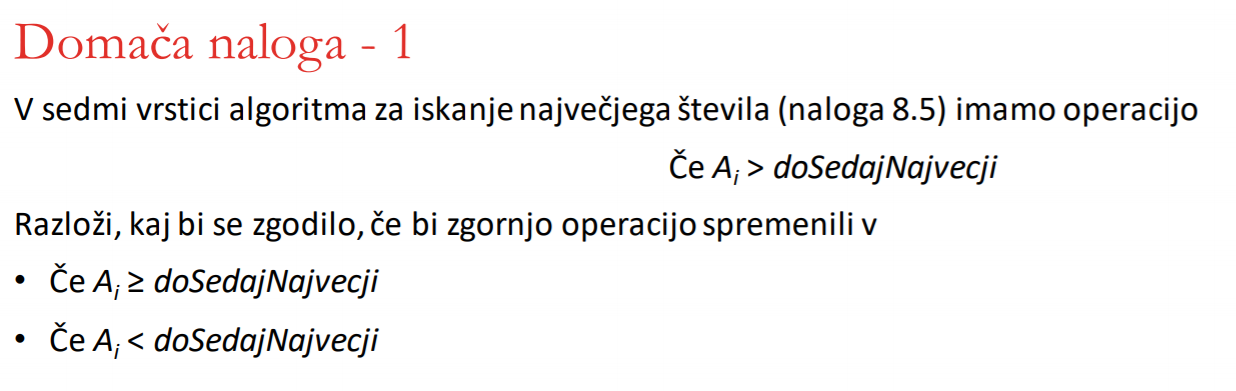
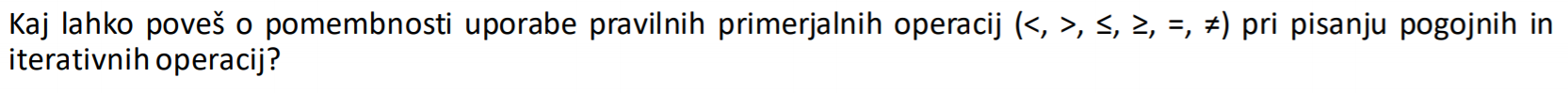
**VAJE 8 – Domača naloga**

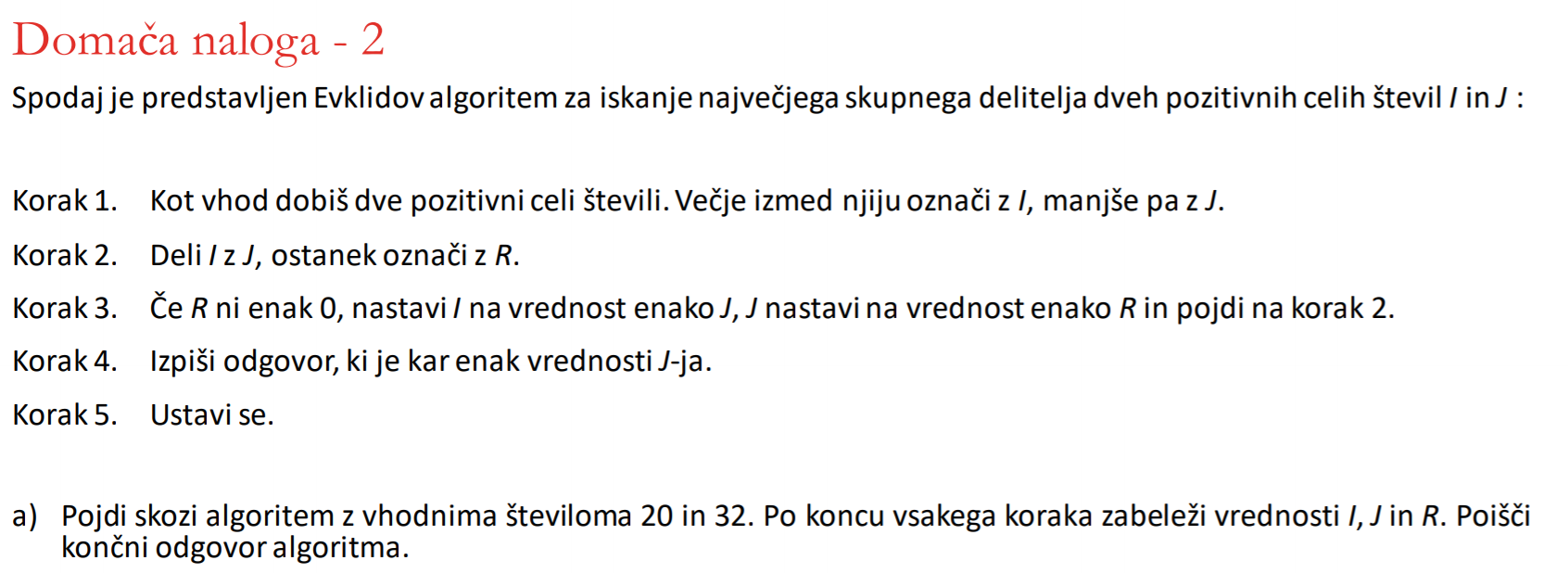


Dobili bi pravilen rezultat, a za razliko od prej bi algoritem izpisal zadnjo pojavitev največjega števila in njen položaj (če bi bilo pojavitev seveda več).

Dobili bi napačen rezultat, saj bi nam algoritem namesto največjega izpisal najmanjše število (in položaj) v danem seznamu.



Odg.: Pri pisanju primerjalnih operacij v pogojnih ali iterativnih stavkih je potrebno biti zelo natančen, saj nam lahko napačna operacija obrne celoten program (primer je lahko kar ta zgoraj, če bi obrnili simbol za večje v manjše, dobimo čisto nasproten rezultat). Prav tako lahko že s pozabljenim ali pa nepotrebnim enačajem pri večje ali enako, dobimo čisto napačen rezultat ali pa (pri iterativnih operacijah) indeksiramo izven področja (»out of range«), zaradi česar se program ne more izvesti.



Korak 1.: I = 32, J = 20, R še nima vrednosti Korak 2.: I = 32, J = 20, R = 12

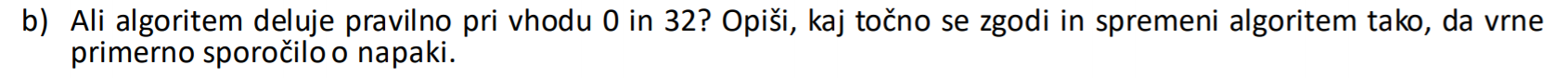
Korak 3.: I = 20, J = 12, R = 12 Korak 2: I = 20, J = 12, R = 8

Korak 3.: I = 12, J = 8, R = 8 Korak 2.: I = 12, J = 8, R = 4

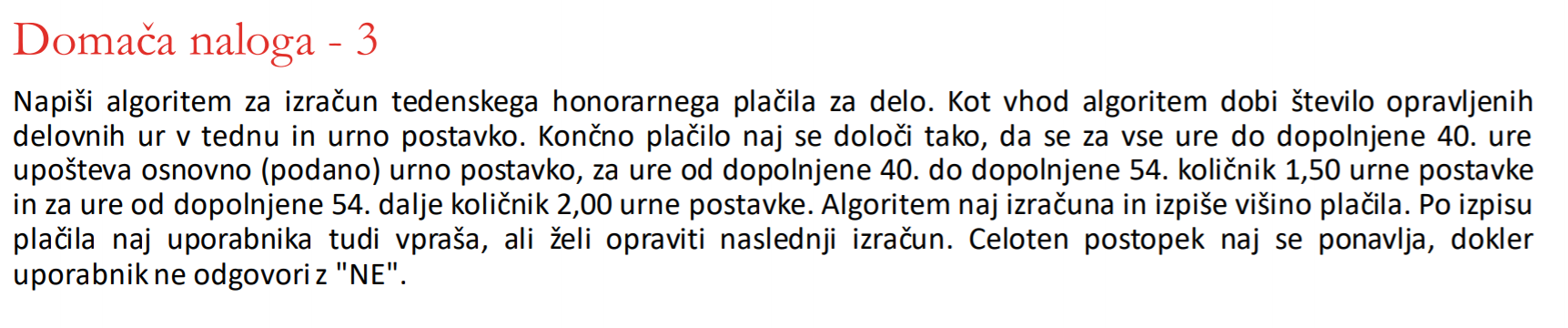
Korak 3.: I = 8, J = 4, R = 4 Korak 2.: I = 8, J = 4, R = 0

Korak 4.: I = 8, J = 4, R = 0 Korak 5.: I = 8, J = 4, R = 0

Odg.: Končni odgovor je enak končni vrednosti J-ja, torej 4.



Odg.: Ne deluje pravilno, saj 32 ne moremo deliti z 0 (nobenega števila ne moramo deliti z 0). Vhodni števili morata biti pozitivni celi števili. Algoritem bi lahko spremenili tako, da bi npr. med korak 1 in korak 2 vrinili Korak 1.2) Če je J >= (večji ali enak) 0, izpiši »Vneseno število ni pozitivno celo število. Prosimo, vnesite veljavni števili« in ponovi korak 1.



*Dobili bi torej podatek o urni postavki, ki ga bom označil z »urna\_postavka« in pa število opravljenih delovnih ur, kar bom označil z »stevilo\_ur« .*

Korak 1: placilo = 0

Korak 2: Dokler stevilo\_ur >= (večje ali enako) 40

Korak 3: Dokler stevilo\_ur >= (večje ali enako) 54

Korak 4: placilo = placilo + (urna\_postavka \* 2)

Korak 5: stevilo\_ur = stevilo\_ur – 1

Korak 6: placilo = placilo + (urna\_postavka \* 1,5)

Korak 7: stevilo\_ur = stevilo\_ur – 1

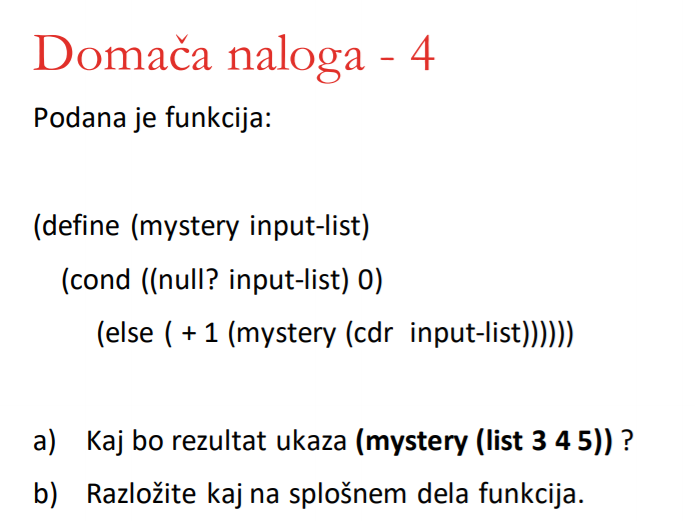
Korak 8: placilo = placilo + (urna\_postavka \* stevilo\_ur)

Korak 9: Izpiši placilo

Korak 10: Izpiši » Ali želite opraviti naslednji izračun?«

Korak 11: Če je odgovor »DA«, ponovi korake 1 do 10.

Korak 12: Ustavi se



a) Rezultat bo 3.

b) Funkcija mystery nam pove število elementov vhodnega seznama oz. dolžino seznama. Naredi torej enako kot bi npr. len([3, 4, 5]) v Pythonu.

Funkcija prejme kot argument nek seznam, nato preveri če je ta seznam prazen – če je izpiše oz. doda 0, če ni prišteje 1 in postopek ponovi za seznam brez prvega elementa.

* **Tilen Padovnik, vpisna št.:** **63190211**