

PROGRAMIRANJE 1

**Milena Vujošević Jančić, Jovana Kovačević,
Danijela Simić, Anđelka Zečević**

PROGRAMIRANJE 1

Zbirka zadataka

**Beograd
2016.**

Autori:

dr Milena Vujošević Jančić, docent na Matematičkom fakultetu u Beogradu

dr Jovana Kovačević, docent na Matematičkom fakultetu u Beogradu

Danijela Simić, asistent na Matematičkom fakultetu u Beogradu

Anđelka Zečević, asistent na Matematičkom fakultetu u Beogradu

PROGRAMIRANJE 1

Zbirka zadataka

Sadržaj

0.1	Petlje	v
0.2	Rešenja	xxxiv

0.1 Petlje

Zadatak 0.1.1 Napisati program koji 5 puta ispisuje tekst Mi volimo da programiramo.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.
```

[Rešenje 0.1.1]

Zadatak 0.1.2 Napisati program koji učitava ceo broj n i ispisuje n puta tekst Mi volimo da programiramo.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: 6  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj 0
```

[Rešenje 0.1.2]

Zadatak 0.1.3 Napisati program koji učitava pozitivan ceo broj n a potom ispisuje sve cele brojeve od 0 do n .

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo pozitivan broj: 4
|| 0 1 2 3 4
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo pozitivan broj: -10
|| Neispravan unos. Promenljiva mora biti
|| pozitivna!
```

[Rešenje 0.1.3]

Zadatak 0.1.4 Napisati program koji učitava dva cela broja n i m ispisuje sve cele brojeve iz intervala $[n, m]$.

- (a) Koristiti `while` petlju.
- (b) Koristiti `for` petlju.
- (c) Koristiti `do-while` petlju.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva cela broja: -2 4
|| -2 -1 0 1 2 3 4
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva cela broja: 10 6
|| Neispravan unos. Nisu dobro zadate granice
|| intervala!
```

[Rešenje 0.1.4]

Zadatak 0.1.5 Napisati program koji učitava ceo pozitivan broj i izračunava njegov faktorijel. U slučaju neispravnog unosa ispisati odgovarajuću poruku.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan broj: 18
|| Faktorijel = 640237370572800
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan broj: 8
|| Faktorijel = 40320
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan broj: 40
|| Broj je veliki, dolazi do
|| prekoracenja.
```

[Rešenje 0.1.5]

Zadatak 0.1.6 Sa standradnog ulaza unose se realan broj x i ceo pozitivan broj n . Napisati program koji izračunava n -ti stepen broja x , tj. x^n .

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite redom brojeve x i n: ||  
|| 4 3 ||  
|| 64.00000 ||
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite redom brojeve x i n: ||  
|| 5.8 5 ||  
|| 6563.56768 ||
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite redom brojeve x i n: ||  
|| 11.43 0 ||  
|| 1.00000 ||
```

[Rešenje 0.1.6]

Zadatak 0.1.7 Sa standardnog ulaza unose se realan broj x i ceo broj n . Napisati program koji izračunava n -ti stepen broja x .

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite redom brojeve x i n: 2 -3 ||  
|| 0.125 ||
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite redom brojeve x i n: -3 2 ||  
|| 9.000 ||
```

[Rešenje 0.1.7]

Zadatak 0.1.8 Pravi delioci celog broja su svi delioci sem jedinice i samog tog broja. Napisati program za uneti ceo pozitivan broj x ispisuje sve njegove prave delioce. U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite ceo broj veci od 0: 100 ||  
|| 2 4 5 10 20 25 50 ||
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite ceo broj: -6 ||  
|| neispravan unos. ||
```

[Rešenje 0.1.8]

Zadatak 0.1.9 Napisati program koji za uneti prirodan broj uklanja sve nule sa njegove desne strane. Ispisati novodobijeni broj.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite broj: 12000 ||  
|| 12 ||
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite broj: 856 ||  
|| 856 ||
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite broj: 140 ||  
|| 14 ||
```

[Rešenje 0.1.9]

Zadatak 0.1.10 Napisati program koji učitava ceo broj i ispisuje njegove cifre u obrnutom poretku.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: 6789  
|| 9 8 7 6
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: -892345  
|| 5 4 3 2 9 8
```

[Rešenje [0.1.10](#)]

Zadatak 0.1.11 Napisati program koji za uneti prirodan broj ispisuje da li je on deljiv sumom svojih cifara.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 12  
|| Deljiv je sumom svojih  
||   cifara.
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 2564  
|| Nije deljiv sumom svojih  
||   cifara.
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -4  
|| Neispravan ulaz.
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 0  
|| Neispravan ulaz.
```

[Rešenje [0.1.11](#)]

Zadatak 0.1.12 Napisati program koji učitava pozitivan ceo broj n , a zatim učitava n celih brojeva i ispisuje sumu pozitivnih i sumu negativnih unetih brojeva.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 7  
|| Unesite brojeve:  
|| 8 -50 45 2007 -67 -123 14  
|| Suma pozitivnih: 2074  
|| Suma negativnih: -240
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 5  
|| Unesite brojeve:  
|| -5 -20 -4 -200 -8  
|| Suma pozitivnih: 0  
|| Suma negativnih: -237
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -6  
|| Neispravan unos.
```

[Rešenje [0.1.12](#)]

Zadatak 0.1.13 Program unosi ceo pozitivan broj n , a potom i n celih brojeva. Izračunati i ispisati zbir onih brojeva koji su neparni i negativni.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva:
1 -5 -6 3 -11
-16
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Unesite n brojeva:
5 8 13 17
0
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -4
Neispravan unos.
```

[Rešenje [0.1.13](#)]

Zadatak 0.1.14 Program učitava ceo pozitivan broj n , a potom n celih brojeva. Naći sumu brojeva koji su deljivi sa 5, a nisu deljivi sa 7. U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite brojeve: :2 35 5 -175 -20
Suma je -15.
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -3
Neispravan unos.
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 10
Unesite brojeve:
-5 6 175 -20 -25 -8 42 245 1 6
Suma je -50.
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 6
Unesite brojeve:
2205 -1904 2 7 -540 5
Suma je -535.
```

[Rešenje [0.1.14](#)]

Zadatak 0.1.15 Nikola želi da obraduje baku i da joj kupi jedan poklon u radnji. On na raspolaganju ima m novaca. U radnji se nalazi n artikala i zanima ga koliko ima artikala u radnji čija cena je manja ili jednaka m . Napisati program koji pomaže Nikoli da brzo odrediti broj artikala. Program učitava realan pozitivan broj m , ceo pozitivan broj n i n realnih pozitivnih brojeva različitih od 0. Ispisati koliko artikala ima manju ili jednaku cenu od m . U slučaju greške ispisati odgovarajuću poruku.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj m: 12.37
Unesite broj n: 5
Unesite n brojeva: 11 54.13 6 13 8
3
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj m: 2
Unesite broj n: 4
Unesite n brojeva: 1 11 4.32 3
1
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj m: 2
|| Unesite broj n: -4
|| Broj artikala ne može biti negativan.
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj m: 30
|| Unesite broj n: 4
|| Unesite n brojeva: 67 -100 23 98
|| Cena ne može biti negativna.
```

[Rešenje [0.1.15](#)]

Zadatak 0.1.16 Napisati program koji učitava cele brojeve sve dok se ne unese nula. Nakon toga ispisati proizvod onih unetih brojeva koji su pozitivni.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve:
|| -87 12 -108 -13 56 0
|| Proizvod pozitivnih unetih
|| brojeva je 672.
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve:
|| -5 -200 -43 0
|| Nisu uneseni pozitivni
|| brojevi.
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 0
|| Nisu uneseni brojevi.
```

[Rešenje [0.1.16](#)]

Zadatak 0.1.17 Napisati program koji za pozitivan ceo broj proverava i ispisuje da li se cifra 5 nalazi u njegovom zapisu.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 1857
|| Cifra 5 se nalazi u zapisu!
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 84
|| Cifra 5 se ne nalazi u
|| zapisu!
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: -235515
|| Cifra 5 se nalazi u zapisu!
```

[Rešenje [0.1.17](#)]

Zadatak 0.1.18 Program učitava cele brojeve sve do unosa broja nula 0. Napisati program koji izračunava i ispisuje aritmetičku sredinu unetih brojeva na četiri decimale.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 8 5 6 3 0
|| Aritmeticka sredina: 5.5000
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 762 -12 800 2010 -356 899 -101
|| 0
|| Aritmeticka sredina: 571.7143
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite brojeve: 0  
|| Nisu uneseni brojevi.
```

[Rešenje 0.1.18]

Zadatak 0.1.19 U prodavnici se nalaze artikala čije cene su realni pozitivni brojevi. Program unosi cene artikala sve do unosa broja nula 0. Napisati program koji izračunava i ispisuje prosečnu vrednost cena u radnji.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cene: 8 5.2 6.11 3 0  
|| Prosečna cena je: 5.5775
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cene: 6.32 -9  
|| Cena ne može biti negativana.
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cene: 0  
|| Nisu unesene cene.
```

[Rešenje 0.1.19]

Zadatak 0.1.20 Program učitava ceo pozitivan broj n , a potom n realnih brojeva. Odrediti koliko puta je prilikom unosa došlo do promene znaka. Ispisati dobijenu vrednost.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 10  
|| Unesite brojeve:  
|| 7.82 4.3 -1.2 56.8 -3.4 -72.1 8.9 11.2 -11.2  
|| -102.4  
|| Broj promena je 5.
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 5  
|| Unesite brojeve:  
|| -23.8 -11.2 0 5.6 7.2  
|| Broj promena je 1.
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: -6  
|| Neispravan unos.
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 0  
|| Broj promena je 0.
```

[Rešenje 0.1.20]

Zadatak 0.1.21 U prodavnici se nalazi n artikala čije cene su realni brojevi. Napisati program koji učitava n , a potom i cenu svakog od n artikala i određuje i ispisuje najmanju cenu.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj artikla: 6
|| Unesite artikle:
|| 12 3.4 90 100.53 53.2 12.8
|| Minimalna cena je: 3.400000
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj artikla: 3
|| Unesite artikle: 4 -8 92
|| Cena ne moze biti negativna.
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj artikla: -9
|| Neispravan unos.
```

[Rešenje [0.1.21](#)]

Zadatak 0.1.22 Program učitava ceo pozitivan broj n , a zatim i n celih brojeva. Napisati program koji ispisuje broj sa najvećom cifrom desetica. Ukoliko ima više takvih, ispisati prvi.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| Unesite brojeve:
|| 18 365 25 1 78
|| Broj sa najvećom cifrom desetica je 78.
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 8
|| Unesite brojeve:
|| 14 1576 -1267 -89 109 122 306 918
|| Broj sa najvećom cifrom desetica je -89.
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -12
|| Neispravan unos.
```

[Rešenje [0.1.22](#)]

Zadatak 0.1.23 Program učitava ceo pozitivan broj n , a zatim i n celih brojeva. Napisati program koji ispisuje broj sa najvećim brojem cifara. Ukoliko ima više takvih, ispisati prvi.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| Unesite n brojeva: 18 365 25 1 78
|| Najviše cifara ima broj 365.
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 7
|| Unesite n brojeva:
|| 3 892 18 21 639 742 85
|| Najviše cifara ima broj 892.
```

[Rešenje [0.1.23](#)]

Zadatak 0.1.24 Program učitava ceo pozitivan broj n , a zatim i n celih brojeva. Napisati program koji ispisuje broj sa najvećom vodećom cifrom. Vodeća cifra je cifra najveće težine u zapisu broja. Ukoliko ima više takvih, ispisati prvi.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| Unesite n brojeva: 8 964 32 511 27
|| Broj sa najvećom vodećom cifrom je 964.
```

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| Unesite n brojeva: 41 669 8
|| Broj sa najvećom vodećom cifrom je 8.
```

[Rešenje 0.1.24]

Zadatak 0.1.25 Vršna su merenja nadmorskih visina na određenom delu teritorije i naučnike zanima razlika između najveće i najmanje nadmorske visine. Napisati program koji učitava realne brojeve sve do unosa 0 koji označavaju nadmorske visine i ispisuje razliku najveće i najmanje nadmorske visine.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 8 6 5 2 11 7 0
|| Razlika: 9
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 8 -1 8 6 0
|| Razlika: 9
```

[Rešenje 0.1.25]

Zadatak 0.1.26 Program učitava cele pozitivane brojeve n ($n > 1$) i d , a zatim i n celih brojeva. Napisati program koji izračunava koliko ima parova uzastopnih brojeva među unetim brojevima koji se nalaze na rastojanju d . Rastojanje između brojeva je definisano sa $d(x, y) = |y - x|$. Ispisati rezultat.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve n i d: 5 2
|| Unesite n brojeva: 2 3 5 1 -1
|| Broj parova: 2
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve n i d: 10 5
|| Unesite n brojeva:
|| -3 6 11 -20 -25 -8 42 37 1 6
|| Broj parova: 4
```

[Rešenje 0.1.26]

Zadatak 0.1.27 Napisati program koji uneti prirodan broj transformiše tako što svaku parnu cifru u zapisu broja uveća za jedan. Ispisati novodobijeni broj.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 2417  
|| 3517
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 138  
|| 139
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 59  
|| 59
```

[Rešenje [0.1.27](#)]

Zadatak 0.1.28 Napisati program koji formira i ispisuje broj koji se dobija izbacivanjem svake druge cifre polaznog celog broja, počevši od krajnje desne cifre.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 21854  
|| 284
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 18  
|| 8
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 1  
|| 1
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -67123  
|| -613
```

[Rešenje [0.1.28](#)]

* **Zadatak 0.1.29** Napisati program koji na osnovu unetog prirodnog broja formira i ispisuje broj koji se dobija izbacivanjem cifara koje su jednake zbiru svojih suseda.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 28631  
|| 2631
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 440  
|| 40
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -5  
|| Neispravan unos.
```

[Rešenje [0.1.29](#)]

* **Zadatak 0.1.30** Broj je *palindrom* ukoliko se isto čita i sa leve i sa desne strane. Napisati program koji učitava prirodan broj i proverava da li je učitani broj palindrom.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 25452  
|| Broj je palindrom!
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 895  
|| Broj nije palindrom!
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 5  
|| Broj je palindrom!
```

[Rešenje 0.1.30]

Zadatak 0.1.31 Fibonačijev niz počinje ciframa 1 i 1, a svaki član se dobija zbirom prethodna dva. Napisati program koji učitava ceo prirodan broj n i određuje i ispisuje n -ti član Fibonačijevog niza.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: 10  
|| Trazeni broj je: 55
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: -100  
|| Neispravan unos. Pozicija u Fibonacijevom  
|| nizu mora biti pozitivan broj koji nije 0!
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: 78  
|| Trazeni broj je: 375819880
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: 20  
|| Trazeni broj je: 6765
```

[Rešenje 0.1.31]

* **Zadatak 0.1.32** Niz prirodnih brojeva formira se prema sledećem pravilu:

$$a_{n+1} = \begin{cases} \frac{a_n}{2} & \text{ako je } a_n \text{ parno} \\ \frac{3 \cdot a_n + 1}{2} & \text{ako je } a_n \text{ neparno} \end{cases}$$

Napisati program koji za uneti početni član niza a_0 (ceo pozitivan broj) štampa niz brojeva sve do onog člana niza koji je jednak 1.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: 56  
|| 56 28 14 7 11 17 26 13 20 10  
|| 5 8 4 2 1
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: -48  
|| Nekorektan unos. Broj mora biti pozitivan.
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: 67  
|| 67 101 152 76 38 19 29 44 22 11  
|| 17 26 13 20 10 5 8 4 2 1
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: 33  
|| 33 50 25 38 19 29 44 22  
|| 11 17 26 13 20 10 5 8 4 2 1
```

[Rešenje 0.1.32]

* **Zadatak 0.1.33** Papir A_0 ima površinu $1m^2$ i odnos stranica $1 : \sqrt{2}$. Papir A_1 dobija se podelom papira A_0 po dužoj ivici. Papir A_2 dobija se podelom A_1 papira po dužoj ivici itd. Napisati program koji za uneti prirodan broj k ispisuje dimenzije papira A_k u milimetrima. Rezultat ispisati kao celobrojne vrednosti.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite format papira: 4  
|| 210 297
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite format papira: 3  
|| 297 420
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite format papira: 7  
|| 74 105
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite format papira: 9  
|| 37 52
```

[Rešenje [0.1.33](#)]

Zadatak 0.1.34 Napisati program koji učitava karaktere dok se ne unese karakter tačka, i ako je karakter malo slovo ispisuje odgovarajuće veliko, ako je karakter veliko slovo ispisuje odgovarajuće malo, a u suprotnom ispisuje isti karakter kao i uneti.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Danas je Veoma Lep DAN.  
|| dANAS JE vEOMA lEP dan
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| PROGRAMIRANJE 1 je zanimljivo!  
|| programiranje 1 JE ZANIMLJIVO!
```

[Rešenje [0.1.34](#)]

Zadatak 0.1.35 Napisati program koji učitava karaktere sve do kraja ulaza, a potom ispisuje broj velikih slova, broj malih slova, broj cifara, broj belina i zbir unetih cifara.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Tekst sa brojevima: 124, -8900, 23...  
|| velika: 1, mala: 15, cifre: 9, beline: 5  
|| suma cifara: 29
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| NEMA cifara!  
|| velika: 4, mala: 6, cifre: 0, beline: 1  
|| suma cifara: 0
```

[Rešenje [0.1.35](#)]

Zadatak 0.1.36 Program učitava ceo pozitivan broj n , a potom i n karaktera. Za svaki od samoglasnika ispisati koliko puta se pojavio među unetim karakterima. Ne praviti razliku između malih i velikih slova.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n karaktera: uAbao
Samoglasnik a: 2
Samoglasnik e: 0
Samoglasnik i: 0
Samoglasnik o: 1
Samoglasnik u: 1
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 7
Unesite n karaktera: jk+EEae
Samoglasnik a: 1
Samoglasnik e: 3
Samoglasnik i: 0
Samoglasnik o: 0
Samoglasnik u: 0
```

[Rešenje 0.1.36]

Zadatak 0.1.37 Program učitava ceo broj n , a zatim i n karaktera. Napisati program koji proverava da li se od unetih karaktera može napisati reč *Zima*.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Unestite 1. karakter: +
Unestite 2. karakter: o
Unestite 3. karakter: Z
Unestite 4. karakter: j
Ne može se napisati rec Zima.
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 10
Unestite 1. karakter: i
Unestite 2. karakter: 9
Unestite 3. karakter: 0
Unestite 4. karakter: p
Unestite 5. karakter: a
Unestite 6. karakter: Z
Unestite 7. karakter: o
Unestite 8. karakter: m
Unestite 9. karakter: M
Unestite 10. karakter: -
Može se napisati rec Zima.
```

[Rešenje 0.1.37]

Zadatak 0.1.38 Napisati program koji učitava ceo pozitivan broj n i ispisuje vrednost sume kubova brojeva od 1 do n , odnosno $s = 1 + 2^3 + 3^3 + \dots + n^3$. U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite pozitivan ceo broj: 14
Suma kubova od 1 do 14: 11025
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite pozitivan ceo broj: 25
Suma kubova od 1 do 25: 105625
```

[Rešenje 0.1.38]

Zadatak 0.1.39 Napisati program koji učitava ceo pozitivan broj n i ispisuje sumu kubova, $s = 1 + 2^3 + 3^3 + \dots + k^3$, za svaku vrednost $k = 1, \dots, n$. U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan ceo broj: 5
|| i=1, s=1
|| i=2, s=9
|| i=3, s=36
|| i=4, s=100
|| i=5, s=225
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan ceo broj 8
|| i=1, s=1
|| i=2, s=9
|| i=3, s=36
|| i=4, s=100
|| i=5, s=225
|| i=6, s=441
|| i=7, s=784
|| i=8, s=1296
```

[Rešenje [0.1.39](#)]

Zadatak 0.1.40 Program učitava realan broj x i ceo pozitivan broj n . Napisati program koji izračunava i ispisuje sumu $S = x + 2 \cdot x^2 + 3 \cdot x^3 + \dots + n \cdot x^n$.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 2 3
|| S=34.000000
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 1.5 5
|| S=74.343750
```

[Rešenje [0.1.40](#)]

Zadatak 0.1.41 Program učitava realan broj x i ceo pozitivan broj n . Napisati program koji izračunava i ispisuje sumu $S = 1 + \frac{1}{x} + \frac{1}{x^2} + \dots + \frac{1}{x^n}$.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 2 4
|| S=1.937500
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 1.8 6
|| S=2.213249
```

[Rešenje [0.1.41](#)]

*** Zadatak 0.1.42** Napisati program koji učitava realane brojeve x i eps i sa zadatom tačnošću eps izračunava i ispisuje sumu $S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$. Izračunati sumu u odnosu na tačnost eps znači uporediti poslednji član sume sa eps i ukoliko je taj poslednji član manji od eps prekinuti dalja izračunavanja. UPUTSTVO: Prilikom računanja sume koristiti prethodni izračunati član sume u računanju sledećeg člana sume. Naime, ako je izračunat član sume $\frac{x^n}{n!}$ na osnovu njega se lako može dobiti član $\frac{x^{n+1}}{(n+1)!}$. Nikako ne računati stepen i faktorijel odvojeno zbog neefikasnosti takvog rešenja i zbog mogućnosti prekoračenja.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x: 2
|| Unesite tacnost eps: 0.001
|| S=7.388713
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x: 3
|| Unesite tacnost eps: 0.01
|| S=20.079666
```

[Rešenje 0.1.42]

* **Zadatak 0.1.43** Napisati program koji učitava realane brojeve x i eps i sa zadatom tačnošću eps izračunava i ispisuje sumu $S = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!} \dots$. NAPOMENA: *Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.*

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x: 3
|| Unesite tacnost eps: 0.000001
|| S=0.049787
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x: 3.14
|| Unesite tacnost eps: 0.01
|| S=0.049072
```

[Rešenje 0.1.43]

Zadatak 0.1.44 Napisati program koji učitava realan broj x i prirodan broj n izračunava sumu $S = (1 + \cos(x)) \cdot (1 + \cos(x^2)) \cdot \dots \cdot (1 + \cos(x^n))$. NAPOMENA: *Voditi računa o efikasnosti rešenja.*

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 3.4 5
|| Proizvod = 0.026817
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 12 8
|| Proizvod = 2.640565
```

[Rešenje 0.1.44]

* **Zadatak 0.1.45** Napisati program koji učitava ceo prirodan broj n i ispisuje vrednost razlomka

$$1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{4 + \frac{1}{\dots + \frac{1}{(n-1) + \frac{1}{n}}}}}}$$

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite prirodan broj: 4
|| Razlomak = 0.697674
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite prirodan broj: 20
|| Razlomak = 0.697775
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite prirodan broj: 0
|| Neispravan unos.
```

[Rešenje 0.1.45]

* **Zadatak 0.1.46** Napisati program koji računa sumu

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!}.$$

za unete cele brojeve x i n . NAPOMENA: *Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.*

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 5.6 8
|| S=0.735084
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 14.32 11
|| S=17273.136719
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite prirodan broj: -6
|| Neispravan unos.
```

[Rešenje 0.1.46]

* **Zadatak 0.1.47** Program učitava ceo pozitivan broj n veći od 0. Napisati program koji računa proizvod

$$S = (1 + \frac{1}{2!})(1 + \frac{1}{3!}) \dots (1 + \frac{1}{n!}).$$

U slučaju greške pri unosu podataka ispisati odgovarajuću poruku. NAPOMENA: *Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.*

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| 1.838108
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 7
|| 1.841026
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 0
|| Neispravan unos.
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 10
|| 1.841077
```

[Rešenje 0.1.47]

* **Zadatak 0.1.48** Program učitava ceo pozitivan neparan broj n . Napisati program koji za uneto n izračunava:

$$S = 1 \cdot 3 \cdot 5 - 1 \cdot 3 \cdot 5 \cdot 7 + 1 \cdot 3 \cdot 5 \cdot 7 \cdot 9 - 1 \cdot 3 \cdot 5 \cdot 7 \cdot 9 \cdot 11 + \dots (-1)^{\frac{n-1}{2}+1} \cdot 1 \cdot 3 \cdot \dots \cdot n.$$

U slučaju greške pri unosu podataka ispisati odgovarajuću poruku. NAPOMENA: Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 9
|| 855
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 11
|| -9540
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 20
|| Neispravan unos
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -3
|| Neispravan unos.
```

[Rešenje 0.1.48]

Zadatak 0.1.49 Program učitava realne brojeve x i a i ceo pozitivan broj n veći od 0. Napisati program koji izračunava:

$$\underbrace{((\dots(((x+a)^2+a)^2+a)^2+\dots a)^2)}_n.$$

U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva relana broja x i a:: 3.2 0.2
|| Unesite prirodan broj: 5
|| Izraz = 135380494030332048.000000
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva relana broja x i a:: 2 1
|| Unesite prirodan broj: 3
|| Izraz = 10201.000000
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva relana broja x i a:: 2.6 0.3
|| Unesite prirodan broj: 3
|| Izraz = 5800.970129
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva relana broja x i a:: 5.4 7
|| Unesite prirodan broj: -2
|| Neispravan unos.
```

[Rešenje 0.1.49]

Zadatak 0.1.50 Za unetu pozitivnu celobrojnu vrednost n napisati programe koji ispisuju odgovarajuće brojeve. Pretpostaviti da je unos korektan.

- (a) Napisati program koji za unetu pozitivnu celobrojnu vrednost n ispisuje tablicu množenja.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj n: 1  
| 1
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj n: 2  
| 1 2  
| 2 4
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj n: 3  
| 1 2 3  
| 2 4 6  
| 3 6 9
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj n: 4  
| 1 2 3 4  
| 2 4 6 8  
| 3 6 9 12  
| 4 8 12 16
```

[Rešenje ??]

Napisati program koji za unetu n ispisuje sve brojeve od 1 do n^2 pri čemu se ispisuje po n brojeva u jednoj vrsti.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj n: 3  
| 1 2 3  
| 4 5 6  
| 7 8 9
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj n: 4  
| 1 2 3 4  
| 5 6 7 8  
| 9 10 11 12  
| 13 14 15 16
```

[Rešenje ??]

Napisati program koji za unetu n ispisuje tablicu brojeva tako da su u prvoj vrsti svi brojevi od 1 do n , a svaka naredna vrsta dobija se rotiranjem prethodne vrste za jedno mesto u levo.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 3  
|| 1 2 3  
|| 2 3 1  
|| 3 1 2
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 4  
|| 1 2 3 4  
|| 2 3 4 1  
|| 3 4 1 2  
|| 4 1 2 3
```

[Rešenje ??]

Napisati program koji za uneto n iscrtava pravougli „trougao” sačinjen od „koordinata” svojih tačaka. „Koordinata” tačke je oblika (i, j) pri čemu $i, j = 0, \dots, n$. Prav ugao se nalazi u gornjem levom uglu slike i njegova koordinata je $(0, 0)$. Koordinata i se uvećava po vrsti, a koordinata j po koloni, pa je zato koordinata tačke koja je ispod tačke $(0, 0)$ jednaka $(1, 0)$, a koordinata tačke koja je desno od tačke $(0, 0)$ jednaka $(0, 1)$.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 1  
|| (0,0)
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 2  
|| (0,0) (0,1)  
|| (1,0)
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 3  
|| (0,0) (0,1) (0,2)  
|| (1,0) (1,1)  
|| (2,0)
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| (0,0) (0,1) (0,2) (0,3)
|| (1,0) (1,1) (1,2)
|| (2,0) (2,1)
|| (3,0)
```

[Rešenje ??]

Zadatak 0.1.51 Napisati program koji za unet prirodan broj n zvezdicama iscrtava odgovarajuću sliku. Pretpostaviti da je unos korektan.

a) Slika sadrži kvadrat stranice n sastavljen od zvezdica.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| ***
|| ***
|| ***
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| ****
|| ****
|| ****
|| ****
```

[Rešenje ??]

b) Slika sadrži rub kvadrata dimenzije n .

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| *****
|| *   *
|| *   *
|| *   *
|| *****
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 2
|| **
|| **
```

[Rešenje ??]

- c) Slika sadrži rub kvadrata dimenzije n koji i na glavnoj dijagonali ima zvezdice.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
*****
** *
* * *
* **
*****
```

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
****
** *
* **
****
```

[Rešenje ??]

* **Zadatak 0.1.52** Napisati program koji za uneti prirodan broj n zvezdicama iscrtava slovo X dimenzije n . Pretpostaviti da je unos korektan.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
*  *
*  *
*
*  *
*  *
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
* *
*
* *
```

[Rešenje [0.1.52](#)]

* **Zadatak 0.1.53** Napisati program koji za uneti prirodan neparan broj n korišćenjem znaka $+$ iscrtava veliko $+$ dimenzije n . Pretpostaviti da je unet prirodan broj.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
+
+
+++++
+
+
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
+
+++
+
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| Pogresan unos.
```

[Rešenje 0.1.53]

Zadatak 0.1.54 Napisati program koji učitava prirodan broj n , a potom iscrtava odgovarajuću sliku. Pretpostaviti da je unos korektan.

- (a) Slika sadrži pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine n , a prav ugao se nalazi u gornjem levom uglu slike.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| ***
|| **
|| *
```

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| ****
|| ***
|| **
|| *
```

[Rešenje ??]

Slika sadrži pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine n , a prav ugao se nalazi u donjem levom uglu slike.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| *
|| **
|| ***
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| *
|| **
|| ***
|| ****
```

[Rešenje ??]

Slika sadrži pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine n , a prav ugao se nalazi u gornjem desnom uglu slike.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| ***
|| **
|| *
```

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| ****
|| ***
|| **
|| *
```

[Rešenje ??]

Slika sadrži pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine n , a prav ugao se nalazi u donjem desnom uglu slike.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| *
|| **
|| ***
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| *
|| **
|| ***
|| ****
```

[Rešenje ??]

Slika sadrži trougao sastavljen od zvezdica. Trougao se dobija spajanjem dva pravougla trougla čija kateta je dužine n , pri čemu je prav ugao prvog trougla u njegovom donjem levom uglu, dok je prav ugao drugog trougla u njegovom gornjem levom uglu, a spajanje se vrši po horizontalnoj kateti.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
*
**
***
**
*
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
*
**
***
****
***
**
*
```

[Rešenje ??]

Slika sadrži rub jednakokrakog pravouglog trougla čije su katete dužine n .
Program učitava karakter c i taj karakter koristi za iscrtavanje ruba trougla.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Unesite karakter c: *
*
**
* *
****
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite karakter c: +
+
++
+ +
+ +
+++++
```

[Rešenje ??]

Zadatak 0.1.55 Napisati program koji učitava ceo broj n , a potom iscrtava odgovarajuću sliku.

- (a) Slika sadrži jednakokranični trougao stranice n koji je sastavljen od zvezdica.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
| Unesite broj n: 3
|
| *
| ***
| *****
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
| Unesite broj n: 4
|
| *
| ***
| *****
| *****
```

[Rešenje ??]

Slika sadrži jednakokranični trougao stranice n koji je sastavljen od zvezdica pri čemu je vrh trougla na dnu slike.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
| Unesite broj n: 3
|
| *****
| ***
| *
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
| Unesite broj n: 4
|
| *****
| *****
| ***
| *
```

[Rešenje ??]

Slika sadrži trougao koji se dobija spajanjem dva jednakokranični trougla stranice n koji su sastavljeni od zvezdica.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
| Unesite broj n: 3
|
| *
| **
| ***
| ****
| ***
| **
| *
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
      *
     ***
    *****
   *       *
  *       *
 *       *
*       *
 *       *
  *       *
   *       *
    *****
     ***
      *
```

[Rešenje ??]

Slika sadrži rub jednakostraničnog trougla čija stranica je dužine n .

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
      *
     * *
    * * *
```

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
      *
     * *
    * * *
   *   *
  *   *
 *   *
*   *
 *   *
  *   *
   *   *
    * * *
     * *
      *
```

[Rešenje ??]

Slika se dobija spajanjem dva jednakostranična trougla čija stranica je dužine n . Iscrtavati samo rub trouglova.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
      *
     * *
    * * *
   *   *
  *   *
 *   *
*   *
 *   *
  *   *
   *   *
    * * *
     * *
      *
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5

  *
 * *
*   *
* * * *
 *   *
  * *
   *
```

[Rešenje ??]

* **Zadatak 0.1.56** Napisati program koji za uneti prirodan broj n iscrtava strelice dimenzije n . Pretpostaviti da je unos korektan.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3

*
 *
***
 *
*
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5

*
 *
  *
   *
    *****
   *
  *
 *
*
```

[Rešenje [0.1.56](#)]

* **Zadatak 0.1.57** Napisati program koji učitava ceo broj n , i iscrtava sliku koja se dobija na sledeći način: u prvom redu je jedna zvezdica, u drugom redu su dve zvezdice razdvojene razmakom, treći red je sastavljen od zvezdica i iste je dužine kao i drugi red, četvrti red se sastoji od tri zvezdice razdvojene razmakom, a peti red je sastavljen od zvezdica i iste je dužine kao i četvrti red itd. Ukupna visina slike je n . Pretpostaviti da je unos korektan.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 7
*
* *
***
* * *
*****
* * * *
*****
```

[Rešenje 0.1.57]

* **Zadatak 0.1.58** Program učitava prirodne brojeve m i n . Napisati program koji iscrtava jedan do drugog stranice n kvadrata čija je svaka strana sastavljena od m zvezdica razdvojenih prazninom. Podrazumevati da je unos korektan.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
  Unesite broj n: 5 3
  * * * * *
  *       *       *
  *     *       *
  *       *       *
  *     *       *
  * * * * *
```

Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4 4
* * * * *
*       *       *       *       *
*       *       *       *       *
* * * * *

```

[Rešenje 0.1.58]

* **Zadatak 0.1.59** Program učitava prirodan broj n . Napisati program koji štampa romb sastavljen od minusa u pravougaoniku sastavljenom od zvezdica. Podrazumevati da je unos korektan.

Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 6

*****
*****--*****
*****-----*****
***-----***
**-----**
*-----*
*-----*
**-----**
***-----***
*****-----*****
*****--*****
*****

```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 2
****
*--*
****
```

[Rešenje 0.1.59]

Zadatak 0.1.60 Napisati program koji učitava ceo broj n ($n \geq 2$) i koji iscrtava sliku kuće sa krovom: kuća je kocka stranice n , a krov jednakostranični trougao stranice n . Pretpostaviti da je unos korektan.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
||
||   *
||  * *
|| *   *
||*   * *
||* * * *
||*   * *
||*   * *
||* * * *
```

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
||
||   *
||  * *
|| * * *
||*   *
||* * *
```

[Rešenje [0.1.60](#)]

* **Zadatak 0.1.61** Program učitava ceo pozitivan broj n . Napisati program koji ispisuje brojeve od 1 do n , zatim od 2 do $n - 1$, 3 do $n - 2$, itd. Ispis se završava kada nije moguće ispisati ni jedan broj. Za neispravan unos, program ispisuje odgovarajuću poruku. Pretpostaviti da je unos korektan.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| 1 2 3 4 5 2 3 4 3
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -4
|| -1
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 7
|| 1 2 3 4 5 6 7 2 3 4 5 6 3 4 5 4
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| 1 2 3 2
```

[Rešenje [0.1.61](#)]

* **Zadatak 0.1.62** Napisati program koji učitava ceo pozitivan broj n i ispisuje sve brojeve od 1 do n , zatim svaki drugi broj od 1 do n , zatim svaki treći broj od 1 do n itd., završavajući sa svakim n -tim (tj. samo sa 1). U slučaju greške pri unosu podataka odštampati odgovarajuću poruku.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
1 2 3
1 3
1
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 7
1 2 3 4 5 6 7
1 3 5 7
1 4 7
1 5
1 6
1 7
1
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 1
1
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -23
Neispravan unos.
```

[Rešenje [0.1.62](#)]

0.2 Rešenja

Rešenje 0.1.1

```
#include <stdio.h>

2
int main()
4 {
    /* Promenljiva i kontrolise koliko puta ce se petlja izvorsiti i
       naziva se brojac petlje.
       Njenu pocetnu vrednost postavljamo na 0 jer se u pocetku petlja
       nije ni jednom izvorsila. */
6     int i = 0;

8     /* Pre ulaska u telo petlje proverava se da li je
       ispunjen uslov ulaska u petlju.
       */
10    while(i < 5)
    {
12        /* Ukoliko uslov ulaska u petlju jeste ispunjen, izvorsava se telo
           petlje. */
           /* Ispisujemo trazeni tekst. */
14        printf("Mi volimo da programiramo.\n");

16        /* Uvecava se brojac za jedan jer je jednom izvorseno telo petlje.
           Ako bi ova vrednost ostala nepromenjena, petlja bi se
           izvorsavala beskonacno. */
18        i++;
20    }
```

```

22     /* Nakon poslednje naredbe tela petlje ponovo se ispituje uslov
    petlje.*/
24 }
26 return 0;
}

```

Rešenje 0.1.2

```

#include<stdio.h>
2
int main()
4 {
    /* Brojac u petlji. */
6     int i=0;
    /* Promenljiva koja oznacava koliko puta cemo ispisati trazeni
    tekst. */
8     int n;

10     printf("Unesite ceo broj: ");
    scanf("%d", &n);

12     /* Pre ulaska u telo petlje proverava se da li je ispunjen uslov
    ulaska u petlju. */
14     while (i<n)
    {
16         printf("Mi volimo da programiramo.\n");
        i++;
18     }
    return 0;
20 }

```

Rešenje 0.1.3

```

1  #include <stdio.h>

3  int main()
    {
5      /* Promenljivu x koristimo u dve svrhe. Prvo, ova promenljiva
        kontroliše koliko puta se petlja izvršila.
        Drugo, ovu promenljivu koristimo za ispis potrebnih vrednosti.
        */
7      int x;
        /* Promenljiva n se unosi i određuje koliko brojeva ispisujemo.
        */
9      int n;

11     printf("Unesi pozitivan ceo broj: ");

```

```

13     scanf("%d", &n);

15     /* U slucaju neispravnih podataka ispisujemo odgovarajucu poruku
16        i izlazimo iz programa. */
17     if (n < 0)
18     {
19         printf("Neispravan unos. Promenljiva mora biti pozitivna!\n");
20         return -1;
21     }

22     /* Ispis pocinjemo od 0, zato promenljivu x postavljamo na 0. */
23     x=0;
24     while (x<= n)
25     {
26         /* Ispisujemo broj. */
27         printf("%d\n", x);
28         /* Uvecavamo promenljivu za jedan jer smo broj ispisali i sada
29            zelimo da ispisemo sledeci broj. */
30         x++;
31     }
32     return 0;
33 }

```

Rešenje 0.1.4

```

1  /* Resenje pod a). */

3  #include <stdio.h>

5  int main()
6  {
7      /* Promenljive koje oznacavaju granice intervala. */
8      int n,m;
9      /* Promenljiva koja oznacava trenutno ispisani broj intervala. */
10     int i;

11     printf("Unesi dva cela broja: ");
12     scanf("%d%d",&n,&m);

13     if (m < n)
14     {
15         printf("Neispravan unos. Nisu dobro zadate granice intervala!\n");
16     };
17     return -1;
18 }

19

20

21 /* Na pocetku ispisujemo prvi broj intervala, a to je n. */
22 i=n;
23 /* uslov petlje se proverava pre ulaska u telo petlje */
24 while (i<=m)
25 {

```

```

    printf("%d ", i);
27     i++;
    }
29
    printf("\n");
31
33     return 0;
}

```

```

/* Resenje pod b). */
2
#include <stdio.h>
4
int main()
6 {
8     /* Promenljive koje oznacavaju granice intervala. */
    int n,m;
10    /* Promenljiva koja oznacava trenutno ispisani broj intervala. */
    int i;
12
    printf("Unesi dva cela broja: ");
14    scanf("%d%d",&n,&m);
16
    if (m < n)
    {
18        printf("Neispravan unos. Nisu dobro zadate granice intervala!\n"
        );
        return -1;
20    }
22
    /* naredba i=n se izvorsava jednom, pre prve
    iteracije */
24    for(i=n;i<=m;i++) /* uslov petlje i<=m se proverava pre svake
    iteracije */
        printf("%d ", i); /* naredba i++ se izvorsava nakon svake
    iteracije */
26
    printf("\n");
28
30    return 0;
}

```

```

1 /* Resenje pod c). */
3
#include <stdio.h>
5
int main()

```

```

{
7
    /* Promenljive koje oznacavaju granice intervala. */
9    int n,m;
    /* Promenljiva koja oznacava trenutno ispisani broj intervala. */
11   int i;

13   printf("Unesi dva cela broja: ");
   scanf("%d%d",&n,&m);

15   if (m < n)
17   {
       printf("Neispravan unos. Nisu dobro zadate granice intervala!\n"
   );
19       return -1;
   }

21

23   /* Uslov petlje se proverava na kraju svake iteracije. */
25   /* Zbog toga se do while petlja izvršava bar jednom, čak i u
       slučaju
       da uslov petlje nikada nije ispunjen. */
27   i=n;
   do
29   {
       /* Petlja se zapocinje bez provere uslova. */
       printf("%d ",i); /* Stampa se vrednost promenljive i. */
31       i++;           /* Uvecava se vrednost promenljive i. */
   }
33   while(i<=m);        /* Proverava se uslov i ukoliko je ispunjen,
       nastavlja se sa sledecom iteracijom. */
       /* U suprotnom, petlja se završava i program
       se nastavlja od prve naredbe koja sledi za petljom. */
35   printf("\n");

37   return 0;
}

```

Rešenje 0.1.5

```

1  #include<stdio.h>

3  int main()
   {
5       int x;
       /* U promenljivoj f se pamti izracunati faktorijel.
7       * Kako faktorijel je jako veliki broj, za tip podataka
       * se uzima unsigned long, da bi mogla da se upise
9       * sto veca vrednost. */
       unsigned long f;
11      int i;

```

```

13     int original;

15     printf("Unesite pozitivan broj: ");
16     scanf("%d",&x);

17     if (x < 0)
18     {
19         printf("Nekorektan unos\n");
20         return -1;
21     }

23     if (x >= 22)
24     {
25         printf("Broj je veliki, dolazi do prekoracenja.\n");
26         return -1;
27     }

29     original=x;
30     f=1;

31     while (x>1)
32     {
33         f=f*x;
34         x--;
35     }

37     printf("Faktorijel = %lu\n", f);

39

41     return 0;
43 }

```

Rešenje 0.1.6

```

1  #include <stdio.h>

3  int main(){

5      int n;
6      float x;
7      float vrednost;
8      unsigned exp;

9

11     printf("Unesite redom brojeve x i n: ");
12     scanf("%f %d", &x, &n );

13     if (n < 0)
14     {
15         printf("Neispravan unos.\n");
16         return -1;

```

```

17     }

19     /* Pocetna vrednost stepena koji se racuna. */
    vrednost=1;

21     for(exp=1; exp<=n; exp++)
23         vrednost=vrednost*x;

25     printf("%f\n",vrednost);

27     return 0;
}

```

Rešenje 0.1.7

```

1  #include <stdio.h>

3  int main(void){

5      int n, n_abs;
       float x;
       float vrednost;
       unsigned exp;

9      printf("Unesite redom brojeve x i n: ");
11     scanf("%f %d", &x, &n );

13     /* Pocetna vrednost stepena koji se racuna. */
    vrednost=1;

15     /* Stepenovanje. */
    n_abs=abs(n);
17     for(exp=1; exp<=n_abs; exp++)
19         vrednost=vrednost*x;

21     /* Ukoliko je stepen bio negativan treba
       * odrediti 1/x^n, sto je zapravo 1/vrednost. */
23     if(n<0){
25         printf("%.3f\n",1/vrednost);
27     }
       else{
29         printf("%.3f\n",vrednost);
31     }

    return 0;
}

```

Rešenje 0.1.8


```

1  #include<stdio.h>
   #include<math.h>
3
4  int main()
5  {
6      int x;
7      /* Brojac u petlji. */
8      int i;
9
10     /* Ucitavamo broj. */
11     printf("Unesi ceo broj veci od 0: ");
12     scanf("%d", &x);
13
14     if (x<=0)
15     {
16         printf("Neispravan unos.\n");
17         return -1;
18     }
19
20     /* 1. nacin */
21     printf("----- 1. nacin ----- \n");
22     for(i=2;i<x;i++)
23     {
24         /* Proverava se da li i deli broj x i
25          * ako je to slucaj ispusje se i.
26          */
27         if (x%i==0)
28             printf("%d \n",i);
29     }
30
31     /* 2. nacin (brzi) -- Ne proveravaju se svi brojevi od 2 do x, vec
32      * se petlja izvrsava dok ne stignemo do korena broja.
33      */
34     printf("----- 2. nacin ----- \n");
35     for(i=2;i<=sqrt(x);i++)
36     {
37         /* Proveravamo da li i deli broj x. */
38         if (x%i==0)
39             /* U slucaju kada je delilac koren broja, npr. 4 za 16,
40              ispisujemo ga jednom. */
41             if (i==x/i)
42                 printf("%d \n",i);
43             /* U suprotnom, npr. 2 za 16, ispisujemo i 2 i 8. */
44             else
45                 printf("%d %d \n",i,x/i);
46     }
47     return 0;
48 }

```

Rešenje 0.1.9

```

1  #include <stdio.h>

3  int main(){
    int n;

5      /* Ucitavamo broj */
7      printf("Unesite broj: ");
      scanf("%d", &n);

9      if(n==0){
11         printf("0\n");
      }
13     else{
        /* Sve dok je poslednja cifra u zapisu broja n nula */
15         while(n%10==0){
            /* Broj delimo sa 10 tj. uklanjamo mu nulu sa kraja */
17             n=n/10;
        }

19         /* Ispisujemo rezultat */
21         printf("%d\n", n);
    }

23

25     return 0;
}

```

Rešenje 0.1.10

```

1  #include<stdio.h>
   #include<stdlib.h>

3

5  int main()
   {
7      int x;
      char cifra;
      printf("Unesi ceo broj:");
9      scanf("%d", &x);

11     /* Pretvaranje u apsolutnu vrednost se vrsi za slucaj kada je unet
       * negativan broj kako bismo osigurali da ce nam izdvojene cifre
13     * biti pozitivne.
       */
15     x = abs(x);

17     /* Kako uklanjamo cifre broja (pogledati telo petlje)
       * u nekom trenutku broj ce postati 0 jer smo uklonili
19     * sve njegove cifre. Tada prekidamo rad petlje.
       */
21     while(x>0)

```

```

23     {
        /* Izdvajamo poslednju cifru broja x. */
        cifra=x%10;
25         printf("%d\n", cifra);
        /* Uklanjammo poslednju cifru broja x. */
27         x/=10;
    }
29
    return 0;
31 }

```

Rešenje 0.1.11

```

1  #include <stdio.h>
3  int main()
4  {
5      /* Prirodni broj koji se unosi. */
        int n;
7      /* Promenljiva u koju se smesta suma cifara broja. */
        int suma = 0;
9      /* Pomocna promenljiva u koju se smesta unesen broj. */
        int pom_n;
11
        printf("Unesi broj ");
13         scanf("%d", &n);

15         /* U zadatku pise da se unosi prirodan broj,
            * sto znaci da treba proveriti da li je veci
17             * od 0
            */
19
        if (n <= 0)
21         {
            printf("Neispravan unos.\n");
23             return -1;
        }

25
        /* Potrebno je koristiti pomocnu promenljivu jer u telu petlje
27         * se odstranjuju cifre broja i na taj nacin uneseni broj
            * se menja. Nakon rada petlje potrebno je ponovo koristiti
29         * uneseni broj, a to znaci da treba sacuvati neizmenjen
            * broj.
31         */
        pom_n = n;
33
        while(pom_n != 0)
35         {
            /* Na sumu dodajemo poslednju cifru. */
37             suma += pom_n % 10;
            /* Sa broja skidamo poslednju cifru. */

```

```

39     pom_n /= 10;
40 }
41
42 if (n % suma == 0)
43     printf("Deljiv je sumom svojih cifara.\n");
44 else
45     printf("Nije deljiv sumom svojih cifara.\n");
46
47 return 0;
48 }

```

Rešenje 0.1.12

```

1  #include<stdio.h>
2
3  int main()
4  {
5      int n;
6      /* Oznaka broja koji unosimo u jednoj iteraciji petlje. */
7      int x;
8      int suma_poz;
9      int suma_neg;
10     /* Brojac. */
11     int i;
12
13     printf("Unesi pozitivan ceo broj:");
14     scanf("%d",&n);
15
16     if (n < 0)
17     {
18         printf("Neispravan unos.\n");
19         return -1;
20     }
21
22     /* Promenljivama koje ce sadrzati sume se pre ulaska u petlju
23      * dodeljuje 0 (neutral za sabiranje).
24      */
25     suma_poz=0;
26     suma_neg=0;
27     i=0;
28
29     printf("Unesite brojeve: ");
30     while(i<n)
31     {
32         scanf("%d", &x);
33
34         if (x<0)
35             suma_neg+=x;
36         else
37             suma_poz+=x;
38     }
39 }

```

```

39     i++;
40 }
41
42 printf(" Suma pozitivnih: %d\n Suma negativnih: %d\n", suma_poz,
43        suma_neg);
44 return 0;
45 }

```

Rešenje 0.1.13

```

1  #include <stdio.h>
2
3  int main()
4  {
5      /* Promenljiva x oznacava tekuci uneti broj. */
6      int n, x;
7      /* Brojac. */
8      int i;
9      int zbir=0;
10
11     printf("Unesite broj n: ");
12     scanf("%d", &n);
13
14     if (n < 0)
15     {
16         printf("Neispravan unos.\n");
17         return -1;
18     }
19
20     printf("Unesite n brojeva: ");
21
22     /* Inicijalizuje se brojac sa kojim se kontrolise broj
23      * učitavanja - treba da ih bude tacno n.
24      */
25     i=0;
26     while(i<n){
27         /* Ucitava se broj. */
28         scanf("%d", &x);
29
30         /* Proverava se da li broj negativan i neparan. */
31         if(x<0 && x%2!=0){
32             /* Ako jeste, dodajemo ga na zbir. */
33             zbir=zbir+x;
34         }
35
36         /* Uvecava se brojac iteracija. */
37         i++;
38     }
39
40     /* Ispisuje se rezultat. */
41     printf("%d\n", zbir);

```

```
43     return 0;
}
```

Rešenje 0.1.14

```
1  #include <stdio.h>
3  int main()
4  {
5      int n, broj;
6      int suma = 0;
7      /* Brojac. */
8      int i;
9
10     printf("Unesite broj n: ");
11     scanf("%d", &n);
12
13     if (n < 0)
14     {
15         printf("Neispravan unos.\n");
16         return -1;
17     }
18
19     printf("Unesite brojeve: ");
20     for(i=0; i<n; i++)
21     {
22         scanf("%d", &broj);
23
24         if (broj % 5 == 0 && broj % 7 != 0)
25             suma += broj;
26     }
27
28     printf("Suma je %d.\n", suma);
29
30     return 0;
31 }
```

Rešenje 0.1.15

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Promenljiva cena oznacava trenutno unesenu cenu. */
6      float cena;
7      float m;
8      int n, i;
9      int broj_brojeva=0;
```

```

11 printf("Unesite koliko novaca ima Nikola: ");
   scanf("%f", &m);
13
   if (m < 0)
15   {
       printf("Nikola ne moze imati negativno novaca.\n");
17       return -1;
   }
19
   printf("Unesite broj artikala: ");
21   scanf("%d", &n);

   if (n < 0)
23   {
       printf("Broj artikala ne moze biti negativan.\n");
25       return -1;
   }
27

   printf("Unesite cene artikala: ");

   i=0;
   while(i<n){
31       /* Ucitava se cena artikla. */
33       scanf("%f", &cena);
35

       if (cena <= 0)
37       {
           printf("Cena ne moze biti negativna.\n");
39           return -1;
       }

           /* Provera da li je cena manji od zadatog broja m. */
43       if(cena < m){
           /* Ako jeste, uvecava se brojac brojeva za 1. */
45           broj_brojeva++;
       }

           i++;
47
       }
49

   printf("%d\n", broj_brojeva);

51   return 0;
53 }

```

Rešenje 0.1.16

```

1 #include <stdio.h>
   int main()
3 {

```

```

5  int x;
   /* U promenljivoj p se cuva proizvod. */
   int p;
7  /* Promenljiva u služi za proveru da li
   * su brojevi uopšte uneseni. Na početku se
9  * pretpostavlja da nisu i postavlja se na 0.
   */
11 int u = 0;

13 /* Promenljiva unesen_pozitivan služi za proveru da li
   * su pozitivni brojevi uopšte uneseni. Na početku se
15 * pretpostavlja da nisu i postavlja se na 0.
   */
17 int unesen_pozitivan = 0;

19 p=1;
   /* Izraz 1 je konstantan, različit je od nule
21  * što znači da je to tačan izraz. Uslov petlje je
   * uvek tačan!
23  */
   printf("Unesite brojeve:");
25 while (1)
   {
27     scanf("%d", &x);
       /* Proveravanje da li je uneta nula. */
29     if (x==0)
       /* Naredba break prekida petlju. Izvršavanje
31      * se nastavlja od prve naredbe nakon petlje.
       */
33         break;

35     /* Ako je makar 1 broj različit od 0 promenljiva u
       * će biti postavljena na 1.
37     */
       u = 1;
39     /* Ako je unet negativan broj, taj broj se ne množi
       * sa ukupnim proizvodom p; zato se nastavlja dalje.
41     */
       if (x<0)
43         /* Naredba continue prekida trenutnu iteraciju petlje
           * tako što preskače sve naredbe koje nakon njega slede.
45         * Izvršavanje se nastavlja od provere uslova petlje.
           */
47         continue;

49     /* Ako je makar jedan broj pozitivan, promenljiva
       * unesen_pozitivan se postavlja na 1. */
51     unesen_pozitivan = 1;
       p=p*x;
53 }

55 if (u == 0)

```



```

    printf("Nisu uneseni brojevi.\n");
57 else if (unesen_pozitivan == 0)
    printf("Nisu uneseni pozitivni brojevi. \n");
59 else
    printf("Proizvod pozitivnih unetih brojevi je %d.\n",p);
61
    return 0;
63 }

```

Rešenje 0.1.17

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   int main(){
5       int n, cifra;
       int indikator=0;
7
       /* Ucitavamo broj. */
9       printf("Unesite broj: ");
       scanf("%d", &n);
11
       if (n < 0)
13         n = abs(n);
15
       /* Sve dok imamo cifara u zapisu broja. */
       while(n>0){
17
           /* Izdvajamo poslednju cifru broja. */
19           cifra=n%10;
21
           /* Proveravamo da li je bas ona jednaka broju 5 */
           if(cifra==5){
23               /* Ako jeste postavljamo indikator na vrednost 1 tako da
                znamo
                 * da smo pronasli peticu i prekidamo sa izvršavanjem petlje.
                */
25               indikator=1;
27               break;
           }
29           /* Ako izdvojena cifra nije jednaka broju 5, broj delimo sa 10
            * kako bi mogli da izdvojimo i preostale cifre broja na isti
            * nacin.
            */
31           n=n/10;
33       }
35
       /* Ispisujemo rezultat */
37       if(indikator==0){
           printf("Cifra 5 se ne nalazi u zapisu!\n");
39       }

```

```

41     else{
42         printf("Cifra 5 se nalazi u zapisu!\n");
43     }
44     return 0;
45 }

```

Rešenje 0.1.18

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main(){
5
6      int x;
7      int broj_brojeva;
8      int suma;
9
10     broj_brojeva=0;
11     suma=0;
12
13     printf("Unesite brojeve: ");
14
15     while(1){
16         /* Ucitavanje broja. */
17         scanf("%d", &x);
18
19         /* Ako je unesena 0, prekida se petlja. */
20         if(x==0)
21             break;
22
23         /* Procitani broj dodaje se na sumu. */
24         suma+=x;
25
26         /* I uvecava se broj ucitanih brojeva. */
27         broj_brojeva++;
28     }
29
30     if (broj_brojeva == 0)
31         printf("Nisu uneseni brojevi.\n");
32     else
33         /* Prilikom deljenja celih brojeva kao rezultat se dobija ceo broj.
34          * Kako je aritmeticka sredina realan broj, potrebno je izvesti
35          * konverziju prilikom deljenja da bi se dobio ispravan rezultat.
36          */
37         printf("Aritmeticka sredina: %.4f\n", (double)suma/broj_brojeva);
38
39     return 0;
40 }

```

```
}
```

Rešenje 0.1.19

```
1  #include <stdio.h>
   #include <math.h>
3
4  int main(){
5
6      float cena;
7      int broj_artikla;
8      float suma;
9
10     broj_artikla=0;
11     suma=0;
12
13     printf("Unesite cene: ");
14
15     while(1){
16         scanf("%f", &cena);
17
18         if(cena == 0)
19             break;
20
21         if (cena < 0)
22         {
23             printf("Cena ne moze biti negativna.\n");
24             return -1;
25         }
26
27         suma+=cena;
28
29         /* I uvecava se broj ucitanih brojeva. */
30         broj_artikla++;
31     }
32
33     if (broj_artikla == 0)
34         printf("Nisu unesene cene.\n");
35     else
36         printf("Aritmeticka sredina: %.4f\n", suma/broj_brojeva);
37
38     return 0;
39 }
```

Rešenje 0.1.20

```
1  #include <stdio.h>
```

```

3 int main()
4 {
5     int n;
6     /* Ucitavaju se dva broja, broj i sledbenik, i proverava se da li
7     su
8     * razlicitog znaka. */
9     double broj, sledbenik;
10    /* Brojac. */
11    int i;
12    int broj_promena = 0;
13
14    printf("Unesite broj n ");
15    scanf("%d", &n);
16
17    if (n < 0)
18    {
19        printf("Neispravan unos.\n");
20        return -1;
21    }
22
23    /* Prvo se proveara da li uopste ima unosa, i ako
24    * unosa nema, ispisuje se odgovarajuca poruka i
25    * izlazi iz programa. */
26    if (n == 0)
27    {
28        printf("Broj promena je 0.\n");
29        return 0;
30    }
31
32    printf("Unesite brojeve: ");
33    /* Pre petlje ucitava se jedan broj, a u petlji
34    * se ucitava njegov sledbenik i proverava se
35    * da li su razlicitog znaka. */
36    scanf("%lf", &broj);
37
38    /* Kako je vec jedan broj unesen, brojac se postavlja
39    * na 1, a ne na 0. */
40    for(i=1; i<n; i++)
41    {
42        /* Ucitava se sledbenik. */
43        scanf("%lf", &sledbenik);
44
45        /* Ako su razlicitog znaka proizvod je manji od 0. */
46        if (sledbenik * broj < 0)
47            broj_promena++;
48        /* Problem je ako je proizvod jednak 0. Tada mora
49        * proveriti da li je jedan od brojeva negativan jer tada
50        * postoji promena znaka. */
51        else if (sledbenik*broj==0 && (sledbenik < 0 || broj < 0))
52            broj_promena++;
53
54        /* Tekuci sledbenik postaje tekuci broj, a

```

```

55     * u sledecoj iteraciji petlje se ucitava
    * sledeci sledbenik. */
    broj = sledbenik;
57 }

59 printf("Broj promena je %d.\n", broj_promena);

61 return 0;
}

```

Rešenje 0.1.21

```

1  #include <stdio.h>
   int main()
3  {
   /* Broj artikala. */
5   int n;
   /* Brojac. */
7   int i;
   /* Cena trenutno unetnog artikla. */
9   float cena;
   /* Minimalna cena. */
11  float min_cena;

13  printf("Unesite broj artikala:");
   scanf("%d", &n);

15
   if (n<=0)
17   {
       printf("Neispravan unos\n");
19       return -1;
   }

21

23  /* Prva cena se unosi iznad petlje
   * kako bi bio njegova vrednost bila dodeljena promenljivoj
   min_cena.
   * Neophodno je da promenljiva min bude inicijalizovana pre
   ulaska u petlju
   * da bi uslov x<min mogao da bude ispitan u prvoj iteraciji.
   */
27  printf("Unesite cenu artikala:");
29  scanf("%f", &cena);
   /* Proveravamo da li je cena isprano uneta vrednost. */
31  if (cena <= 0)
   {
33  printf("Cena ne moze biti negativna.\n");
   return -1;
35  }

37  min_cena = cena;

```

```

39     i=0;
41     while(i< n-1)
42     {
43         scanf("%f", &cena);

45         if (cena <= 0)
46         {
47             printf("Cena ne moze biti negativna.\n");
48             return -1;
49         }

51         /* Provera da li je uneta cena manja od tekuće minimalne cene.
52         */
53         if(cena<min_cena)
54             min_cena = cena;
55         i++;
56     }

57     printf("Minimalna cena je: %f\n", min_cena);
58     return 0;
59 }

```

Rešenje 0.1.22

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5
6      int n;
7      int x, x_desetica;
8      int max_desetica, broj;
9      int i;
10
11     printf("Unesite broj n: ");
12     scanf("%d", &n);
13
14     if (n < 0)
15     {
16         printf("Neispravan unos.\n");
17         return -1;
18     }
19
20     if (n == 0)
21     {
22         printf("Nisu uneseni brojevi.\n");
23         return 0;
24     }
25

```

```

27  /* Maksimalna cifra desetice se postavlja na 0 jer
    * 0 je svakako najmanja cifra pa je pocetna vrednost
    neutralna tj.
    * ne moze da utice na izracunavanje maksimuma. Ipak, treba
    biti pazljiv
29  * jer nije uvek zgodno pretpostaviti da je maksimalna
    vrednost 0.
    * Na primer, ako je zadatak naci maksimum celih brojeva, a
    korisnik
31  * unese -32 -7 i -22, maksimalni je broj -7, sto je manje od
    0.
    */
33  max_desetica=0;

35  printf("Unesite brojeve: ");
    for(i=0; i<n; i++){
37      scanf("%d", &x);

39      /* Izdvajanje cifre desetice procitanog broja. */
        x_desetica=(abs(x)/10)%10;

41      /* Proverava da li je izdvojena cifra veca od
        * trenutne maksimalne cifre desetice. */
43      if(x_desetica>max_desetica)
45      {
47          /* Ako jeste vece, pamti se nova najveća cifra,
            * kao i broj u kom se pojavila. */
            max_desetica=x_desetica;
            broj=x;
49      }
51  }

53  printf("Broj sa najvećom cifrom desetice je %d\n", broj);

55  return 0;
57  }

```

Rešenje 0.1.23

```

1  #include <stdio.h>
    #include <stdlib.h>
3
    int main(){
5
        int n;
        int x, x_kopija, br_cifara;
7        int max_br_cifara, broj;
9        int i;

11     printf("Unesite broj n: ");

```

```

scanf("%d", &n);

13     if (n < 0)
15     {
17         printf("Neispravan unos.\n");
19         return -1;
21     }

23     if (n == 0)
25     {
27         printf("Nisu uneseni brojevi.\n");
29         return 0;
31     }

33     /* Maksimalan broj cifara se postavlja na 0,
35      * svaki broj ima vise od 0 cifara pa je ova vrednost
37      * neutralna. */
39     max_br_cifara=0;

41     printf("Unesite n brojeva: ");
43     for(i=0; i<n; i++){
45         scanf("%d", &x);

47         /* Odredjivanje broja cifara unetog broja x. */
49         x_kopija=abs(x);
51         br_cifara=0;
53         while(x_kopija!=0){
55             x_kopija=x_kopija/10;
57             br_cifara++;
59         }

61         /* Ako je broj cifara unetog broja veci od maksimalnog */
63         if(br_cifara>max_br_cifara){
65             /* Cuvamo ga */
67             max_br_cifara=br_cifara;
69             /* I zbog ispisa rezultata, cuvamo i originalni broj */
71             /* Zbog ovoga smo morali i da racunamo broj cifara nad kopijom
73             broja x kako ne bismo promenili njegovu vrednost */
75             broj=x;
77         }
79     }

81     printf("Najvise cifara ima broj %d\n", broj);

83     return 0;

85 }

```

Rešenje [0.1.24](#)


```

/* Sa standardnog ulaza se unosi ceo pozitivan broj n, a zatim i n
   celih brojeva. Napisati program koji ispisuje
2 broj sa najvecom vodecom cifrom. Vodeca cifra je prva cifra iz zapisa
   broja. Ukoliko ima vise takvih, ispisati
   prvi. */
4
#include <stdio.h>
6 #include <math.h>

8 int main(){

10     int n;
11     int x, x_kopija;
12     int broj;
13     int vodeca_cifra, max_vodeca_cifra;
14     int i;

16     /* Citamo vrednost sa ulaza */
17     printf("Unesite broj n: ");
18     scanf("%d", &n);

20     /* Postavljamo maksimalnu vodecu cifru na 0 - cifre broja su vece
       ili jednake od 0 pa je ova vrednost neutralna */
21     max_vodeca_cifra=0;

22     /* Ucitavamo broj po broj */
23     printf("Unesite n brojeva: ");
24     for(i=0; i<n; i++){
25         scanf("%d", &x);

26         /* Odredjujemo vodecu cifru broja */
27         x_kopija=abs(x);
28         while(x_kopija>10){
29             x_kopija=x_kopija/10;
30         }
31         vodeca_cifra=x_kopija;

32         /* Ako je izdvojena cifra veca od maksimalne vodece cifre */
33         if(vodeca_cifra>max_vodeca_cifra){
34             /* Cuvamo je */
35             max_vodeca_cifra=vodeca_cifra;
36             /* I zbog ispisa, cuvamo i broj u kojem se ona pojavljuje */
37             /* Zbog ovoga smo morali i da racunamo vodecu cifru nad kopijom
38                broja x kako ne bismo promenili njegovu vrednost */
39             broj=x;
40         }
41     }

42     /* Ispisujemo rezultat */
43     printf("%d\n", broj);
44
45     return 0;
46 }

```

```
48     return 0;
50 }
```

Rešenje 0.1.25

```
1  #include <stdio.h>
   #include <math.h>
3
   int main(){
5
       int x;
       int min, max;
7
       printf("Unesite brojeve: ");
9
       /* Prvi broj se učitava izvan petlje zbog
          * inicijalizacije maksimuma i minimuma. */
11      scanf("%d", &x);
       max=x;
13      min=x;
15
       /* Sve dok se ne unese 0, učitavaju se brojevi u petlji. */
       while(x!=0){
17
19           /* Provera da li je procitani broj veci od aktuelnog maksimuma.
              */
21           if(x>max)
               max=x;
23           /* Provera da li je procitani broj manji od aktuelnog minimuma.
              */
               if(x<min)
25                 min=x;
27
               /* Učitavanje narednog broja. */
               scanf("%d", &x);
29         }
31
33         printf("Razlika: %d\n", max-min);
35     return 0;
}
```

Rešenje 0.1.26

```
1  #include <stdio.h>
   #include <stdlib.h>
3
```

```

5 int main(){
7     int n;
7     int d;
        /* Uzastopni brojevi za koje se racuna rastojanje. */
9     int x, y;
    int broj_parova;
11    int i;

13
13    printf("Unesite brojeve n i d: ");
15    scanf("%d %d", &n, &d);

17        if (n < 0 || d < 0)
19        {
21            printf("Neispravan unos.\n");
            return -1;
        }

23    broj_parova=0;

25    printf("Unesite n brojeva: ");

27    /* Prvi broj se ucitava pre petlje. */
    scanf("%d", &x);

29
31    for(i=1; i<n; i++){
        scanf("%d", &y);

33        /* Provera da li su brojevi na rastojanju d. */
        if(abs(y-x)==d)
35            broj_parova++;

37        /* Broj iz tekuce iteracije se cuva kako bi
            * mogao da se upotrebljava u narednoj iteraciji. */
39        x=y;
    }

41
43    printf("Broj parova: %d\n", broj_parova);
45
    return 0;
}

```

Rešenje 0.1.27

```

1 #include <stdio.h>
3 int main()
4 {
5     int x;

```

```

7      /* Tezina trenutne pozicije u broju. Moze biti 1, 10, 100,
      * 1000 itd.
      */
9      int pozicija;
      /* Trenutna izdvojena cifra iz broja x. */
11     int cifra;
      /* Broj dobijen nakon transformacije. */
13     unsigned int y;

15     printf("Unesite broj: ");
      scanf("%d", &x);

17

19     if(x <= 0)
      {
21         printf("Nekorektan unos.\n");
          return -1;
23     }

25     /* Posto pocinjemo sa izdvajanjem cifara od cifre jedinica,
      * postavljamo se tezinu (stepen) pozicije na 1.
      */
27     pozicija = 1;
29     y = 0;

31     /* Provera da li ima cifara u zapisu broja. */
      while(x > 0) {

33         /* Izdvaaja se poslednja cifra iz zapisa. */
35         cifra = x % 10;

37         /* Provera da li je cifra parna. */
          if(cifra % 2 == 0){
39             /* I ako jeste, uvecava se. */
              cifra++;

41         }

43         /* Novi broj se formira tako sto se izdvojena cifra pomnozi
          * odgovarajucom tezinom (stepenom) pozicije.
          */
45         y += cifra*pozicija;

47         /* Priprema se broj za izdvajanje naredne cifre, uklanja se
          * poslednja cifra broja.
          */
51         x /= 10;

53         /* Uvecava se tezinu (stepen) pozicije. */
55         pozicija *= 10;

57     }

```

```

59     /* Ispisuje se izracunatu vrednost. */
    printf("%d\n", y);

61     return 0;
63 }

```

Rešenje 0.1.28

```

1  #include <stdio.h>
   #include <math.h>
3  #include <stdlib.h>

5  int main() {

7      int x;
      /* Tezina trenutne pozicije u broju. Moze biti 1, 10, 100,
9       * 1000 itd.
       */
11     int stepen_deset;
      /* Trenutna izdvojena cifra iz broja x. */
13     int cifra;
      /* Redni broj cifre koja se trenutno obradjuje, gledano s desna
15     * na levo.
       */
17     int rbr;
      /* Broj dobijen nakon transformacije. */
19     int y;
      /* Promenljiva znak cuva znak unesenog broja. Moze biti
21     * -1 za negativnu vrednost ili 1 za pozitiivnu vrednost.
       */
23     int znak = 1;

25     /* Ucitavanje broja. */
    printf("Unesite broj: ");
27     scanf("%d", &x);

29     if(x <= 0)
    {
31         x = abs(x);
        znak = -1;
33     }
      /* Postavlja se vrednost stepena na 0 - to znaci da se prvo mnozi
        sa
35     * 10^0=1.
       */
37     stepen_deset = 0;

39     /* Postavlja se vrednost broja koji se formira na 0. */
    y = 0;
41     /* Postavlja se redni broj pozicije na 0. */

```

```

43     rbr = 0;

45     /* Provera da li ima cifara u zapisu broja. */
46     while(x > 0) {

47         /* Izdvajanje cifre. */
48         cifra = x%10;

49

51         /* Proverava se da li je pozicija izdvojene cifre parna -
52          * cifre na parnim pozicijama se zadržavaju.
53          */
54         if(rbr % 2 == 0) {
55             /* Ako jeste parna
56              * izdvojena cifra se dodaje novom broju.
57              * Neophodno je izvršiti promenu tipova, jer je double
58              povratni
59              * tip funkcije pow.
60              */
61             y += cifra * ((int) pow(10, stepen_deset));

62             /* Uvecava se stepen zbog naredne cifre. */
63             stepen_deset++;
64         }

65         /* Azurira se redni broj cifre. */
66         rbr++;
67         /* I priprema se broj za naredno izdvajanje. */
68         x /= 10;
69     }

71     y = znak*y;

73     /* Ispisuje se rezultat. */
74     printf("%d\n", y);

75

77     return 0;
}

```

Rešenje 0.1.29

```

1  #include <stdio.h>

3  int main(){
4      int n, novo_n;
5      int stepen;
6      int cifra_levo, cifra_sredina, cifra_desno;

7

8      /* Ucitavanje broja. */
9      printf("Unesite broj: ");
10     scanf("%d", &n);

```

```

11  if (n <= 0)
13  {
14      printf("Neispravan unos.\n");
15      return -1;
16  }
17
18  /* Stepen broja 10 sa kojim se mnoze cifre izdvojenog broja. */
19  stepen=1;
20
21  /* Nova vrednost broja. */
22  novo_n=0;
23
24  /* Provera da li u zapisu broja postoje barem tri cifre. */
25  while(n>99){
26      /* Izdvaja se srednja cifra, cifra desno od nje i cifra
27       * levo od nje: npr. za trojku 583 8 je srednja cifra,
28       * 3 je cifra desno, a 5 cifra levo.
29       */
30      cifra_desno=n%10;
31      cifra_sredina=(n/10)%10;
32      cifra_levo=(n/100)%10;
33
34      /* U novi broj se smesta desna cifra. */
35      novo_n+=cifra_desno*stepen;
36
37      /* Azurira se vrednost stepena. */
38      stepen=stepen*10;
39
40      /* Provera da li je srednja cifra jednaka zbiru leve i
41       * desne cifre.
42       */
43      if(cifra_levo+cifra_desno==cifra_sredina){
44
45          /* Treba izbaciti srednju cifru, pa broj n se azurira
46           * tako sto se podeli sa 100.
47           */
48          n=n/100;
49      }
50      else{
51
52          /* Inace, zadrzava se srednja cifra i odbacuje
53           * se samo poslednja.
54           */
55          n=n/10;
56      }
57  }
58
59  /* Na novi broj se dodaje preostali dvocifreni
60   * ili jednocifreni broj.
61   */
62  novo_n=n*stepen+novo_n;

```

```

63      /* Ispisivanje rezultata. */
65      printf("%d\n", novo_n);

67      return 0;

69  }

```

Rešenje 0.1.30

```

1  #include <stdio.h>
   #include <math.h>

3

5  int main() {

7      int x;
       int broj_cifara;
       int min_stepen, max_stepen;
9      int pom;
       int leva_cifra, desna_cifra;
11     int indikator;

13     printf("Unesite broj: ");
       scanf("%d", &x);

15

17     /* Ako je korisnik uneo negativan broj, analizira se njegova
       * apsolutna vrednost.
       */
19     if(x < 0)
        x=-x;

21

23     /* Odredjuje se broj cifara u zapisu broja x da bi moglo da
       * se izdvajaju istovremeno cifre i sa leve i sa desne strane.
       */
25     broj_cifara = 0;
       pom = x;
       while(pom > 0) {
27         pom /= 10;
29         broj_cifara++;
31     }

33     /* Odredjuje se stepen koji stoji uz krajnju levu cifru broja. */
       max_stepen = (int) pow (10, broj_cifara-1);

35

37     /* Indikator je promenljiva koja ukazuje da li je broj
       * palindrom ili ne.
       */
39     indikator=1;
       while(x!=0 && indikator==1){

```



```

41      /* Izdvaja se leva cifra. */
      leva_cifra=x/max_stepen;
43      /* Izdvaja se desna cifra. */
      desna_cifra=x%10;
45      /* Ako su cifre razlicite, odmah se moze zakljuciti da
      * broj nije palindrom i prekida se izvršavanje petlje.
47      */
      if(leva_cifra!=desna_cifra){
49          indikator=0;
          break;
51      }
      /* Formira se nova vrednost broja x tako sto se odbacuje
53      * krajnja leva i krajnja desna cifra.
      */
      x=(x%max_stepen-x%10)/10;
      /* Koriguje se maksimalan stepen tako dobijenog broja -
57      * deli se sa 100 jer su odbacene 2 cifre.
      */
      max_stepen=max_stepen/100;
59  }

61      /* Ispisuje se rezultat. */
63      if(indikator==1)
          printf("Broj je palindrom!\n");
65      else
          printf("Broj nije palindrom!\n");
67
69      return 0;
    }

```

Rešenje 0.1.31

```

1  #include <stdio.h>
2
3  int main()
4  {
5      /* Pamtimo uzastopna dva Fibonacijeva broja i na osnovu njih
      racunamo sledeci. */
      /* Promenljive prvi i drugi su brojevi koje pamtimo i na osnovu
      njih racunamo treci. */
7     /* Na osnovu teksta zadatka, promenljive prvi i drugi postavljamo
      na 1. */
      int prvi = 1;
9     int drugi = 1;
      int treci;
11    /* Promenljiva pozicija je podatak koji ucitavamo i odnosi se na
      poziciju u Fibonacijevom nizu
      za koju treba izracunati vrednost. */
13    int pozicija;

```

```

15  /* Promenljiva i oznacava do koje pozicije smo izracunali vrednosti
    . Kako imamo prve dve
16  vrednosti, ovu promenljivo postavljamo na 2. */
17  int i = 2;
18
19  printf("Unesite poziciju u Fibonacijevom nizu: ");
20  scanf("%d", &pozicija);
21
22  /* Pozicija ne moze biti 0 i ne moze biti negativan broj. */
23  if (pozicija < 1)
24  {
25      printf("Neispravan unos. Pozicija u Fibonacijevom nizu mora biti
    pozitivan broj koji nije 0!\n");
26      return -1;
27  }
28
29  while(i < pozicija)
30  {
31      /* Na osnovu dva uzastopna racunamo treci. */
32      treci = prvi + drugi;
33
34      /* Potom razmenjujemo vrednosti. Uzastopna dva koja pamtimo
    postaju
35      sledeca uzastopna dva broja Fibonacijevog niza. */
36      prvi = drugi;
37      drugi = treci;
38
39      /* Prelazimo na racunanje sledeceg broja na sledecoj poziciji. */
40      i++;
41  }
42
43  printf("Trazeni broj je: %d\n", drugi);
44
45  return 0;
46  }

```

Rešenje 0.1.32

```

#include<stdio.h>
2  int main()
3  {
4      int a0;
5      int an,an1;
6
7      printf("Unesi pocetni clan niza brojeva:");
8      scanf("%d",&a0);
9
10     if (a0 <= 0)
11     {
12         printf("Nekorektan unos. Broj mora biti pozitivan.\n");
13         return -1;
14     }
15 }

```

```

14     }

16     printf("%d\n", a0);

18     an=a0;
19     while(an!=1)
20     {
21         if (an%2) /* Ukoliko je vrednost izraza an%2 razlicita od nule,*/
22         {          /* izraz se tumaci kao tacan i izvorsavaju se naredbe iz
23             if grane. */
24                 an1=(3*an+1)/2;
25         }
26         else /* U suprotnom, ukoliko je vrednost izraza an%2 jednaka nuli
27             , izraz */
28         {      /* se tumaci kao netacan i izvorsavaju se naredbe iz else
29             grane. */
30                 an1=an/2;
31         }
32         printf("%d\n",an1);
33         an=an1;
34     }

35     return 0;
36 }

```

Rešenje 0.1.33

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      int format;
7      /* Pomocna promenljiva koja služi kao brojac u petlji. */
8      int i;
9      /* Trenutne vrednosti za sirinu i visinu i
10         * pomocna promenljiva za promene u petlji.
11         */
12     double sirina, duzina, nova_duzina;
13     unsigned int konacna_sirina, konacna_duzina;
14
15     printf("Uneti format papira: ");
16     scanf("%d", &format);
17
18     if (format <= 0)
19     {
20         printf("Neispravan unos.\n");
21         return -1;
22     }
23

```

```

25  /* duzina/sirina = 1 : sqrt(2) duzina*sirina = 1000x1000mm^2
    * Na osnovu ovih odnosa dobijamo pocetnu vrednost za sirinu
    * i duzinu, odnosno vrednosti za papir A0.
27  */
    duzina = sqrt(1000*1000/sqrt(2));
29  sirina = sqrt(2)*duzina;

31  /* Kako vec imamo odredjenu sirinu i duzinu za papir A0,
    * petlju krecemo od izracunavanja za papir A1, pa
33  * brojac i postavljamo na 1.
    */
35  for(i=1; i<=format; i++)
    {
37      nova_duzina = sirina/2;
        sirina = duzina;
39      duzina = nova_duzina;
    }

41  /* Duzina i sirina celi brojevi. */
43  konacna_sirina = (unsigned int)sirina;
    konacna_duzina = (unsigned int)duzina;

45  printf("%u %u\n", konacna_duzina, konacna_sirina);

47  return 0;
49 }

```

Rešenje 0.1.34

```

1  #include <stdio.h>

3  int main()
    {
5      char c;

7      /* Funkcija getchar ucitava jedan karakter.
        * Naredbom dodele (c=getchar()) promenljivoj c
9        * bice dodeljena vrednost ascii koda unetog karaktera.
        * Obratiti posebnu paznju na zagrade. */

11     while((c=getchar())!='.')
13     {
        if (c>='A' && c<='Z')
15         /* Razlika izmedju ascii koda svakog malog i odgovarajuceg
            * velikog slova je konstanta koja se moze sracunati izrazom
17         * 'a'-'A' (i iznosi 32). */
            putchar(c+'a'-'A');
19         else if (c>='a' && c<='z')
            putchar(c-'a'+'A');
21         else
            putchar(c);
    }

```

```
23     }
25     return 0;
    }
```

Rešenje 0.1.35

```
#include <stdio.h>

2
int main()
4 {
    char c;

6
    /* Inicijalizacija brojaca na 0. */
8     int br_v=0;
    int br_m=0;
10    int br_c=0;
    int br_b=0;
12    int br_k=0;
    int suma=0;

14
    /* Petlja se završava kada korisnik ne unese karakter,
16     * vec zada konstantu EOF .
    * Ova konstanta se zadaje kombinacijom tastera CTRL+D.
18     * U tom slucaju, getchar() vraca -1. */
    while((c=getchar())!=EOF)
20     {
        if (c>='A' && c<='Z')
22             br_v++;
        else if (c>='a' && c<='z')
24             br_m++;
        else if (c>='0' && c<='9')
26             {
                br_c++;
28                /* Kada od promenljive tipa char oduzimamo karakter
                * (ili neku drugu promenljivu tipa char), zapravo se
30                * vrsi oduzimanje njihovih ascii vrednosti i dobija se
                * broj. */
32                suma=suma+c-'0';
            }
34        else if (c=='\t' || c=='\n' || c==' ')
            br_b++;

36        br_k++;
38    }

40    printf("velika: %d, mala: %d, cifre: %d, beline: %d \n", br_v, br_m
        , br_c, br_b);
    printf("suma cifara: %d\n", suma);

42
    return 0;
```

```
44 }
```

Rešenje 0.1.36

```
#include <stdio.h>

2
int main()
4 {
    /* Promenljiva i je brojac. */
    int n, i;
    /* Brojaci za svaki od samoglasnika. */
    int broj_a = 0, broj_e = 0, broj_i = 0, broj_o = 0, broj_u = 0;
    /* Promenljiva c je tekuci ucitani karakter. */
    char c, belina;

    printf("Unesite broj n: ");
    scanf("%d", &n);

    if (n < 0)
    {
        printf("Neispravan unos.\n");
        return -1;
    }

    for(i=0; i<n; i++)
    {
        /* Prvo se cita belina koja se nalazi nakon prethodnog
        * unosa, pa tek posle procitane beline se cita uneseni
        * karakter. */
        scanf("%c%c", &belina, &c);

        /* Provera da li je ucitani karakter samoglasnik. */
        switch(c)
        {
            case 'a': case 'A': broj_a++; break;
            case 'e': case 'E': broj_e++; break;
            case 'i': case 'I': broj_i++; break;
            case 'o': case 'O': broj_o++; break;
            case 'u': case 'U': broj_u++; break;
        }
    }

    printf("samoglasnik a: %d\n", broj_a);
    printf("samoglasnik e: %d\n", broj_e);
    printf("samoglasnik i: %d\n", broj_i);
    printf("samoglasnik o: %d\n", broj_o);
    printf("samoglasnik u: %d\n", broj_u);

    return 0;
46 }
```

Rešenje 0.1.37

```
1  /* Sa standardnog ulaza se unosi ceo broj n, a zatim i n karaktera.
   Napisati program koji proverava da li se od
   unetih karaktera moze napisati rec Zima. */
3
5  #include <stdio.h>
   #include <math.h>
7
   int main(){
9
   int n;
   int broj_Z, broj_i, broj_m, broj_a;
11  char novi_red, c;
   int i;
13
   broj_Z=0;
15  broj_i=0;
   broj_m=0;
17  broj_a=0;
19
   printf("Unesite broj: ");
21  scanf("%d", &n);
23
   /* Ucitavanje karakter po karakter. */
   for(i=0; i<n; i++){
25     printf("Unestite %d. karakter: ", i+1);
     /* Prvo se cita belina koja se nalazi nakon prethodnog
27     * unosa, pa tek posle procitane beline se cita
     uneseni
     * karakter. */
29     scanf("%c%c", &novi_red, &c);
31
     /* Analiziramo karakter */
     switch(c){
33         case 'Z':
             broj_Z++;
35         break;
         case 'i':
             broj_i++;
37         break;
         case 'm':
             broj_m++;
39         break;
         case 'a':
             broj_a++;
41         break;
43     }
45 }
47
   /* Ako u unosu ima barem jedno veliko slovo z i
```

```

49         * barem po jedno malo slovo i, m i a, rec se moze
        * napisati. A u suprotnom ne moze. */
51     if(broj_Z && broj_i && broj_m && broj_a){
        printf("Moze se napisati rec Zima.\n");
53     }
    else{
55         printf("Ne moze se napisati rec Zima.\n");
    }
57
    return 0;
59 }

```

Rešenje 0.1.38

```

1  #include <stdio.h>

3  int main()
4  {
5      int n;
        /* Brojac. */
7      int i;
        /* Promenljiva u kojoj se cuva suma kubova. */
9      int s;

11

12     printf("Unesite pozitivan ceo broj:");
13     scanf("%d", &n);

14
15     if (n < 0)
16     {
17         printf("Neispravan unos.\n");
18         return -1;
19     }

20
21     i=1;
22     s=0;

23
24     for(i=1;i<=n;i++)
25         s+=i*i*i;

26
27     printf("Suma kubova od 1 do %d: %d\n", n, s);
28     return 0;
29 }

```

Rešenje 0.1.39

```

#include <stdio.h>
2

```



```

1  int main()
4  {
6      int n;
7      /* Brojac. */
8      int i;
9      /* Promenljiva u kojoj se cuva suma kubova. */
10     int s;

12     printf("Unesite pozitivan ceo broj:");
13     scanf("%d", &n);

14     if (n < 0)
15     {
16         printf("Neispravan unos.\n");
17         return -1;
18     }

20     i=1;
21     s=0;

22     for(i=1;i<=n;i++)
23     {
24         s+=i*i*i;
25         printf("i=%d, s=%d\n", i, s);
26     }

28     return 0;
30 }

```

Rešenje 0.1.40

```

1  #include <stdio.h>

3  int main(){
4      int n, i;
5      float x, S, stepen;

7      printf("Unesite redom brojeve x i n: ");
8      scanf("%f %d", &x, &n);

9      if (n < 0)
10     {
11         printf("Neispravan unos.\n");
12         return -1;
13     }

15     /* Inicijalizacija sume. */
16     S=0;

18     /* Stepen promenljiva ce sadrzati vrednosti stepena x^n.

```

```

21     * Pocetna vrednost joj je 1 jer je  $x^0 = 1$ . */
    stepen=1;

23     for(i=1; i<=n; i++)
    {
25         stepen=stepen*x;
        S=S+i*stepen;
27     }

29     printf("S=%f\n", S);

31     return 0;
}

```

Rešenje 0.1.41

```

1  #include <stdio.h>
int main(){
3     unsigned n, i;
    float x, S, stepen;

5

    printf("Unesite redom brojeve x i n: ");
7     scanf("%f %u", &x, &n);

9     S=1;
    stepen=1;
11    for(i=1; i<=n; i++)
    {
13        stepen=stepen*x;
        S=S+1/stepen;
15    }

17    printf("S=%f\n", S);

19    return 0;
}

```

Rešenje 0.1.42

```

1  #include <stdio.h>
2  #include <math.h>
int main(){
4     /* Promenljiva i je brojac, promenljiva S cuva izracunatu sumu,
    * a promenljiva clan je tekuci clan niza. */
    int i;
6     float S;
    float x, eps;
8     float clan;

10

```

```

12     printf("Unesite x: ");
    scanf("%f", &x);

14     printf("Unesite tacnost eps: ");
    scanf("%f", &eps);

16
    S=0;
18     /* Prvi clan sume je 1. */
    clan = 1;
20     i=1;
    while(clan > eps){
22         S=S+ clan;
        clan = clan*x / i;
24         i++;
    }

26     printf("S=%f\n", S);

28     return 0;

30 }

```

Rešenje 0.1.43

```

#include <stdio.h>
2 #include <math.h>
int main(){
4     /* Promenljiva i je brojac, promenljiva S cuva izracunatu
        sumu,
        * promenljiva znak moze biti 1 ili -1 i odredjuje znak
6     * trenutnog clana sume, a promenljiva clan je tekuci clan
        niza. */
    int i, znak;
8    float S;
    float x, eps, clan;

10    printf("Unesite x: ");
12    scanf("%f", &x);

14    printf("Unesite tacnost eps: ");
    scanf("%f", &eps);

16

18    S=0;
    clan = 1;
20    i=1;
    znak=-1;

22    /* Kako clanovi sume mogu biti negativni, potrebno je
        posmatrati
24    * apsolutnu vrednost clana. */
    while(fabs(clan)>eps)

```

```

26         {
27             S = S + clan;
28
29             /* Promena znaka. */
30             clan = clan*x / i;
31             clan *= znak;
32
33             i++;
34         }
35
36     printf("S=%f\n", S);
37
38     return 0;
39 }

```

Rešenje 0.1.44

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      int n, i;
7      double x;
8      double stepen = 1;
9      double proizvod = 1;
10
11     printf("Unesite redom brojeve x i n: ");
12     scanf("%lf %d", &x, &n);
13
14     if (n <= 0)
15     {
16         printf("Neispravan unos.\n");
17         return -1;
18     }
19
20     for(i=0; i<n; i++)
21     {
22         stepen *= x;
23         proizvod *= 1 + cos(stepen);
24     }
25
26     printf("Proizvod = %lf\n", proizvod);
27
28     return 0;
29 }

```

Rešenje 0.1.45

```

1  #include <stdio.h>

3  int main()
4  {
5      int n, i;
6      double Razlomak;

7

8      printf("Unesite prirodan broj: ");
9      scanf("%d", &n);

10

11     if (n <= 0)
12     {
13         printf("Neispravan unos.\n");
14         return -1;
15     }

16

17     Razlomak = n;

18

19     /* Razlomak se izracunava "od nazad", odnosno, krece se
20      * od najnizeg razlomka 1/n i od njega se nadalje formira
21      * sledeci, "visi" razlomak itd. Završava se kada se stigne
22      * do koraka 0 + 1/R. */
23     for(i=n-1; i>= 0; i--)
24         Razlomak = i + 1/Razlomak;

25

26     printf("Razlomak = %lf\n", Razlomak);

27

28     return 0;
29 }

```

Rešenje 0.1.46

```

1  #include <stdio.h>
2  #include <math.h>
3  int main(){
4      /* Promenljiva i je brojac, promenljiva S cuva izracunatu
5       * sumu,
6       * promenljiva znak moze biti 1 ili -1 i odredjuje znak
7       * trenutnog clana sume, a promenljiva clan je tekuci clan
8       * niza. */
9      int i, znak, n;
10     float S;
11     float x, clan;

12     printf("Unesite x i n: ");
13     scanf("%f%d", &x, &n);

14

15     if (n <= 0)
16     {
17         printf("Neispravan unos.\n");
18     }
19 }

```

```

17         return -1;
18     }
19
20
21     S=0;
22     clan = 1;
23     i=1;
24     znak=-1;
25
26     /* Kako clanovi sume mogu biti negativni, potrebno je
27        posmatrati
28        * apsolutnu vrednost clana. */
29     while(i <= 2*n)
30     {
31         S = S + clan;
32
33         /* Promena znaka. */
34         /* Svaki clan suma se od prethodnog clana razlikuje za x
35            ^2/(i*(i+1)). */
36         clan = clan*x*x / (i*(i+1));
37         clan *= znak;
38
39         i+=2;
40     }
41
42     printf("S=%f\n", S);
43
44     return 0;
45 }

```

Rešenje 0.147

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      int n, i;
7      /* Promenljiva clan je deo proizvoda i predstavlja 1/i!. */
8      double clan;
9      double S = 1;
10
11      printf("Unesite prirodan broj: ");
12      scanf("%d", &n);
13
14      if (n <= 1)
15      {
16          printf("Neispravan unos.\n");
17          return -1;
18      }
19

```

```

21     clan = 1;
    for(i=2; i<=n; i++)
    {
23         clan = clan/i;
        S *= 1 + clan;
25     }

27     printf("S = %lf\n", S);

29     return 0;
}

```

Rešenje 0.1.48

```

1  #include <stdio.h>
    #include <math.h>
3
    int main()
5  {
        int n, i, znak = -1;
7      /* Promenljiva clan je deo proizvoda i predstavlja 1*3*5*...*i.
        */
        long int clan;
9        long int S = 0;

11       printf("Unesite prirodan broj: ");
        scanf("%d", &n);

13
        if (n < 5 || n % 2 == 0)
15         {
            printf("Neispravan unos.\n");
            return -1;
17         }

19
        clan = 1*3;
21       for(i=5; i<=n; i+=2)
        {
23           clan = znak*clan*i;
            S += clan;
25         }

27       printf("S = %ld\n", S);

29       return 0;
}

```

Rešenje 0.1.49

```

#include <stdio.h>

```

```

2  #include <math.h>

4  int main()
{
6      int n, i;
       double P;
8      double x, a;

10     printf("Unesite dva relana broja x i a: ");
       scanf("%lf%lf", &x, &a);

12     printf("Unesite prirodan broj: ");
       scanf("%d", &n);

16     if (n <= 0)
       {
18         printf("Neispravan unos.\n");
           return -1;
20     }

22     P = x;
       for(i=0; i<n; i++)
24         P = (P+a)*(P+a);

26     printf("Izraz = %lf\n", P);

28     return 0;
}

```

Rešenje 0.1.50

```

2  #include <stdio.h>

4  int main()
{
6      unsigned int n, i, j;

       printf("Unesite broj n: ");
       scanf("%u", &n);

10     for (i=1; i<=n; i++)
       {
12         for (j=1; j<=n; j++)
           /* U tablici mnozenja vrednost svakog polja je proizvod
14            * vrste i kolone u kojoj se nalazi. */
           printf("%3d ", i*j);

16         printf("\n");
18     }

20     return 0;
}

```



```
}
```

```
1  #include <stdio.h>
3  int main()
4  {
5      unsigned int n, i, j;
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     /* Brojac koji broji koliko brojeva je ispisano
11      * u jednom redu. */
12     j = 0;
13     for (i=1; i<=n; i++)
14     {
15         printf("%3d ", i);
16         /* Uvecavamo brojac */
17         j++;
18
19         /* Kada je ispisano n brojeva u jednom redu,
20          * ispisuje se znak za novi red, da bi ispis
21          * krenuo u novom redu i vrednost brojaca j
22          * se postavlja na 0 jer u novom redu jos ni
23          * jedan broj nije ispisan. */
24         if (j==n)
25         {
26             j = 0;
27             printf("\n");
28         }
29     }
31     return 0;
32 }
```

```
1  #include <stdio.h>
2
3  int main()
4  {
5      unsigned int n, i, j;
6
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     for (i=1; i<=n; i++)
11     {
12         for (j=0; j<n; j++)
13             if ((j+i)%n == 0)
14                 printf("%3d", n);
15         else
16             printf("%3d", (j+i)%n);
```

```

18     printf("\n");
19 }
20
21     return 0;
22 }

```

```

#include <stdio.h>

2
int main()
4 {
    unsigned int n, i, j;

6     printf("Unesite broj n: ");
8     scanf("%u", &n);

10    for (i=0; i<n; i++)
    {
12        for (j=0; j<n-i; j++)
            printf("(%d, %d)", i, j);

14        printf("\n");
16    }

18    return 0;
19 }

```

Rešenje 0.1.51

```

#include <stdio.h>

2
int main()
4 {
    unsigned int n, i, j;

6     printf("Unesite broj n: ");
8     scanf("%u", &n);

10    for (i=0; i<n; i++)
    {
12        /* Kvadrat predstavlja tabelu sa n vrsta
            * gde svaka vrsta sadrzi n polja, a svako
14        * polje je isto i predstavlja karakter *.
            */
16        for (j=0; j<n; j++)
            printf("*");
18        printf("\n");
19    }

20    return 0;

```

22 }

```
#include <stdio.h>

2
int main()
4 {
    unsigned int n, i, j;

6
    printf("Unesite broj n: ");
8    scanf("%u", &n);

10
    for (i=0; i<n; i++)
    {
12        /* Kvadrat predstavlja tabelu sa n vrsta
         * gde su ivice karakter *, a unutrasnjost
14        * kvadrata je karakter blanko.
         */
16        for (j=0; j<n; j++)
            /* Provera da li je ivica. */
18            if (j ==0 || j==n-1 || i == 0 || i== n-1)
                printf("*");
20            else
                printf(" ");
22        printf("\n");
    }

24
    return 0;
26 }
```

```
#include <stdio.h>

2
int main()
4 {
    unsigned int n, i, j;

6
    printf("Unesite broj n: ");
8    scanf("%u", &n);

10
    for (i=0; i<n; i++)
    {
12        /* Kvadrat predstavlja tabelu sa n vrsta
         * gde su ivice karakter *, a unutrasnjost
14        * kvadrata je karakter blanko osim na mestima
         * na kojima je glavna dijagonala.
         */
16        for (j=0; j<n; j++)
            /* Provera da li je ivica ili glavna dijagonala. */
18            if (j ==0 || j==n-1 || i == 0 || i== n-1 || i==j)
                printf("*");
20            else
                printf(" ");
22        printf("\n");
    }
}
```

```

24     printf("\n");
    }

26     return 0;
    }

```

Rešenje 0.1.52

```

1  #include <stdio.h>

3  int main()
4  {
5      unsigned int n, i, j;

7      printf("Unesite broj n: ");
8      scanf("%u", &n);

9      for (i=0; i<n; i++)
10     {
11         /* Veliko slovo X se moze posmatrati kao dijagonale
12          * kvadrata (glavna i sporedna). Zato, treba ispisivati
13          * blanko na mestima gde nije dijagonala, a
14          * karakter * na mestima gde je neka od dijagonala.
15          */
16         for (j=0; j<n; j++)
17             /* Provera da li je mesto glavne ili sporedne dijagonale. */
18             if (i == j || i+j == n-1)
19                 printf("*");
20             else
21                 printf(" ");
22             printf("\n");
23     }

25     return 0;
27 }

```

Rešenje 0.1.53

```

1  #include <stdio.h>

3  int main()
4  {
5      unsigned int n, i, j;

7      printf("Unesite broj n: ");
8      scanf("%u", &n);

9      if (n % 2 == 0)
10     {
11

```

```

13     printf("Pogresan unos.\n");
14     return -1;
15 }
16
17 for (i=0; i<n; i++)
18 {
19     for (j=0; j<n; j++)
20         if (i == n/2 || j==n/2)
21             printf("+");
22     else
23         printf(" ");
24     printf("\n");
25 }
26
27 return 0;
28 }

```

Rešenje 0.1.54

```

#include <stdio.h>
2
int main()
4 {
    unsigned int n, i, j;
6
    printf("Unesite broj n: ");
8    scanf("%u", &n);
10
    for (i=0; i<n; i++)
    {
12        for (j=0; j<n-i; j++)
            printf("*");
14        printf("\n");
    }
16
    return 0;
18 }

```

```

#include <stdio.h>
2
int main()
4 {
    unsigned int n, i, j;
6
    printf("Unesite broj n: ");
8    scanf("%u", &n);
10
    for (i=0; i<n; i++)
    {
12        for (j=0; j<=i; j++)

```

```
14     printf("*");
    printf("\n");
    }
16
18     return 0;
    }
```

```
2     #include <stdio.h>
3
4     int main()
5     {
6         unsigned int n, i, j;
7
8         printf("Unesite broj n: ");
9         scanf("%u", &n);
10
11        for (i=0; i<n; i++)
12        {
13            /* Prvo se ispisuju beline koje prethode karakterima *. */
14            for (j=0; j<i; j++)
15                printf(" ");
16            /* Posle belina se ispisuje potreban broj karaktera *. */
17            for (j=0; j<n-i; j++)
18                printf("*");
19            printf("\n");
20        }
21
22        return 0;
    }
```

```
2     #include <stdio.h>
3
4     int main()
5     {
6         unsigned int n, i, j;
7
8         printf("Unesite broj n: ");
9         scanf("%u", &n);
10
11        /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
12        for (i=0; i<n; i++)
13        {
14            /* Prvo se ispisuju beline koje prethode karakterima *. */
15            for (j=0; j<n-i-1; j++)
16                printf(" ");
17            /* Posle belina se ispisuje potreban broj karaktera *. */
18            for (j=0; j<=i; j++)
19                printf("*");
20            printf("\n");
21        }
22    }
```

```
22 return 0;
}
```

```
1 #include <stdio.h>

3 int main()
{
5     unsigned int n, i, j;

7     printf("Unesite broj n: ");
    scanf("%u", &n);

9

11    /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
    for (i=0; i<n; i++)
13    {
        /* Prvo se ispisuju beline koje prethode karakterima *. */
15        for (j=0; j<n-i-1; j++)
            printf(" ");
17        /* Posle belina se ispisuje potreban broj karaktera *. */
        for (j=0; j<=i; j++)
19            printf("*");
        printf("\n");
21    }

23    /* Potrebno je iscrtati i donji deo slike, odnosno donji trougao.
     * Brojac i odredjuje koji red donjeg trougla se trenutno iscrtava.
25     * Kako je prvi red donjeg trougla vec iscrtan (to je poslednji
     * red gornjeg trougla), brojac se postavlja na 1. */
27    for (i=1; i<n; i++)
    {
29        /* Prvo se ispisuju beline koje prethode karakterima *. */
        for (j=0; j<i; j++)
31            printf(" ");
        /* Posle belina se ispisuje potreban broj karaktera *. */
33        for (j=0; j<n-i; j++)
            printf("*");
35        printf("\n");
    }

37    return 0;
39 }
```

```
1 #include <stdio.h>

3 int main()
{
5     unsigned int n, i, j;
    char c, blanko;

7

    printf("Unesite broj n: ");
```

```

9      scanf("%u", &n);

11     printf("Unesite karakter c: ");
    /* Zbog pritiskanja tastera ENTER nakon unosa promenljive
13      * broj potrebno je učitati karakter za novi red u promenljivu
      * blanko pre učitavanja karaktera kojim se iscrtava
15      * trougao. */
    scanf("%c%c", &blanko, &c);

17

19     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
    for (i=0; i<n; i++)
    {
21         /* Iscrtavaju se samo ivice trougla, ostalo se popunjava belinama
          . */
        for (j=0; j<=i; j++)
23             if (i == n-1 || j==0 || j==i)
                printf("%c", c);
25             else
                printf(" ");
27         printf("\n");
    }

29

31     return 0;
}

```

Rešenje 0.1.55

```

1      #include <stdio.h>

3      int main()
    {
5          unsigned int n, i, j;

7          printf("Unesite broj n: ");
          scanf("%u", &n);

9

11         /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
        for (i=0; i<n; i++)
        {
13             /* Prvo se ispisuju beline koje prethode karakterima *. */
            for (j=0; j<n-i-1; j++)
15                 printf(" ");
            /* Posle belina se ispisuje potreban broj karaktera *. */
17             for (j=0; j<2*i+1; j++)
                printf("*");
19             printf("\n");
        }

21

23     return 0;
}

```



```

1  #include <stdio.h>

3  int main()
4  {
5      unsigned int n, i, j;

7      printf("Unesite broj n: ");
8      scanf("%u", &n);

9      /* Brojac i odredjuje koliko redova se ispisuje.
10     * Radi lakseg izracunavanja koliko zvezdica i praznina
11     * je potrebno ispisati u svakom redu, i se postavlja na
12     * n-1 i smanjuje u svakoj iteraciji petlje. */
13     for (i=n-1; ;i--)
14     {
15         /* Prvo se ispisuju beline koje prethode karakterima *. */
16         for (j=0; j<n-i+1; j++)
17             printf(" ");
18         /* Posle belina se ispisuje potreban broj karaktera *. */
19         for (j=0; j<2*i+1; j++)
20             printf("*");
21         printf("\n");

22         /* Posebna paznja mora da se obrati na cinjenicu da su brojac
23         * tipa unsigned int. Problem nastaje kada je i==0 i pokusa se
24         * oduzimanje
25         * (i--). Posto su brojevi unsigned, nova vrednost nece biti -1,
26         * vec pozitivan ceo broj. Imajuci to na umu, uslov i>=0 ne moze
27         * da
28         * se stavi u uslov za for petlju. Mnogo sigurnije je brojace
29         * deklarirati da budu tipa int i izbeci ovakvu vrstu problema.
30         */
31         if (i == 0)
32             break;
33     }

34     return 0;
35 }

```

```

1  #include <stdio.h>

3  int main()
4  {
5      unsigned int n;
6      int i, j;

7      printf("Unesite broj n: ");
8      scanf("%u", &n);

9      /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
10     for (i=0; i<n; i++)

```

```

13 {
14     /* Prvo se ispisuju beline koje prethode karakterima *. */
15     for (j=0; j<n-i-1; j++)
16         printf(" ");
17     /* Posle belina se ispisuje potreban broj karaktera *. */
18     for (j=0; j<2*i+1; j++)
19         printf("*");
20     printf("\n");
21 }

22
23 /*Sada se ispisuje donji trougao. Kako je prvi red donjeg trougla
24 * vec ispisan (poslednji red gornjeg trougla), potrebno je
25 * naciniti jednu
26 * iteraciju manje. */

27 /* Brojac i odredjuje koliko redova se ispisuje.
28 * Radi lakseg izracunavanja koliko zvezdica i praznina
29 * je potrebno ispisati u svakom redu, i se postavlja na
30 * n-1 i smanjuje u svakoj iteraciji petlje. */
31 for (i=n-2; i>= 0; i--)
32 {
33     /* Prvo se ispisuju beline koje prethode karakterima *. */
34     for (j=0; j<n-i-1; j++)
35         printf(" ");
36     /* Posle belina se ispisuje potreban broj karaktera *. */
37     for (j=0; j<2*i+1; j++)
38         printf("*");
39     printf("\n");
40 }

41
42 return 0;
43 }

```

```

1 #include <stdio.h>

2
3 int main()
4 {
5     unsigned int n, i, j;

6
7     printf("Unesite broj n: ");
8     scanf("%u", &n);

9
10    /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
11    for (i=0; i<n; i++)
12    {
13        /* Prvo se ispisuju beline koje prethode karakterima *. */
14        for (j=0; j<n-i-1; j++)
15            printf(" ");
16        /* Posle belina se ispisuje sam trougao. Ako je brojac na ivici
17        onda
18        * se ispisuje karakter *, a inace praznina. Takodje, proverava
19        se

```

```

19     * da li se ispisuje poslednji red (i==n) i u njemu se ispisuje
    * svaki drugi put *, a inace praznina. Kako se ispisuje svaki
    drugi put
    * vrsi se provera j%2 == 0. */
21     for (j=0; j<2*i+1; j++)
        if (j==0 || j==2*i || (i==n-1 && j%2==0))
23         printf("*");
        else
25         printf(" ");
        printf("\n");
27     }
29     return 0;
}

```

```

#include <stdio.h>
2
int main()
4 {
    unsigned int n;
    int i, j;

6     printf("Unesite broj n: ");
    scanf("%u", &n);

8     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
    for (i=0; i<n; i++)
12     {
        /* Prvo se ispisuju beline koje prethode karakterima *. */
        for (j=0; j<n-i-1; j++)
14             printf(" ");
        /* Posle belina se ispisuje sam trougao. Ako je brojac na ivici
        onda
18         * se ispisuje karakter *, a inace praznina. Takodje, proverava
        se
        * da li se ispisuje poslednji red (i==n) i u njemu se ispisuje
        * svaki drugi put *, a inace praznina. Kako se ispisuje svaki
        drugi put
        * vrsi se provera j%2 == 0. */
22         for (j=0; j<2*i+1; j++)
            if (j==0 || j==2*i || (i==n-1 && j%2==0))
24             printf("*");
            else
26             printf(" ");
            printf("\n");
28     }

30     /*Sada se ispisuje donji trougao. Kako je prvi red donjeg trougla
    * vec ispisan (poslednji red gornjeg trougla), potrebno je
    naciniti jednu
32     * iteraciju manje. */

```

```

34     for (i=n-2; i>= 0;i--)
35     {
36         /* Prvo se ispisuju beline koje prethode karakterima *. */
37         for (j=0;j<n-i-1;j++)
38             printf(" ");
39         /* Posle belina se ispisuje potreban broj karaktera *. */
40         for (j=0;j<2*i+1;j++)
41             if (j==0 || j==2*i)
42                 printf("*");
43             else
44                 printf(" ");
45             printf("\n");
46     }
47
48     return 0;
49 }

```

Rešenje 0.1.56

```

1  #include <stdio.h>
2
3  int main()
4  {
5      unsigned int n, i, j;
6
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     /* Strelica se moze posmatrati kao spojena dva pravougla
11      * trougla kojima se ispisuje hipotenuza i jedna,
12      * donja kateta. */
13
14     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
15     for (i=0; i<n; i++)
16     {
17         for (j=0;j<=i;j++)
18             /* Proverava se da li se ispisuje karakter na
19              * hipotenuzi (j == i-1) ili da se ispisuje poslednji
20              * red (i == n-1). */
21             if (j == i || i == n-1)
22                 printf("*");
23             else
24                 printf(" ");
25             printf("\n");
26     }
27
28     /* Potrebno je iscrtati i donji deo slike, odnosno donji trougao.
29     * Brojac i odredjuje koji red donjeg trougla se trenutno iscrtava.
30     * Kako je prvi red donjeg trougla vec iscrtan (to je poslednji
31     * red gornjeg trougla), brojac se postavlja na 1. */
32     for (i=1; i<n; i++)

```

```

33     {
34         for (j=0; j<n-i; j++)
35             /* Provera da li se ispisuje hipotenuza. */
36             if (j == n-i - 1)
37                 printf("*");
38             else
39                 printf(" ");
40             printf("\n");
41     }
42
43     return 0;
44 }

```

Rešenje 0.1.57

```

1  #include <stdio.h>
2
3  int main()
4  {
5      unsigned int n;
6      int i, j, k;
7
8      printf("Unesite broj n: ");
9      scanf("%u", &n);
10
11     /* Brojac j odredjuje koliko ukupno karaktera (praznina i karaktera
12      * *)
13     * u svakom redu se ispisuje. U svakom drugom redu ovaj broj se
14     * povecava za 2. Na pocetku je 1 (jer se ispisuje samo jedna
15     * zvezda). */
16     j = 1;
17
18     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
19     for (i=1; i<=n; i++)
20     {
21         /* U svakom drugom redu broj ispisanih karaktera se uvecava za 2.
22          * */
23         if (i%2 == 0)
24             j+= 2;
25         for(k=0; k<j; k++)
26         {
27             /* U svakom drugom redu se naizmenicno ispisuje * ili praznina
28              * */
29             if (i%2 == 0)
30             {
31                 if (k%2 == 0)
32                     printf("*");
33                 else
34                     printf(" ");
35             }
36             else
37                 printf("*");
38         }
39     }
40 }

```

```

33         printf("\n");
35     }
37     return 0;
38 }

```

Rešenje 0.1.58

```

#include <stdio.h>

2 int main()
4 {
    unsigned int n, m;
    int i, j, k;

    printf("Unesite brojeve n i m: ");
    scanf("%u%u", &n, &m);

    for(i=1; i<=m; i++)
    {
        /* Za svaki kvadrat se racuna duzina bez poslednje ivice.
        * Kvadrat je sastavljen od (m-1) zvezdice i (m-1) praznine
        * (praznine se nalaze izmedju zvezdica).
        * Znaci ukupna duzina je 2*(m-1) karakter, a kako ima
        * n kvadrata, duzina je n*2*(m-1). */
        for(j=0; j<= n*2*(m - 1); j++)
            /* Provera da li se ispisuje prvi ili poslednji red. */
            if (i == 1 || i == m)
                /* Naizmenicno se ispisuje * i praznina. */
                if (j%2 == 0)
                    printf("*");
                else
                    printf(" ");
            else
                printf(" ");
        /* Na kraju svakog kvadrata (nakon svake (m-1) zvezdice i
        * (m-1) praznine se ispisuje ivica kvadrata. */
        if (j%(2*(m-1)) == 0)
            printf("*");
        else
            printf(" ");

        printf("\n");
    }

    return 0;
38 }

```

Rešenje 0.1.59

```

1  #include <stdio.h>

3  int main()
4  {
5      unsigned int n;
6      int i, j;

7
8      printf("Unesite broj n: ");
9      scanf("%u", &n);

11     /* Potrebno je spojiti sve slike u jednu, sliku gornjeg dela romba
12      * i sliku donjeg dela romba. */

13
14     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
15     for (i=0; i<n; i++)
16     {
17         /* Prvo se ispisuju * koje prethode karakterima -. */
18         for (j=0; j<n-i; j++)
19             printf("*");
20         /* Posle * se ispisuje potreban karakter -. */
21         for (j=0; j<2*i; j++)
22             printf("-");
23         /* Potom se ispisuju * koje su nakon karaktera -. */
24         for (j=0; j<n-i; j++)
25             printf("*");
26         printf("\n");
27     }

28
29     /*Sada se ispisuje donji trougao. Kako je prvi red donjeg trougla
30     * vec ispisan (poslednji red gornjeg trougla), potrebno je
31     * naciniti jednu
32     * iteraciju manje. */

33     /* Brojac i odredjuje koliko redova se ispisuje.
34     * Radi lakseg izracunavanja koliko zvezdica i praznina
35     * je potrebno ispisati u svakom redu, i se postavlja na
36     * n-1 i smanjuje u svakoj iteraciji petlje. */
37     for (i=n-2; i>= 0; i--)
38     {
39         /* Prvo se ispisuju * koje prethode karakterima -. */
40         for (j=0; j<n-i; j++)
41             printf("*");
42         /* Posle * se ispisuje potreban karakter -. */
43         for (j=0; j<2*i; j++)
44             printf("-");
45         /* Potom se ispisuju * koje su nakon karaktera -. */
46         for (j=0; j<n-i; j++)
47             printf("*");
48         printf("\n");
49     }

```

```
51 return 0;
}
```

Rešenje 0.1.60

```
1  #include <stdio.h>
3  int main()
4  {
5      unsigned int n, i, j;
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     /* Slika se sastoji iz dva dela, trougla i kvadrata
11        * i svaki deo se nezavisno iscrtava. */
12
13     /* Prvo se iscrtava krov, odnosno trougao. */
14     for (i=0; i<n-1; i++)
15     {
16         /* Prvo se ispisuju beline koje prethode karakterima *. */
17         for (j=0; j<n-i-1; j++)
18             printf(" ");
19         for (j=0; j<2*i+1; j++)
20             if (j==0 || j==2*i)
21                 printf("*");
22             else
23                 printf(" ");
24             printf("\n");
25     }
26
27     /* Potom se iscrtava kvadrat. Da bi iscrtavanje bilo lakse
28        * istovremeno se ispisuju dva karaktera. */
29     for (i=0; i<n; i++)
30     {
31         for (j=0; j<n; j++)
32             /* Provera da li je ivica. */
33             if (j ==0 || j==n-1 || i == 0 || i== n-1)
34                 printf("* ");
35             else
36                 printf(" ");
37             printf("\n");
38     }
39
40     return 0;
41 }
```

Rešenje 0.1.61


```

1  #include <stdio.h>

3  int main()
4  {
5      unsigned int n, i, j;

7      printf("Unesite broj n: ");
8      scanf("%u", &n);

9      for(i=1; i<=(n+1)/2; i++)
10     {
11         for(j=i; j<=n+1-i; j++)
12             printf("%d ", j);
13     }

15     return 0;
17 }

```

Rešenje 0.1.62

```

1  #include <stdio.h>

3  int main()
4  {
5      unsigned int n, i, j;

7      printf("Unesite broj n: ");
8      scanf("%u", &n);

9      for(i=1; i<=n; i++)
10     {
11         for(j=1; j<=n; j++)
12             if (j % i == 1 || i==1)
13                 printf("%d ", j);

15         printf("\n");
17     }

19     return 0;
21 }

```