

PRIPREMA ZA TEST 1

NAPOMENE

- Svaki zadatak realizovati u zasebnom .py modulu
- Svaki zadatak treba da ima modul u kome se implementiraju funkcije i modul *main* u kome se zadatak testira

ZADACI

1. Sabrati brojeve od 1 pa nadalje dok zbir ne pređe broj *x*, gde je *x* parametar komandne linije. Ispisati konačan zbir i broj koji je poslednji sabran.

Primer:

Krajnji zbir: 250

Izlaz: Zbir iznosi 253, a poslednji sabran broj je 22.

Krajnji zbir: 20

Izlaz: Zbir iznosi 21, a poslednji sabran broj je 6.

2. Napisati funkciju koja računa faktorijel broja. Broj čiji faktorijel se računa se unosi sa tastature.
3. Napisati funkciju koja računa prosečnu temperaturu za nepoznat broj dana. Vrednosti za svaki dan se unose sa tastature i čuvaju u rečniku. Ključ rečnika je redni broj dana, a vrednost je vrednost temperature za određeni dan. Unos se prekida kada korisnik unese vrednost koja nije cifra. Na izlazu ispisati kreirani rečnik kao i izračunatu prosečnu temperaturu.

Primer:

Temperatura za 1. dan: 8

Temperatura za 2. dan: -2

Temperatura za 3. dan: 4

Temperatura za 4. dan: 0

Temperatura za 5. dan: 5

Temperatura za 6. dan: a

Prosečna temperatura iznosi 3.00 stepeni.

Rečnik: {1: 8, 2: -2, 3: 4, 4: 0, 5: 5}

4. Nasumično se bira broj između 1 i 100. Pogađati koji broj je izbran dok se ne pogodi broj. Na izlazu ispisati iz kog pokušaja je broj pogođen.

Primer izlaza:

Program je "zamislio" jedan broj. Pokušajte da ga pogodite!

Unesite Vaš broj: 50

Probajte sa manjim brojem.

Unesite Vaš broj: 25

Probajte sa većim brojem.

Unesite Vaš broj: 30

Probajte sa većim brojem.

Unesite Vaš broj: 43

Bravo, pogodili ste iz 4. pokušaja!

5. Nasumično se biraju dva broja u intervalu između 10 i 100. Ispisivati izraz zbira dva broja i broj tačnih i pogrešnih odgovora, kao i procenat uspešnosti. Korisnik unosi pretpostavljeni iznos zbira sa tastature. Kraj sabiranja obezbediti unosom karaktera koji nije cifra.

Primer:

$63 + 21 = 84$

Tačno!

$42 + 90 = 132$

Tačno!

$48 + 71 = 22$

Netačno!

$87 + 68 = 155$

Tačno!

$11 + 94 = 116$

Netačno!

$25 + 11 = 100$

Netačno!

$16 + 89 = a$

Tačno ste odgovorili na 3 pitanja, a pogrešno na 3 pitanja. Procenat uspešnosti iznosi 50.00%.

6. Implementirati funkciju *randomList* koja kao povratnu vrednost vraća listu nasumično odabranih celih brojeva u interval [1, 500]. Implementirati funkciju *divisibleBy* koja iz generisane liste vraća novu listu koja sadrži samo brojeve deljive sa X. Na izlazu ispisati polaznu listu i listu brojeva deljivih sa X. Dužina liste koju kreira funkcija *randomList* kao i vrednost X se prosleđuju kao parametri komandne linije.

Primer:

Dužina liste: 10

Broj X: 13

Izlaz: U listi: [15, 26, 98, 390, 402, 208, 47, 498, 196, 205], brojevi deljivi sa 13 su: [26, 390, 208].

7. Napisati funkciju koja kao povratnu vrednost vraća listu delioca broja X. Broj X se prosleđuje kao parametar komandne linije.
8. Napisati funkciju *ChooseNumbers* koja nasumično bira tri celobrojna broja iz opsega [0, 100]. Funkcija vraća logičku vrednost *True* ukoliko su tri izabrana broja ista, u suprotnom vraća *False*. Ispisati poruku koliko je biranja bilo potrebno dok se ne izaberu tri ista broja. U istom modulu napisati glavni deo programa koji testira funkcionalnost.
9. Korisnik unosi celobrojnu vrednost sa tastature dok se ne prosledi vrednost koja nije cifra. Napisati funkciju *CalculateSum* koja za uneti broj računa zbir njegovih cifara. U istom modulu napisati funkciju *IsPrime* koja proverava da li je uneti broj prost. Glavni deo programa koji testira funkcionalnost implementirati u istom modulu na sledeći način: uneti broj predstaviti kao ključ rečnika, a vrednosti koje funkcije *CalculateSum* i *IsPrime* vraćaju predstavljaju uređeni par (*tuple*) koji će biti vrednost u rečniku.

Primer:

Unete vrednosti: 3, 18, 205, a

Izlaz: {3: (3, True), 18: (9, False), 205: (7, False)}

10. Implementirati igricu *papir, kamen, makaze* za dva igrača na sledeći način:
Vrednosti se unose sa tastature, prvo prvi igrač bira, zatim drugi. Unete vrednosti čuvati u rečniku, gde je ključ id igrača (1 ili 2), a vrednosti je igračev odabir (papir, kamen ili makaze). Ispisati id pobednika kao i razlog pobeđe.
Igra se završava kada jedan od igrača unese vrednost *predaja*.
Potrebno je zaštititi program od nevalidnog unosa.

Primer:

Prvi igrač bira: *papir*

Drugi igrač bira: *kamen*

Izlaz: *Pobednik je prvi igrač jer papir pobeđuje kamen.*

Prvi igrač bira: *makaze*

Drugi igrač bira: *papir*

Izlaz: *Pobednik je prvi igrač jer makaze pobeđuju papir.*

Da se podsetimo pravila:

- **PAPIR** pobeđuje **KAMEN**
- **KAMEN** pobeđuje **MAKAZE**
- **MAKAZE** pobeđuju **PAPIR**

11. Napisati program koji na ekran ispisuje Paskalov trougao. Paskalov trougao predstavlja beskonačan niz prirodnih brojeva koji je u obliku piramide. Svaki broj u jednom redu predstavlja zbir brojeva koji su iznad njega. Krajnji brojevi su uvek jedinice. Korisnik sa tastature unosi visinu trougla.

Primer:

Uneta visina: 6

Izlaz:

```
  1
 1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```