

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:**

cnt

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

cnt

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

cnt

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

PATTERN: Decision, Sequence Calculation, Maximum Selection

Decision General Algorithm:

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

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

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

Algorithm :

Main:

main()

INPUT : N , wines []

wines := read_arrange (wines, N)

result := findMaximum(wines, N)

OUTPUT : result

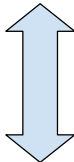
Decision:

$i := 1$

$i \leq \text{length}(X)$ and not $A(X[i])$

$i := i + 1$

exists := ($i \leq \text{length}(X)$)



alreadyAMember(wines[] : wine, input : String, cnt : Integer) : Boolean

$i := 1$

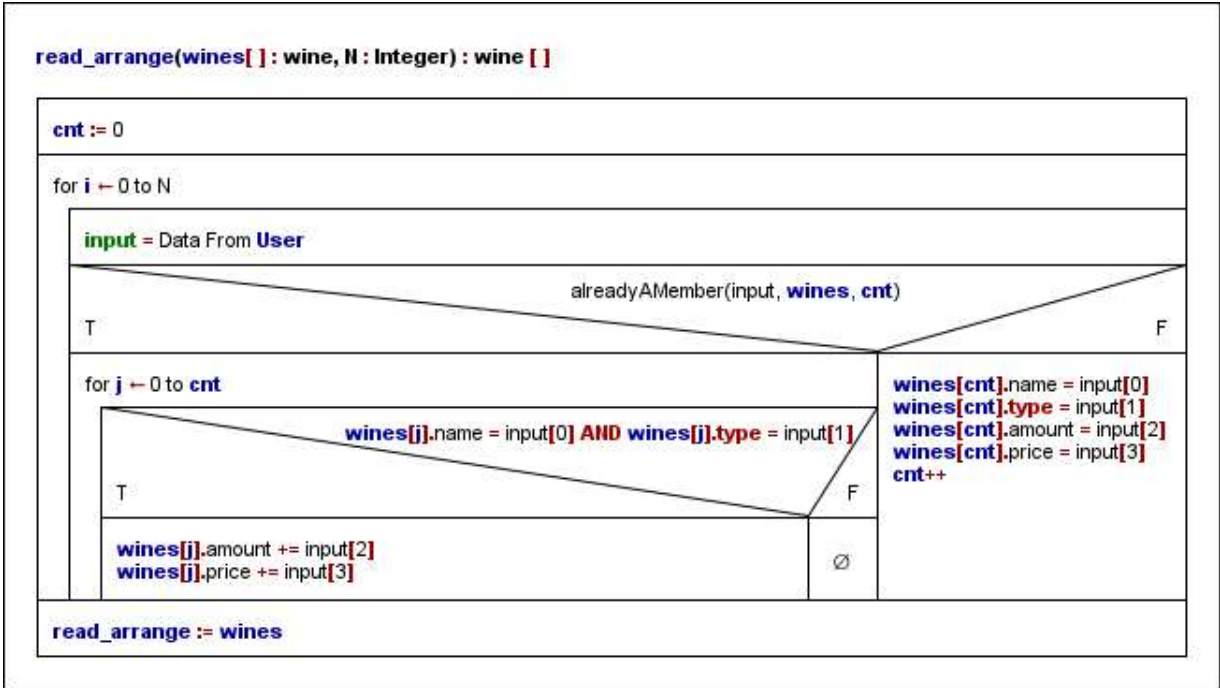
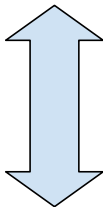
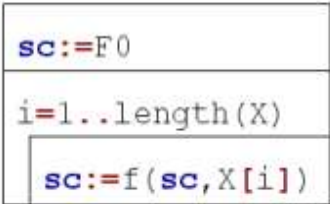
$i < \text{cnt}$ and wines[i].name = name and wines [i].type = type

$i := i + 1$

exists := $i \leq \text{cnt}$

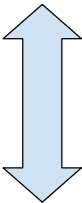
alreadyAMember := exists

Sequence Calculation :



Maximum Selection :

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



findMaximum(wines[] : wine, cnt : Integer) : Output	
<code>max_result.name = wines[0].name</code>	
<code>max_result.amount = wines[0].amount</code>	
<code>max_result.type = wines[0].type;</code>	
<code>maxPrice = wines[0].price</code>	
<code>for i ← 1 to cnt</code>	
<div><div></div><div><div><code>wines[i].price > maxPrice</code></div><div>T</div><div>F</div></div></div>	
<code>max_result.name = wines[i].name</code> <code>max_result.amount = wines[i].amount</code> <code>max_result.type = wines[i].type</code> <code>maxPrice = wines[i].price</code>	Ø
<code>findMaximum := max_result</code>	