

# Uvod v računalništvo vaje

26. – 30. oktober 2015

## Naloga 1

---

Podan je algoritem za zaporedno iskanje oseb v telefonskem imeniku.

- Korak 1. Preberi vrednosti za  $IME, N_1, N_2, \dots, N_{10000}$  in  $T_1, T_2, \dots, T_{10000}$ .
- Korak 2. Nastavi vrednost  $i$  na 1 in vrednost *Najden* na "NE".
- Korak 3. Dokler  $((Najden = NE) \wedge (i \leq 10000))$ , ponavljaj
- Korak 4.       Če se  $IME$  ujema z  $i$ -tim imenom v imeniku,  $N_i$
- Korak 5.       Izpiši telefonsko številko te osebe,  $T_i$ .
- Korak 6.       Nastavi vrednost *Najden* na "DA".
- sicer (t.j.  $IME$  se ne ujema z  $N_i$ )
- Korak 7.       *i* povečaj za 1.
- Korak 8.       Če ( $Najden = NE$ )
- Korak 9.       Izpiši sporočilo "Oprostite, tega naročnika ni v imeniku."
- Korak 10.      Ustavi se.

Utemelji, zakaj zgornji algoritem ne more imeti neskončne zanke. Kako lahko vidiš, da se algoritem vedno ustavi po končnem številu operacij?

## Naloga 2

---

Algoritem v nalogi 1 predpostavlja, da v imeniku ni dveh oseb z enakim imenom. Spremeni algoritem tako, da bo pravilno deloval tudi v primeru, ko se v imeniku lahko nahajajo osebe z enakimi imeni. Dopolnjeni algoritem naj najde vsako pojavitev imena  $IME$  v imeniku in izpiše ustrezne telefonske številke. Po tem, ko je algoritem izpisal telefonske številke, naj izpiše tudi, koliko pojavitev je bilo najdenih. Če se, na primer, neko  $IME$  pojavi trikrat, naj bi izhod algoritma izgledal nekako takole:

05 234 34 33

03 110 02 77

031 224 456

*Najdene so bile tri osebe.*

## Naloga 3

---

Podan je algoritem za iskanje največjega števila v seznamu.

Preberi vrednost  $n$  (velikost seznama).

Preberi vrednosti  $A_1, A_2, \dots, A_n$  (seznam števil).

Nastavi vrednost *doSedajNajvecji* na  $A_1$ .

Nastavi vrednost *polozaj* na 1.

Nastavi vrednost  $i$  na 2.

Dokler ( $i \leq n$ ), ponavljaj

    Če  $A_i > \textit{doSedajNajvecji}$

        Nastavi *doSedajNajvecji* na vrednost  $A_i$ .

        Nastavi *polozaj* na vrednost  $i$ .

$i$  povečaj za 1.

Konec zanke.

Izpiši vrednosti *doSedajNajvecji* in *polozaj*.

Ustavi se.

Zgornji algoritem je bil razvit ob predpostavki, da so vsa števila v seznamu različna. Kaj se zgodi v primeru, če vsa števila niso različna? Ali bi algoritem našel prvo pojavitev največjega števila, zadnjo pojavitev ali morda vsako pojavitev? Razloži obnašanje algoritma v danem primeru.

## Naloga 4

---

V šesti vrstici algoritma za iskanje največjega števila (naloga 3) imamo operacijo

*Dokler ( $i \leq n$ ), ponavljaj*

Razloži, kaj bi se zgodilo, če bi zgornjo operacijo spremenili v

- a) *Dokler ( $i \geq n$ ), ponavljaj*
- b) *Dokler ( $i < n$ ), ponavljaj*
- c) *Dokler ( $i = n$ ), ponavljaj*

## Naloga 5

---

V sedmi vrstici algoritma za iskanje največjega števila (naloga 3) imamo operacijo

*Če  $A_i > \textit{doSedajNajvecji}$*

Razloži, kaj bi se zgodilo, če bi zgornjo operacijo spremenili v

- a) *Če  $A_i \geq \textit{doSedajNajvecji}$*
- b) *Če  $A_i < \textit{doSedajNajvecji}$*

Kaj lahko poveš o pomembnosti uporabe pravih primerjalnih operacij ( $<$ ,  $>$ ,  $\leq$ ,  $\geq$ ,  $=$ ,  $\neq$ ) pri pisanju pogojnih in iterativnih operacij?

## Naloga 6

---

Napiši algoritem za iskanje mediane števil v seznamu, ki vsebuje  $N$  različnih števil. Pri razvoju si pomagaj z uporabo že obstoječega algoritma za iskanje največjega števila (naloga 3). Mediana  $N$  števil v seznamu je opredeljena kot število iz danega seznama, od katerega je (približno) polovica števil v seznamu manjša in (približno) polovica števil v seznamu večja. Recimo, da imamo podan seznam sedmih števil:

26, 50, 83, 44, 91, 20, 55.

Mediana tega seznama je število 50, saj so tri števila (20, 26, 44) manjša, tri števila (55, 83, 91) pa večja. V slučaju, da je  $N$  sod, naj bo število manjših števil za ena manjše od števila večjih števil. Na primer v seznamu:

26, 50, 44, 91, 20, 55

je mediana enaka 44, saj sta dve števili manjši (20, 26), tri (50, 55, 91) pa so večja.

## Naloga 7

---

Napiši algoritem za izračun tedenskega honorarnega plačila za delo. Kot vhod algoritem dobi število opravljenih delovnih ur v tednu in urno postavko. Končno plačilo naj se določi tako, da se za vse ure do dopolnjene 40. ure upošteva osnovno (podano) urno postavko, za ure od dopolnjene 40. do dopolnjene 54. količnik 1,50 urne postavke in za ure od dopolnjene 54. dalje količnik 2,00 urne postavke. Algoritem naj izračuna in izpiše višino plačila. Po izpisu plačila naj uporabnika tudi vpraša, ali želi opraviti naslednji izračun. Celoten postopek naj se ponavlja, dokler uporabnik ne odgovori z "NE".

## Naloga 8

---

Napiši algoritem, ki kot vhod prejme pozitivno celo število  $N$  in ugotovi, ali je  $N$  praštevilo (t.j. naravno število večje od 1, ki ima le dva pozitivna delitelja, število 1 in samega sebe). Kot izhod naj algoritem izpiše ustrezno sporočilo: "podano število JE praštevilo" oziroma "podano število NI praštevilo".

## Naloga 9

---

Napiši algoritem za Cezarjevo šifro, s katero se začetno besedilo spremeni v tajno besedilo, v katerem se vsaka črka začetnega besedila preslika v črko, ki je v abecedi  $k$  mest za njo. Pri tem je preslikovanje krožno (s konca abecede nadaljujemo ponovno na začetku). Če je na primer  $k = 5$ , se črka 'A' preslika v črko 'E', črka 'Z' pa v črko 'Č'. Algoritem na vhodu dobi začetno besedilo (besedilo, ki bi ga radi šifrirali), ki se konča z znakom '\$'. Na vhodu naj algoritem dobi tudi vrednost  $k$ . Algoritem naj kot izhod vrne šifrirano besedilo.

Predpostavimo, da začetno besedilo (če izvzamemo zaključni znak '\$') vsebuje le črke slovenske abecede (t. j. znake od 'A' do 'Ž'). Poleg tega predpostavimo, da je aritmetična operacija seštevanja na črkah primitivna operacija (npr.  $D + 2 = F$ ,  $D - 2 = C$ ).

## Naloga 10

---

Napiši algoritem, ki na vходу dobi celoštevilsko vrednost  $k \geq 0$  in seznam  $k$  števil  $N_1, N_2, \dots, N_k$ . Algoritem naj obrne vrstni red števil na seznamu. Če začetni seznam na primer vsebuje števila

$$N_1 = 5, N_2 = 13, N_3 = 8, N_4 = 27, N_5 = 10, \text{ (pri tem velja } k = 5),$$

naj bo vsebina seznama po koncu izvajanja algoritma naslednja:

$$N_1 = 10, N_2 = 27, N_3 = 8, N_4 = 13, N_5 = 5.$$

## Naloga 11

---

Napiši algoritem, ki kot vhod dobi seznam  $k$  celih števil  $N_1, N_2, \dots, N_k$ , prav tako pa tudi celo število VST. Algoritem mora najti dvojico števil na seznamu, ki skupaj tvorita vsoto VST. Recimo, da imamo seznam s števili 3, 8, 13, 2, 17, 18 in 10, vrednost VST pa je 20. V tem primeru naj bi algoritem na izhodu vrnil dvojico (2, 18) ali pa dvojico (3, 17). Če v seznamu ni ustrezne dvojice, naj algoritem izpiše sporočilo "V danem seznamu ni tovrstne dvojice števil".