7.16 Write one or more rules that will generate all the permutations of a base fact and print them out. For example, the fact:

```
(base-fact red green blue)
```

should generate the output:

Please input a base fact for the permutation ... red green blue

Permutation is (red green blue)
Permutation is (red blue green)
Permutation is (green red blue)
Permutation is (green blue red)
Permutation is (blue red green)
Permutation is (blue green red)
The total number is 6

作業#3:
排列組合

<Hint:>

(declare (salience n)) 設定此規則的
優先權為n，未設定優先權為0

(readline) 可以從鍵盤輸入一行字串

(explode$?str) 可以將一個字串轉
成為multifield欄位的值

(bind ?x ?y) 將?y的值設定給?x變數

```
(deftemplate permutation (multislot values) (multislot rest))

(deffacts initial (total 0))

(defrule read-base-fact
  (declare (salience 100))          ;;-10000~10000, default is 0
  =>
  (printout t "Please input a base fact for the permutation ... ")
  (bind ?input (explode$ (readline)))
  (assert (permutation (values) (rest ?input))))
```

(example 11 22 33 44)
會讓以下的規則

```
(defrule hint
  (example $? ?x $?)
  =>
  (assert (data ?x)))
```
執行4次，產生4個事實
```
(data 11)
(data 22)
(data 33)
(data 44)
```