

**Zadatak 1.** Napiši funkciju koji obračunava zaradu radnika. Radnici koji rade više od 40 sati nedeljno su plaćeni 1.5 puta više nego što bi bili plaćeni da rade do 40 sati nedeljno. Broj radnih sati po danima završene radne nedelje se učitava iz fajla. Funkcija kao parametar prima putanju do fajla sa radnim satima i cenu radnog sata i ispisuje imena i zarade radnika.

Primer izvršavanja programa:

Za fajl radnici.txt

```
pera|8|9|7|8|9
jova|8|8|7|8|7
steva|8|8|9|8|7
```

rezultat poziva funkcije treba da bude:

```
>>> racunanjeZarade("radnici.txt",1000)
ime: pera
zarada: 61500.0
ime: jova
zarada: 38000
ime: steva
zarada: 40000
```

**Zadatak 2.** Napiši funkciju koja prima broj bodova na testu, a vraća ocenu sa testa. Student na testu može da osvoji od 0 do 100 bodova. Ocenjivanje je dato u tabeli:

| <b>bodovi</b> | <b>ocena</b> |
|---------------|--------------|
| od 0 do 55    | 5            |
| od 56 do 65   | 6            |
| od 66 do 75   | 7            |
| od 76 do 85   | 8            |
| od 86 do 95   | 9            |
| od 96 do 100  | 10           |

Primer izvršavanja programa:

```
>>> ocenjivanje(77)
8
>>> ocenjivanje(95)
9
>>> ocenjivanje(96)
10
```

**Zadatak 3.** Indeks telesne mase se računa po sledećoj formuli  $BMI = \frac{m}{h^2}$  u kojoj je  $m$  masa u kilogramima, a  $h$  visina u metrima. U tabeli je data preporučena klasifikacija indeksa telesne mase:

| BMI       | Klasifikacija            |
|-----------|--------------------------|
| <18,5     | Pothranjenost            |
| 18,5 - 25 | Idealna telesna težina   |
| 25 - 30   | Preterana telesna težina |
| >30       | Gojaznost                |

Napiši funkciju koja prima težinu u kilogramima i visinu, a vraća kategoriju iz klasifikacije BMI.

Primer izvršavanja programa:

```
>>> print(indeksTelesneMase(55,1.8))
'pothranjenost'
>>> print(indeksTelesneMase(75,1.8))
'idealna telesna tezina'
>>> print(indeksTelesneMase(81,1.8))
'preterana telesna tezina'
>>> print(indeksTelesneMase(120,1.8))
'gojaznost'
```

**Zadatak 4.** Kazna za brzu vožnju računa se kao 5000 din + 500 din za svaki kilometar preko ograničenja + 10000 din za vožnju preko 120km/h. Napisati funkciju koja prima izmerenu brzinu vozila i ograničenje brzine. Ako je brzina veća od dozvoljene funkcija vraća poruku sa cenom kazne, a ako je manja vraća poruku da je sve u redu.

Primer izvršavanja programa:

```
>>> print(kazna(80,60))
vasa kazna iznosi 15000din
>>> print(kazna(50,60))
niste prekoracili brzinu
>>> print(kazna(130,60))
vasa kazna iznosi 50000din
```

**Zadatak 5.** Dadilja naplaćuje 150din po satu čuvanja dece do 9 sati uveče, a 100din po satu čuvanja dece posle 9 sati uveče. Napiši funkciju koja kao parametre prima vreme kada dadilja počinje da čuva deci i kada završava ućuvanje dece, a vraća poruku o zaradi koja treba se isplati. Vreme je zadato u formatu hh:mm, a predpostavlja se da se čuvanje dece odvija u periodu od 24h.

Primer izvršavanja programa:

```
>>> print(dadilja('18:35','22:50'))
zarada dadilje je 546din
```

**Zadatak 6.** Datum na koji pada uskrs u Gregorijanskom kalendaru za godine 1982–2048 se računa po formuli:

$a = \text{year} \% 19;$

$b = \text{year} \% 4;$

$c = \text{year} \% 7;$

$d = (19a + 24) \% 30;$

$e = (2b + 4c + 6d + 5) \% 7$

Datum na koji pada uskrs je  $22 + d + e$  mart (ako je vrednost veća od 31 onda je april). Napiši funkciju koja računa datum uskrsa. Funkcija kao parametar prima godinu i vraća poruku sa informacijom kada je uskrs ako je godina u zadatom opsegu, odnosno poruku da je došlo do greške ako godina nije u zadatom opsegu.

Primer izvršavanja programa:

```
>>> print(uskr(s(1994)))
uskrs je 29. marta 1994. godine
>>> print(uskr(s(2011)))
uskrs je 19. aprila 2011. godine
>>> print(uskr(s(1962)))
godina nije u predviđenom opsegu
```

**Zadatak 7.** Formula za izračunavanje datuma na koji pada uskrs u Gregorijanskom kalendaru data u prethodnom zadatku važi za sve godine od 1900 do 2099, osim za 1954., 1981., 2049. i 2076. Za te godine ova formula daje rezultat koji je nedelju dana kasnije nego što treba. Modifikuj zadatak 6 tako da važi za sve godine od 1900 do 2099.

**Zadatak 8.** Godina je prestupna ako je deljiva sa 4, osim ako je poslednja godina u veku, a tada je prestupna ako je deljiva sa 400 (na primer 1800 i 1900 nisu prestupne dok 1600 i 2000 jesu). Napisati funkciju koji proverava da li je godina prestupna.

Primer izvršavanja programa:

```
>>> print(prestupnaGodina(1983))
nije prestupna
>>> print(prestupnaGodina(1984))
prestupna
>>> print(prestupnaGodina(1900))
nije prestupna
>>> print(prestupnaGodina(2000))
prestupna
```

**Zadatak 9.** Napiši funkciju koji datum prima u obliku dd/mm/gggg i proverava da li je datum validan. Na primer 24/5/1962 je validan datu, ali 31/9/2000 nije jer septembar nema 31 dan. Takođe voditi računa i o tome da li je godina prestupna.

**Zadatak 10.** Napiši program koji prima datum u obliku dd/mm/gggg i računa redni broj dana u godini. Redni broj dana u godini se računa na sledeći način:

1. danUGodini =  $31(\text{mm} - 1) + \text{dd}$
2. ako je mm posle februara danUGodini umanji za  $(4\text{mm} + 23) / 10$
3. ako je prestupna godina i mm posle februara danUGodini uvećaj za 1