

Prilikom uzimanja ulaznih podataka nemojte ispisivati nikakvu poruku tj. koristite funkciju `input()` sa praznim zagradama, bez argumenta. Prilikom ispisa rezultata potrebno je ispisati samo podatke, bez opisa (npr. ukoliko je rezultat koji se ispisuje 86 ne treba ispisivati `rezultat je: 86`, već samo 86)

1. (5 points) Elipsoid je zatvorena centralno-simetrična ploha drugoga reda. U središtu elipsoida sijeku se tri međusobno okomite ose (glavne ose) simetrije. Elipsoid se može parametarizirati na više načina, a najčešće korišten je:

$$x = a \sin(\theta) \cos(\varphi) \quad (1)$$

$$y = b \cos(\theta) \sin(\varphi) \quad (2)$$

$$z = c \sin(\theta) \quad (3)$$

Napisati program koji od korisnika traži parametare a , b , c , θ i φ , a program potom računa i ispisuje vrijednosti koordinata x , y , z . Primjer:

Input:

```
2.2
1.568
4.8
47.21
39.1
```

Output:

```
1.2529009763613002
0.6717726301427799
3.522472503385265
```

[Napomena: Potrebno je uzeti u obzir da će korisnik unijeti uglove u stepenima, a formula koristi radijane. Python sadrži funkciju `radians()` u biblioteci `math` koja kao parametar prima ugao u stepenima a vraća ga u radijanima.]

2. (5 points) Napisati program koji od korisnika zahtijeva unos cijelog broja. Program provjerava da li su cifre unesenog broja sa lijeva na desno u monotono opadajućem redoslijedu. Drugim riječima, program provjerava da li vrijedi da je $a_n \geq a_{n+1}$, gdje a predstavlja jednu cifru broja, a n njegovu poziciju.

Ukoliko su cifre unesenog broja u monotono opadajućem redoslijedu program ispisuje `jesu`, a u suprotnom program ispisuje `nisu`. Primjeri:

Input:

```
5442110
```

Output:

```
jesu
```

Input:

```
56
```

Output:

```
nisu
```

[Napomena: obratite pažnju na granični slučaj kada korisnik unese jednocifreni broj.]

3. (5 points) Napisati program koji simulira bacanje **dvije kockice**. Kockice imaju šest strana i na njima se nalaze brojevi od 1 do 6. Program treba izračunati, koliko je prosječno potrebno bacanja, kako bi se *tri puta za redom* dobio broj veći od 6.

Do odgovora treba doći simulirajući dobivanje traženog slučaja 10.000 puta. Prilikom pisanja programa potrebno je pozvati funkciju `random.seed()` i proslijediti joj vrijednost 42.