B2. Task

Specification:

Input: $numOfStat \in \mathbb{N}$, $on[1..numOfStat] \in \mathbb{N}^{numOfStat}$, $off[1..numOfStat] \in \mathbb{N}^{numOfStat}$, $seem: \mathbb{N} \to \mathbb{N}$;

$$seem(n) = \sum_{i=0}^{n} (on[i] - off[i])$$

Output: $numOfPassEachStation[maxInd] \in \mathbb{N}^{numOfStat}$

Precondition: $1 \leq numOfStat \leq 100$, $\forall i (1 \leq i \leq numOfStat)$: $0 \leq on_{[i]}$, $\forall i (1 \leq i \leq numOfStat)$: $50 \geq off_{[i]}$

Postcondition:

$$maxNumOfPass := \max_{i=1}^{numOfStat} \left(\forall i (1 \leq i \leq numOfStat): numOfPassEachStation[i] = seem(i) \right)$$

Summation (Pattern table for seem() function in input)

Pattern		Taste
length (x)	→	n
x[]	→	on[] - off[]
sc	→	seem
FO	→	0
f()	→	+

Maximum selection

Pattern		Task
length(x)	→	numOfStat
x[]	→	numOfPassEachStation[]
maxInd	→	maxInd

Copy

Pattern		Task
length(x)	→	numOfStat
x[]	→	on[], off[]
y[]	→	numOfPassEachStation[]
f(i)	→	seem(i)

// Comment: Since I have defined seem(i) function earlier in the input, as a value of f(i) In the copy pattern of algorithm I indicated only the name of the function. If the detail of the function is important, I would have written: $\sum_{j=0}^{i} on[j] - off[j]$. Pattern table →

→ for the seem() function you can find in the First page!

MaxNumOfPassenger

In: numOfStat, on[], off[]

sum := 0, maxInd := 0; numOfPassEachStation[]

i := 1 .. numOfStat

sum := sum + (on[i] - off[i])

numOfPassEachStation[i] = sum

I := 1 .. numOfStat

numOfPassEachStation[maxInd] < numOfPassEachStation[i]

maxInd := i | skip

Out: numOfPassEachStation[maxInd]

Algorithm ↑