizio,fine) variabile rappresenta un nome di variabile che assume il valore inizio al primo elemento della lista ed e' incrementata di uno fino a che supera il valore fine. L'espressione deve ovviamente far uso della variabile, altrimenti tutti gli elementi della lista risultato saranno uguali.

```
(%i10) bb: makelist((2*k+1)^2,k,0,5);  
(%o10) [1, 9, 25, 49, 81, 121]
```

[/ Quando invece makelist ha tre argomenti opera in questo modo: makelist(espressione,variabile,listacontrollo) e produce una lista con tanti elementi quanti ne ha listacontrollo. Ad ogni passo la variabile assume il valore del corrispondente elemento della listacontrollo.

```
(%i11) cc: makelist(x+1/2,x,[-1,3,13,7,5]);  
(%o11) [-1/2, 7/2, 27/2, 15/2, 11/2]
```

[/ Per concatenare due liste va usata la funzione append(lisa1,lista2)

```
(%i12) append(bb,cc);  
(%o12) [1, 9, 25, 49, 81, 121, -1/2, 7/2, 27/2, 15/2, 11/2]
```