

SPECIFICATION:

Input: $N \in \mathbb{N}$, wines $[1..N] \in \text{wine}^N$

wine = (name \times type \times amount \times price),
name = T, type = T, amount = N, price = N

Output: result $\in \text{output}^N$,

output = (name \times type \times amount),
name = T, type = T, amount = N

Precondition: $1 \leq N \leq 100$

and $\forall i (1 \leq i \leq N) : 1 \leq \text{wines}_i.\text{amount} \leq 10,000$

and $\forall i (1 \leq i \leq N) : 1 \leq \text{wines}_i.\text{price} \leq 10,000$

Postcondition:

wines [] = $\bigvee_{i=1}^{\text{cnt}}$ (wines_i.name = name And wines_i.type = type)

wines[] = $\sum_{i=1}^{\text{cnt}}$ (wines_i.amount + amount And wines_i.price + price)

result = $\text{MAX}_{i=1}^{\text{cnt}}$ (wines_i.price)

PATTERN: Decision, Sequence Calculation, Maximum Selection

Decision General Algorithm:

<code>i:=0</code>
<code>exists:=false</code>
<code>i≤length(X) and not exists</code>
<code>i:=i+1</code>
<code>exists:=A(X[i])</code>

Sequence Calculation General Algorithm:

<code>sc:=F0</code>
<code>i=1..length(X)</code>
<code>sc:=f(sc,X[i])</code>

Maximum Selection General Algorithm:

<code>maxVal:=X[1]</code>
<code>i=2..length(X)</code>
<div><div><div><code>T</code></div><div><code>X[i]>maxVal</code></div><div><code>F</code></div></div><div><code>maxVal:=X[i]</code></div><div><code>-</code></div></div>

ALGORITHM PATTERN:

Pattern(Decision)		Task
length(X)	→	cnt
A(X[i])	→	wines _j .name = name AND wines _j .type = type
exists	→	exists

Pattern(Sequence)		Task
length(X)	→	cnt
sc	→	wines _j .name + amount; wines _j .price + price;
X [i]	→	The wines between 1 and N

Pattern(Maximum Selection)		Task
length(X)	→	cnt
maxVal		max_result
X []	→	wines _j .price > maxPrice
X [i]	→	The wines between 1 and N

