

Algorithm Test Solution

Programming L+Pr. (IP-18fPROGEG) - 2022 autumn semester

Table of contents

- [Algorithm Test Solution](#)
 - [A1](#)
 - [A2](#)
 - [A3](#)

This is a sample solution for the 2022 autumn semester algorithm exam.

Alg.test – 2022.12.01. – Group A

In this test, you need to write (draw) 4 specifications and small algorithms on paper with pen. Please look after the good, readable quality of your work.

There are 80 minutes available. For a good solution maximum 80 points could be reached. Write (draw) only on paper (don't forget your name and neptun-code). During work you are not allowed to talk (chat), to use cell phones or the internet (only to upload your work). You should use printed or written papers.

Remember! Cheating is not allowed, and action will be taken.

A1

In the Infinitive Hotel the room's price are S HUF. We store the dates of the check in and the check out of N guests. Give the income of the hotel. (*Input, output, postcondition, algorithm, PoA's name*)

Example input:

Input	Output
S=21000HUF; N=3	504000
1 guest: 12 – 23	(calculation: $21000 \cdot (12+8+4)$)

2 guest: 5 – 12

3 guest: 2 - 5

PoA's name: Sequence calculation

Specification

Input:

$cost \in \mathbb{N}, numberOfGuests \in \mathbb{N},$
 $checkIn[1..numberOfGuests] \in \mathbb{N}^{numberOfGuests},$
 $checkOut[1..numberOfGuests] \in \mathbb{N}^{numberOfGuests}$

Output:

$income \in \mathbb{N}$

Postcondition:

$$income := \sum_{i=1}^{numberOfGuests} checkOut[i] - checkIn[i] + 1$$

Algorithm

In: cost, numberOfGuests, checkIn[], checkOut[]
income := 0
i = 1..numberOfGuests
income := checkOut[i] - checkIn[i] + 1
Out: income

A2

We have an integer sequence. Determine the greatest number which is less than K.
(Input, output, postcondition, algorithm, PoA's name)

Example input:

Input	Output
K=3;	2
Sequence: 4, 6, 2, 1, -2, 0, 4, 1	

PoA's name: Maximum selection

Specification

Input:

$limit \in \mathbb{N}, sequence[1..] \in \mathbb{N}^*$

Output:

$maxNumber \in \mathbb{N}$

Postcondition:

$$maxNumber := \begin{matrix} length(sequence) \\ MAX \\ i = 1 \end{matrix} (sequence[i] < limit)$$

Algorithm

In: limit, sequence[]	
maxNumber := 0	
i = 1..length(sequence)	
(sequence[i] > maxNumber AND sequence[i] < limit)	
T	F
maxNumber := sequence[i]	-
Out: maxNumber	

A3

We store in a matrix the scores that N gamers reached in M computer games. We know the maximum score for each game. Determine a game where nobody has a maximum score.

(Input, output, postcondition, algorithm, PoA's name)

Example input:

Input

N=3, M=4

MaxScores: 34,12,50,35

Output

4

	1.game	2.game	3.game	4.game
1. gamer	12	12	5	30
2. gamer	23	4	12	23
3. gamer	34	12	50	30

PoA's name: Selection

Specification

Input:

$N \in \mathbb{N}, M \in \mathbb{N},$

$maxScores[1..M] \in \mathbb{N}^M,$

$matrix[1..N][1..M] \in \mathbb{N}^{N \times M}$

Output:

$result \in \mathbb{N}$

Postcondition:

$$result := \begin{matrix} N, M \\ SELECT \\ i, j = 1 \end{matrix} (j)$$

$$\forall matrix[i][j] < maxScore[j]$$

Algorithm

