

## Domaći zadatak 2

1. U tajanstvenom svijetu postoji portal koji se otvara samo kada mu se date paran broj. Kao mladi čarobnjak na svom prvom zadatku, dobio si čarobni štapić koji može generisati brojeve. Vaš zadatak je da kreirate algoritam koji će provjeriti da li je broj koji je čarobni štapić generisao paran. Ako jeste, algoritam treba da ispiše: "Portal se otvara!" Ako nije, algoritam treba da ispiše: "Portal ostaje zatvoren."
2. U selu poznatom po svojim jabukama, održava se godišnje takmičenje u berbi jabuka između i najbliži pobjedi su Petar i Miloš. Petar tvrdi da je ubrao p jabuka, dok Miloš tvrdi da je ubrao m jabuka. Vaš zadatak je da kreirate algoritam koji će provjeriti da li je Petar uspio da ubere više jabuka nego Miloš i shodno tome ispiše poruku o pobjedniku. Pretpostaviti da ne mogu ubrati isti broj jabuka.
3. Zamislamo da pravimo program koji treba da odluči da li student može da pristupi ispitu. Postoje dva uslova: student mora imati više od 75% prisustva na predavanjima i mora imati predate sve seminarske radove. Oba uslova moraju biti zadovoljena da bi student mogao pristupiti ispitu. Algoritam treba da štampa odgovarajuću poruku. **Dodatak:** prisustvo se unosi u procentima, a dio za seminarske radove na sledeci nacin -> 0 predstavlja da bar jedan seminarski rad nije urađen, a 1 da su svi seminarski radovi urađeni.
4. Kućni red zabranjuje pravljenje buke prije 6 časova, između 13 i 17 časova i nakon 22 časa. Napiši program koji radnicima govori da li u nekom datom trenutku mogu da izvode bučnije radove.
5. Napisati program kojim se proverava da li se može napraviti bašta u obliku trougla sa datim dužinama stranica.
6. Vaš zadatak je da napravite program kojim proveravate da li se pčela kreće po žici. Žica se može predstaviti pravom  $y = 2 * x + 5$ , dok se pčela predstavlja tačkom (x, y).
7. Takmičari su radili testove iz matematike i programiranja. Za svaki predmet dobili su određeni broj poena (cio broj od 0 do 50). Takmičari se rangiraju po ukupnom broju poena iz oba predmeta. Ako dva takmičara imaju isti broj poena, pobjednik je onaj koji ima više poena iz programiranja. Potrebno je napisati program koji određuje pobjednika takmičenja.
8. Napisati program kojim se na osnovu datog prosjeka učenika prikazuje uspjeh učenika. Odličan uspjeh ima učenik čiji je prosjek veći ili jednak 4.5. Vrlodobar uspeh postiže učenik čiji je prosek veći ili jednak 3.5, a manji od 4.5, dobar uspeh se postiže za prosek koji je veći ili jednak 2.5 a manji od 3.5, dovoljan uspeh za prosek veći ili jednak 2, a manji od 2.5. Ako učenik ima neku jedinicu unijeće se prosjek 1, a uspeh mu je nedovoljan.

9. Kupili ste zavjesu pravouganog oblika. Provjerite da li će zavjesa prekriti prozor koji je takođe pravouganog oblika. Za zavjesu i prozor poznata je gornja lijeva i donja desna koordinata.
10. Vaš zadatak je da napišete program kojim provjerate da li je strelica pogodila pikado tablu. Za pikado tablu je poznat je njegov poluprečnik i koordinate centra, a za strelice koordinate cilja.
11. Vaš zadatak je da napišete program kojim ćete provjeriti da li se mrav kreće po ivici stola. Geometrijski, mrav se predstavlja kao tačka, a za sto su poznate tjemena desne gornje i lijeve donje ivice stola. Radi jednostavnosti smatrati da je sto pravouganik, ne kvadar.
12. Napisati program koji obrađuje dvocifreni broj na sledeći način:
- Ako je prva cifra veća od druge štampati razliku prve i druge cifre
  - Ako je prva cifra manja od druge štampati zbir te dvije cifre
  - Ako su cifre iste štampati njihov proizvod
13. Za dva stola kružnog oblika poznat je njihov poluprečnik. Napisati kod kojim se štampa obim stola sa većom površinom.
14. Date su cijene tri proizvoda. Naći par proizvoda čija cijena u zbiru daje najmanju vrijednost.
15. Napisati kod koji za datu godinu određuje da li je prestupna i štampa odgovarajuću poruku.
16. Napisati program kojim se provjerava da li se tačka nalazi unutar pravouganika. Za pravougaonik su date koordinate gornjeg lijevog i donjeg desnog tjemena.
17. Napisati program koji provjerava da li se od pravouganika poznatih dimenzija stranica mogu napraviti bar dva kvadrata čija je dužina ista kao i dužina pravouganika.
18. Napisati program kojim se na osnovu temperature vode određuje njeno agregatno stanje. Ako je temperatura:
- viša od 0 C i niža od 100C - agregatno stanje je tečno
  - ne viša od 0 C - agregatno stanje je čvrsto,
  - ne niža od 100 C - agregatno stanje je gasovito.
- Za temperaturu od tačno 0 smatra se da je agregatno stanje čvrsto, a za tačno 100 da je gasovito.
- Ulaz: Temperatura - cio broj
- Izlaz: Na standardni izlaz ispisati jednu od sledećih riječi: cvrsto, tecno, gasovito.

19. U jednoj privatnoj školi uvedeno je pravilo kojim se određuje iznos popusta koji ostvaruju učenici prilikom upisa u narednu školsku godinu. Učenici sa odličnim uspehom ostvaruju popust od 40% ukupnog iznosa školarine, sa vrlodobrim 20% a sa dobrim 10%. Takođe, učenici koji su osvojili nagradu na nekom od državnih takmičenja ostvaruju popust od 30% ukupnog iznosa školarine. Ukoliko neki učenik ispunjava dva kriterijuma za popust primenjuje se kriterijum po kome je popust veći. Na osnovu punog iznosa školarine, prosečne ocene učenika i informacije o nagradama sa takmičenja odrediti iznos koji učenik treba da plati pri upisu u narednu školsku godinu.

Ulaz: U prvoj varijabli nalazi se pun iznos školarine (realan broj), u drugoj prosječna ocjena učenika (realan broj od 2.0 do 5.0) a u trećoj 0 ukoliko učenik nema nagradu ili 1 ukoliko je ima.

Izlaz: Iznos školarine koju učenik treba da plati (zaokružen na najbliži cio broj) navodi se u jednoj linije standardnog izlaza.

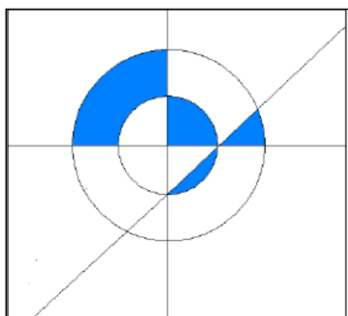
20. Napisati program koji računa zbir parnih cifara ukoliko je broj paran, a ukoliko je neparan proizvod neparnih cifara četvorocifrenog broja. Broj n unosi korisnik.

21. Napisati program koji za zadato x računa y i to na sledeći način:

$$y = \begin{cases} -2x + \frac{7}{2}, & x \leq -7 \\ \frac{x^2 - 3x + 5}{x^2 + 2}, & -7 < x < 1 \\ \sqrt{x^2 + 2x + 2} + \sqrt{\frac{3}{2}x - \frac{4}{7}}, & 1 \leq x \leq 8 \\ \left| \frac{3}{x^2} - 11x \right|, & x > 8 \end{cases}$$

22. Za unijetu tačku sa koordinata (x,y) provjeriti kom kvadratnu pripada u koordinatnom sistemu.

23. (\*) Napisati kod koji za date realne brojeve x i y provjerava da li tačka sa koordinatama (x,y) pripada osjenčenom dijelu ravni. Centar oba kruga je u tački (0,0), poluprečnici su im redom 4 i 6, dok je prava data jednačinom  $x - y - 4 = 0$ . Podsjetite se da je krug skup tačaka u ravni koje su na rastojanju r od date tačke tj. centra kruga. Štampati poruku „Pripada“ ili „Ne pripada“. Pomoć da li se tačka nalazi iznad ili ispod prave se nalazi na [linku](#).



24. Ako tekst ima više od 30 karaktera skratiti ga tako da ostane tačno 30 karaktera, a na kraj skraćenog teksta dodati ...
25. Napisati program kojim se uklanja prvi i poslednji karakter teksta i štampa novi tekst.
26. Napisati program koji provjerava da li je korsnik unio binarni, oktalni, dekadni ili heksadecimalni broj. Binarni broj ima prefiks 0b, oktalni 0o, heksadecimalni 0x, a dekadni nema prefiks.
27. Napisati program kojim provjeravati da li String sadrži bar jedan samoglasnik.
28. Napisati program koji provjerava da li se string završava sa target stringom.  
Primjer 1: string: "Abcd", target: "cd", štampa se "Da"  
Primjer 2: string: "www.google.com", target: "me", štampa se "Ne"
29. Napisati program koji provjerava da li je uneseni string binarni broj (ima samo 0 ili 1).
30. Napisati program koji računa zbir parnih i proizvod neparnih brojeva od 1 do n. Broj n unosi korisnik.  
a. Štampati taj zbir i proizvod.  
b. Štampati koliko ima parnih, a koliko neparnih brojeva iz tog segmenta.
31. Napisati program sa dva ulaza (korisnik unosi) start i end koji predstavljaju početak i kraj segmenta [start, end] (uključujući start i end), a koja kvadrira sve elemente iz tog segmenta koji su djeljivi sa 2 ali ne sa 4, a onda ih sumira. Štampati sumu.
32. Napisati program koji za unijete vrijednosti a, b, *djelilac* vraća sumu i broj elemenata djeljivih sa *djelilac* iz segmenta (a, b) (a i b ne pripadaju intervalu)
33. Napisati program koji sabira sve cifre unijetog broja.
34. Napisati program koji iz teksta izvlači cifre i računa njihov proizvod.
35. Napisati program koji vraća broj cifara u stringu i kreira od njih integer. Primjer: "Hi Mr. Rober53. How are you today? Today is 08.10.2019", štampa 5308102019 i to kao integer. Pomoć: da provjerite da li je karakter broj koristiti isdigit metod.
36. Kreirati program koji unijeti string s (karakter stringa alfabetski karakteri, mala slova) enkriptuje na sledeći način: ako je karakter suglasnik pretvara ga u 0, a ako je karakter samoglasnik pretvara ga u 1. Npr. za s = 'abaae' rezultat je 10111.

37. Igrač nasumično bira listicu na kojoj se nalazi tekst sastavljen od karaktera 1, 0 i \*. Karakter 1 nosi 1 poen, 0 nosi 0 poena, dok \* nosi -1 poen. Napisati program koji provjerava da li je igrač u plusu.
38. Napisati program koji za unijeti string s (provjeriti da li je karakter cifra) enkriptuje na sledeći način: ako je karakter paran broj pretvara se u 0, a ako je karakter neparan broj pretvara se u 1. Npr. za s = '15023' rezultat je 11001.
39. **Narcissistic Number** je broj čija suma cifara (tog broja) stepenova sa njegovim brojem cifara daje isti taj broj.  
Primjer 1: 153 (3 cifre)  
 $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$   
  
Primjer 2: 1634 (4 cifre):  
 $1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 256 = 1634$   
Vaš program treba da štampa "Da" ili "Ne" u zavisnosti od toga da li je broj Narcissistic ili nije. Input je uvijek validan broj.
40. Napisati program koji na osnovu niza cijelih brojeva računa apsolutnu sumu svih negativnih parnih elemenata za unijeti niz. Štampati sumu.  
**Primjer:**  
Input: [-2, 7, -5, 3, 1, -4]  
Output: 6 ( $|-2| + |-4|$ )
41. Napisati program koji za unijetu listu L i vrijednost max vraća broj elemenata koji su manji od max iz te liste. Napomena: lista sadrži samo cijele brojeve  
Input: a = [1,2,3], max = 3; Output: 2  
Input: a = [-1, 0, 5], max = -2; Output: 0
42. Dat je niz koji sadrži cijene proizvoda u jednom marketu. Market je za ovu nedelju odlučio da spusti cijene svim proizvodima. Kolika će zarada marketa od tih proizvoda biti manja u odnosu na originalnu cijenu.
43. Data je lista ocjena na predmetu likovno za sve učenike jednog odjeljenja osnovne škole. Ispostavilo se da nema učenika koji imaju ocjenu 1 i 2. Prebrojati koliko učenika ima ostale ocjene (za svaku ocjenu pojedinačno).
44. Data je lista koja sadrži broj posjeta za poslednjih deset fudbalskih utakmica. Napisati program koji štampa koliko je bilo najviše posjeta u jednom danu.
45. Kreirati program koji prikazuje koliko ima zaposlenih koji imaju veće plate od prosječne plata. Npr. ako su plate = [500, 600, 700] rezultat je 1 jer je samo plata od 700 EUR iznad prosječne plate.

46. Kreirati program koji nalazi platu zaposlenog koji ima drugo najveće primanje. Npr. ako je plate = [540,690, 900] rezultat je 690. Napomena: lista ima bar 2 elementa.
47. Korisnik unosi tri broja. Naći minimum i maksimum među unijetim brojevima i rezultat prikazati korisniku.
48. Napisati program koji račun  $X^n$  koristeći petlju (bez ugrađenog Python operatora za stepenovanje)
49. Napisati program koji skraćuje string do unijete dužine. Na kraj stringa dodati ... Ako je unijeza dužina veća od same dužine stringa, na kraju stringa dodati samo ...  
Primjer 1:  
string: "abcde", dužina: 2, štampa se "ab..."  
Primjer 2:  
string: "abcde", dužina: 10, štampa se "abcde..."
50. Napisati program koji iz zadatog stringa izdvaja samo samoglasnike i vraća taj novi string.
51. Lozinka je jaka ako je njena dužina najmanje 8 simbola, i sadrži mala slova, velika slova i cifre. Napisati program koji provjerava da li je lozinka jaka. Ulaz: Unosi se jedna riječ, dužine ne veće od 100, koja sadrži mala i velika slova i cifre. Izlaz: Štampati YES ili NO.
52. Napisati program koji na osnovu varijabli a, pre, sub i num dodaje prefiks pre, i sufiks suf stringu a num puta i vraća novi prošireni string.  
Input 1: a = 'test', pre = 'pr', suf = 'su', num = 2;  
Output 1: 'prprtestsusu'
53. Fudbal – Petar je posmatrao fudbalsku utakmicu i na papiru zapisivao rezultat sa semafora poslije svakog gola. Npr. mogući zapis je: 1:0, 1:1, 1:2, 2:2, 2:3. Zatim je Petar sabrao sve zapisane brojeve:  $1+0+1+1+1+2+2+2+2+3=15$ . Na osnovu datog zbira, napišite program koji određuje koliko je golova bilo na utakmici. Ulaz: U jednom redu dat je cio broj N – Petrov zbir ( $1 \leq N \leq 1000$ ). Izlaz: Štampati jedan cio broj – broj golova.
54. Dat je string sastavljen od karaktera 0 i 1. Karakter 0 predstavlja slobodno polje, a 1 predstavlja zauzeto polje. Vaš zadatak je da za zadatu poziciju u stringu provjerite da li su susjedna polja slobodna (lijevo i desno). **Napomena:** za prvo polje gledate same desno polje, za poslednje polje samo lijevo polje, a za ostala i lijevo i desno polje. Npr. ako je string 01010, a zadata pozicija 2 (indeksiranje kreće od nule), treba štampati 0 jer nema slobodnih polja.

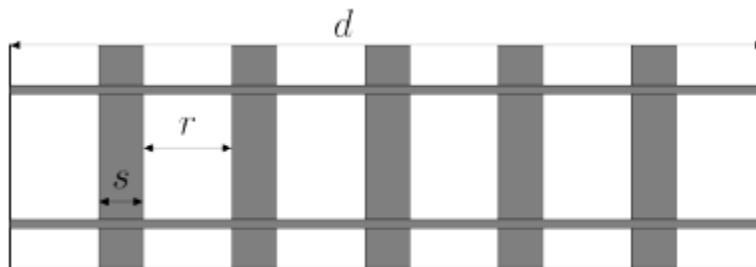
55. Napisati program koji za 2 data prirodna broja h i o koji redom predstavljaju broj molekula vodonika (H) i kiseonika (O), vraća koliko se najviše molekula vode (H<sub>2</sub>O) može dobiti od datih molekula. Npr., ako je h=4, o=3 odgovor je 2.
56. Napisati program kojim za unijeti string provjeravate koliko ima jednocifrenih negativnih brojeva. String se sastoji od negativnih i pozitivnih brojeva i oznaka za negativne (-) i pozitivne (+). Primjer: +23-2-32+4-22-4 izlaz je 2.
57. Narcissistic Number je broj čija suma cifara (tog broja) stepenova sa njegovim brojem cifara daje isti taj broj.  
Primjer 1: 153 (3 cifre)  
 $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$   
Primjer 2: 1634 (4 cifre):  
 $1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 256 = 1634$   
Vaš program treba da štampa "Da" ili "Ne" u zavisnosti od toga da li je broj Narcissistic ili nije. Input je uvijek validan broj.
58. Napisati program koji od zadatog stringa kreira novi string koji se sastoji bez cifara.  
Primjer: "Hi Mr. Rober53. How are you today? Today is 08.10.2019"), vraća "Hi Mr. Rober. How are you today? Today is .." kao string.  
Pomoć: da provjerite da li je karakter slovo koristiti isalpha metod, a da li je cifre koristite isdigit.
59. Napisati program koji za unijeti string s (karakter stringa cifre od 0 do 9) enkriptuje na sledeći način: ako je karakter paran broj pretvara se u 0, a ako je karakter neparan broj pretvara se u 1. Npr. za s = '15023' rezultat je 11001. Pomoć: Inicijalna vrijednost za dodatni string je "", a onda se pomoću operatora + nadodaje 0 ili 1 u zavisnosti u ispunjenosti uslova.
60. Napisati program koji na osnovu dvije varijable start i end koji predstavljaju početak i kraj segmenta [start, end] (uključujući start i end) kvadrira sve elemente iz tog segmenta koji su djeljivi sa 3 ali ne sa 6, a onda ih sumira. Štampati sumu.
61. Napisati program koji vraća velika slova iz zadatog unijetog teksta kao jedan novi string.  
Ulaz: "Prva recenica. Ovo je druga recenica. Na kraju treća."  
Izlaz: PON
62. Napisati program koji za unijeti string a prebrojava koliko ima heksadecimalnih brojeva. Broj je heksadecimalni ako ima prefiks 0x. Svaki broj je razdvojena razmakom (space)  
  
Ulaz 1: a = "12 0x1A 0001 121 0x2";  
Ulaz 2: a = "12 001 31";  
  
Izlaz 1: 2  
Izlaz 2: 0
63. Za dati string koji sadrži praznine (blankove), odrediti najdužu riječ u stringu.

64. Napisati program koji računa zbir najmanje i najveće cifre unesenog broja.

65. Na terasi dužine  $d$  metara treba rasporediti  $n$  stubića širine  $s$  centimetara tako da rastojanje između stubića, kao i između stubića i zida bude isto.

Ulaz: Tri reda standardnog ulaza sadrže tri broja:

- $d$  - realan broj koji predstavlja dužinu terase u m
- $n$  - broj stubića
- $s$  - realan broj koji predstavlja širinu stubića u cm



Izlaz: Rastojanje između stubića u centimetrima, zaokruženo na dve decimale.

**Napomena:** Neka je  $r$  nepoznato rastojanje između stubića. Tada  $n$  stubića pokriva  $n \cdot s$  cm. Između  $n$  stubića postoji  $n - 1$  razmak, jedan razmak je između lijevog zida i prvog stubića i jedan razmak je između poslednjeg stubića i desnog zida. Dužina terase u centimetrima je  $d \cdot 100$  cm.

Dakle, važi uslov  $n \cdot s \text{ cm} + (n + 1) \cdot r \text{ cm} = d \cdot 100 \text{ cm}$

Samo ostaje da  $r$  izracunamo i to je to.

66. Data je lista koja se sastoji od cijelih brojeva. Provjeriti da li u listi ima više dvocifrenih ili trocifrenih brojeva.

67. Napisati program koji provjerava koliko se određeni broj ponavlja u listi (taj broj unosi korisnik).

68. Napisati program koji uvećava zarade koje su veće od prosječne zarade (prosjeak liste) za  $X$  eura.

69. Napisati program koji umanjuje zarade koje su veće od prosječne za 10 %, a zarade manje od prosječne uvećava za 10 %. Prikazati koliko zarada će biti iznad prosjeka nakon uvećanja/umanjenja.

70. Napisati program koji vraća zbir kvadrata elemenata liste koji su djeljivi sa 3.

71. Kreirati program koji analizira zadatu listu brojeva i određuje koliko među njima postoji onih brojeva koji nakon primjene kvadratnog korijena zadržavaju svojstvo da budu cijeli brojevi. Program treba da prikaže ukupan broj takvih brojeva u listi.



72. Lista se sastoji od ocjena studenata na predmetu *Ekonomija i razvoj*. Koliko studenata je dobilo veću ocjenu od prosječne ocjene (ocjena 5 ne utiče na prosjek).
73. Za RPG igru, napravi program koji simulira inventar igrača. Inventar može biti lista tkz. *itema*, odnosno predmeta (npr. mač, šešir, rukavice,...). Vaš zadatak je da na osnovu pozicije u inventaru otkrijete koji je to predmet (za pristup koristiti indeks liste)
74. Napisati program koji za zadatu listu plata zaposlenih u jednoj IT kompaniji (plate se unose u eurima) računa prosječnu vrijednost plata u dolarima ako je poznato da je  $1\text{e} = 1.1\$$ .
75. Napisati program koji izračunava ukupni gubitak banke za godinu dana od kamate na štednju. Korisnik unosi listu početnih iznosa štednje klijenata. Takođe, korisnik unosi fiksnu kamatnu stopu na štednju (za sve korisnike ista). Program izračunava i ispisuje ukupni gubitak banke od kamate nakon datog perioda.
76. Napraviti program koji mijenja svako pojavljivanje određenog elementa u listi sa drugim elementom. Na primjer, zamjena svih 2 sa -2. Npr. lista [1, 2, 3, 2, 4, 2] u [1, -2, 3, -2, 4, -2]
77. Kreirati program koji provjerava da li je lista brojeva sortirana u rastućem poretку. Program treba da vrati True ako je lista sortirana, inače False.
78. Neka je data lista artikala u jednoj prodavnici. Pronaći vrijednost drugog najskupljeg proizvoda u prodavnici.
79. Napisati program koji za unijetu listu elemenata (cijeli brojevi, bez nule) vraća broj elemenata koji imaju suprotnu vrijednost. Broju  $x$ , suprotan broj je  $-x$ . Smatrati da se broj pojavljuje jednom i da nekad ima svoju suprotnu vrijednost, a nekad nema (ne može imati više od jedne suprotne vrijednosti, niti se ponavljati)
80. U igri gdje lik skače po platformama, dužina svakog skoka sačuvana je kao broj u listi. Napravi program koja nalazi razliku između najdužeg i najkraćeg skoka.
81. Napisati program koji analizira promjene cijena akcija. Program treba da učitа niz cijena akcija za određeni period, a zatim izračuna i ispiše najveći pad i najveći porast cijena za susjedne vrijednosti.
82. Pretpostavite da imate listu koja predstavlja broj slobodnih sjedišta u svakom redu u pozorištu. Na primjer, ako pozorište ima 5 redova, lista može izgledati ovako: [10, 8, 15, 12, 7], gdje svaki broj predstavlja broj slobodnih sjedišta u tom redu (listu unosi korisnik). Napisati program koji pronalazi najbolji red za grupu od  $n$  osoba koje mogu sjesti

zajedno (n unosi korisnik). Smatrati da je najbolji onaj red u kome ostane najviše slobodnih mjesta nakon što se smjesti n osoba.

83. Nakon napornog perioda studiranja odlučili ste da odete na kratko putovanje. Na sajtu *putovanja.me* postavljene su razne destinacije sa cijenama. Izbor je velik ali vaš budžet je ograničen. Odlučili ste da potrošite gotovo sav novac kojim raspolazete na ovo putovanje, pa ste odabrali onu destinaciju koja je najskuplja ali u skladu sa mogućnostima. Cijene destinacija se čuvaju u listi. Koliko budžeta vam ostaje na raspolaganju nakon uloženog novca.
84. Pretpostavite da imate restoran sa različitim stolovima različitih kapaciteta (npr. [4, 6, 2, 8, 5] mjesta po stolu). Večeras imate najavu da ćete imati više gostiju nego slobodnih mjesta u restoranu (unosite broj gostiju). Napišite program kojim računate koliko stolova je potrebno dodatno nabaviti ako se zna da uz svaki novi sto dolaze 4 stolice.
85. Napisati program koji za listu koja pokazuje broj slobodnih sjedišta u svakom redu bioskopa određuje koliko redova je potrebno rezervirati za grupu od n posjetilaca. Idealno je zauzeti što manje redova.
86. Napisati program **absolute\_of\_even\_sum** koji ima jedan parametar, niz cijelih brojeva, a koja računa apsolutnu sumu svih negativnih parnih elemenata za unijeti niz. Štampati sumu.  
Input: [-2, 7, -5, 3, 1, -4] Output: 6 ( $|-2| + |-4|$ )
87. Napisati program presjek (a, b) koji za unijete liste a i b vraća listu zajedničkih elementa liste a i liste b. Elementi liste a i liste b su brojevi ili stringovi.  
Input 1: a = [1, 2, 'a'], b = ['a', 2] Output 1: [2, 'a']  
Input 2: a = [2, 3, 4], b = [1, 1, 7] Output 2: []
88. Napisati program br\_elementa(a, max) koji za unijetu listu a vraća broj elemenata koji su manji od max iz te liste. Napomena: lista sadrži samo cijele brojeve.  
Input 1: a = [1, 2, 3], max = 3 Output 1: 2  
Input 2: a = [2, 3, 4], max = 7 Output 2: 3  
Input 3: a = [-1, 0, 5], max = -2 Output 3: 0
89. Napisati program br\_elementa(a) koji za unijetu listu elemenata (cijeli brojevi, bez nule) a vraća broj elemenata koji imaju suprotnu vrijednost. Broju x, suprotan broj je -x. Smatrati da se broj pojavljuje jednom i da nekad ima svoju suprotnu vrijednost, a nekad nema. Input 1: [1, 2, -1, 3, -3] Output: 2 Input 2: [20, 10, -10, 100] Output: 1
90. Napisati program update\_list(a, x) koji za unijetu listu elemenata (cijeli brojevi) a uvećava svaki parni element za vrijednost x. Parametar x je prirodan broj.  
Input 1: [1, 2, -1, 3, -4], 3 Output: [1, 5, -1, 3, -1]  
Input 2: [21, 10, -10, 100], 5 Output: [21, 15, -5, 105]

91. Napisati program `second_max(a)` koji nalazi drugi najveći element liste `a`. Npr. ako je `a = [1, 22, 33]` rezultat je 22. Napomena: lista ima bar 2 elementa.
92. Napisati program koji omogućava korisniku da unese `n` proizvoda (`n` parametar funkcije). Svaki proizvod se predstavlja sa kao dictionary oblika: `{naziv, opis, cijena, broj_artikala}`. Funkcija treba da vrati listu proizvoda. Nakon toga, napisati funkciju koja ima dva parametra i to proizvodi, `search_term`, a koja vraća sve proizvode čiji naziv počinje sa vrijednošću parametra `search_term`.
93. Napisati program koji kao parametar prima listu igrice oblika (`ime:String, izdavac:String, godina_izlaska:Integer, ocjena:Float`), a vraća igrice čija je ocjena veća od ocjene `x` (unos korisnik sa input), a čije je izdavac `y` (unos korisnik sa input). Koristiti list comprehension pristup.
94. Napisati program koji za unijeti string i slovo vraća sve riječi čija je dužina paran broj, a ne sadrže zadato slovo (koje korisnik unosi). Primjer: `get_ewfbyr("words with even number of letters without character d", "d")` Output: `["with", "even", "number", "of"]`
95. Napisati program `longest_increasing` koji ima jedan parametar i to `input_list` (predstavlja listu cijelih brojeva), a nalazi najdužu neopadajuću podlistu pozitivnih cijelih brojeva (brojevi veći od 0) i vraća tu podlistu  
Primjer:  
Input: `[1, 2, 3, -1, 0, 5, 6, 7, 10, 10, 1]`  
Output: `[5, 6, 7, 10, 10]`
96. Napisati program koji ima dva parametra string i number gdje prvi parametar predstavlja ulazni string, dok number predstavlja broj na osnovu koga se radi razbijanje stringa na podstringove. Funkcija treba da vrati niz/listu podstringova zadate dužine. Ako posljednji podstring ne sadrži dovoljno karaktera dopuniti ga sa `*`. Napomena: space se također računa kao karakter.  
Primjer 1:  
`split_string("danas polažemo test", 5) -> ["danas", " pola", "žemo ", "test*"]`  
Primjer 2:  
`split_string("kurs web program.", 6) -> ["kurs w", "eb pro", "gram.*"]`  
Primjer 3:  
`split_string("da", 7) -> ["da*****"]`
97. Napisati program koji omogućava korisniku da unese `n` proizvoda (`n` parametar funkcije). Svaki proizvod se predstavlja sa kao dictionary oblika: `{naziv, opis, cijena, broj_artikala}`. Funkcija treba da vrati listu proizvoda. Nakon toga, napisati funkciju koja ima tri parametra i to proizvodi, naziv i `raspolozivi_novac`, a koja vraća koliko proizvoda (naziv proizvoda se zadaje parametrom naziv) kupac može da trguje. Ako je broj proizvoda koje kupac može da kupi veći od vrijednosti atributa `broj_artikala` za konkretan proizvod,

onda treba vratiti samo broj\_artikala (kupac ne može da kupi više komada određenog proizvoda).

98. Klijenti postavljaju zahtjeve brokeru za kupovinu/prodaju akcija. Zahtjevi mogu da budu jednostavni ili višestruki (više jednostavnih). Zahtjev ima sledeći format: Quote /space/ Quantity /space/ Price /space/ Status gdje Quote predstavlja naziv akcije, sadrži non-whitespace karaktere, Quantity je prirodan broj koji predstavlja broj akcija koje se prodaju/kupuju, Price je float koji predstavlja cijenu pojedine akcije (sa decimalnom tačkom "." ), Status je B (buy) ili S (sell) koji predstavlja da li se akcije prodaju ili kupuju.

Primjer 1 (simple): "GOOG 300 542.0 B"

Višestruki zahtjevi se sastoje od više simple zahtjeva koji su spojeni zarezom

Primjer 2 (multiple-višestruki):

"ZNG 1300 2.66 B,NY 50 56.32 B,OWW 1000 11.623 B,OGG 20 580.1 B"

Da olakšate brokeru posao vaš zadatak je da mu vratite string "Buy: b Sell: s", gdje su b i s formata float zaokruženog na 2. decimale, b predstavlja ukupnu cijenukupljenih akcija, a s ukupnu cijenu prodatih akcija.

Output za primjer 2:

"Buy: 29499.00 Sell: 0"

99. Napisati funkciju split\_string koja ima dva parametra string i number gdje prvi parametar predstavlja ulazni string, dok number predstavlja broj na osnovu koga se radi razbijanje stringa. Funkcija treba da vrati niz/listu podstringova zadate dužine.

Primjer 1: split\_string("danas polažemo test", 5) -> ["danas", " pola", "žemo ", "test"]

Primjer 2: split\_string("kurs web program.", 6) -> ["kurs w", "eb pro", "gram."] (slovo nj se posmatra kao dva karaktera)

Primjer 3: split\_string("da", 7) -> ["da"]

100. Vaš zadatak je da napravite password validator. Ovaj validator treba da funkcioniše za razne slučajeve i to na osnovu toga kako definišete određene parametre programa. Parametri koji mogu da budu True ili False su:

- flagUpper kojim provjeravate da li string ima ili ne bar jedno veliko slovo
- flagLower kojim provjeravate da li string ima ili ne bar jedno malo slovo
- flagDigit kojim provjeravate da li string sadrži bar jednu cifru

Osim ova tri parametra (koji su postavljeni na False ako se ne definišu), treba proslijediti i minimalnu dužinu stringa koja mora biti zadovoljena, kao i sami string koji validirate. Funkciju treba da izgleda check\_password (input\_string, min\_string\_len, flagUpper, flagLower, flagDigit)

Primjer:

input\_string = "Passw123", output: check\_password (input\_string, 10, True, True, False) vrati False jer smo stavili da je minimalna dužina stringa 10, a u našem primjeru je 8, dok check\_password(input\_string, 8, True, True, False) -> True

101. Milošu je ova nedelja bila teška na poslu i bilo je naporno ujutro ustati iz kreveta. Dok čeka autobus na autobusnom stajalištu Miloš je shvatio da ako uspije odložiti dolazak do poslednjeg časa može dobiti dragocjene dodatne minute za spavanje. Do Miloševe adrese saobraća autobus koji ide do njegove kancelarije na svakih 15 minuta, prvi autobus je u 06:00, a poslednji je u 00:00. S obzirom na to da Milošu treba 5 minuta hoda od kuće do autobusnog stajališta, implementirajte funkciju koja će Milošu dati vrijeme (u minutima) za spavanje te će mu preostati još vremena prije nego što krenete. Napomena: Miloš mora uhvatiti prvi autobus.

Input 1: "05:00" Output: 55

Input 2: "10:00" Output: 10

Input 3: "12:10" Output: 0

Input 4: "12:11" Output: 14

102. Svakog jutra sva vrata škole su zatvorena. Škola je prilično velika, ima N vrata. Učenici su počeli da dolaze. Teško je za povjerovati, ali svi oni žele da uče. Škola ima N učenika, a oni dolaze jedno po jedno. Kada dijete prođe kroz vrata, ono izmijeni status za vrata (Open-> Closed, Closed-> Opened). Svaki učenik ima svoj broj, i svaki i-ti mijenja status i-tim vratima. Na primjer: kada prvi učenik dođe u školu, on mijenja status svim prvim vratima (otvara ih sve). Drugi mijenja status za svaka druga vrata (druga, četvrta, šesta, itd.). Treći mijenja status za svaka treća vrata (treća, šesta, itd.). Konačno, zadnji učenik (n-ti), mijenja status za svaka n-ta vrata (samo su jedna takva, zadnja). Vaš zadatak je da izračunate koliko vrata će ostati otvoreno nakon što dođu svi učenici. Primjer:

	Doors				
	1	2	3	4	5
Initial State	Red	Red	Red	Red	Red
After the 1st	Green	Green	Green	Green	Green
After the 2nd	Green	Red	Green	Red	Green
After the 3rd	Green	Red	Red	Red	Green
After the 4th	Green	Red	Red	Green	Green
After the 5th	Green	Red	Red	Green	Red

Crveni kvadrati – zatvorena vrata, zeleni – otvorena vrata.

Input: n – broj vrata i učenika,  $n \in \mathbb{N}$ ,  $n \in [1, 100000]$

Output: o – broj otvorenih vrata,  $o \in \mathbb{N}$  doors(5) treba da vrati 2