Slice – lista[START : STOP : STEP]

- Način kreiranja nove liste na osnovu manipulacije elementima prethodno kreirane liste. Primjer:
  - izdvojiti zadnja dva elementa liste
  - izdvojiti svaki treći element liste
- Po sintaksi jako slično range() naredbi.

- Primjeri na listi brojeva od 1 do 100.

ALGEBRA

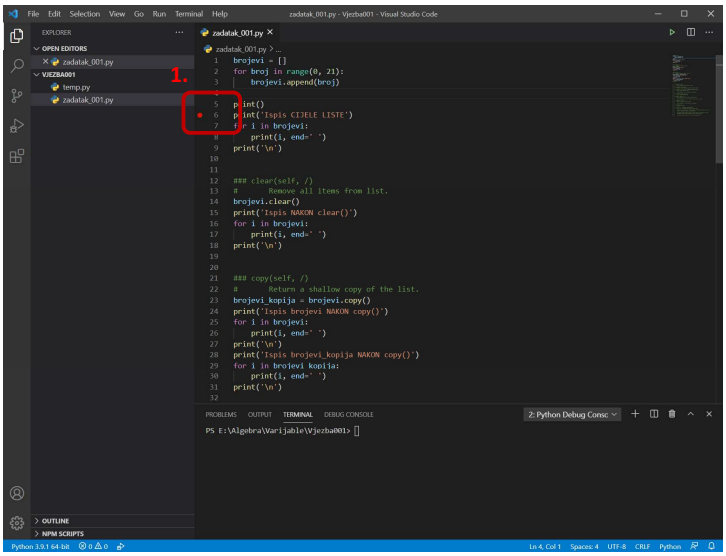
Lista – ostale naredbe

- *naziv_liste**.clear()*** – naredba za brisanje svih elemenata liste
- *nova_lista* = *naziv_liste**.copy()*** – naredba za kopiranje liste
- **PROBLEM!!! .copy() NE kopira listu!**

ALGEBRA

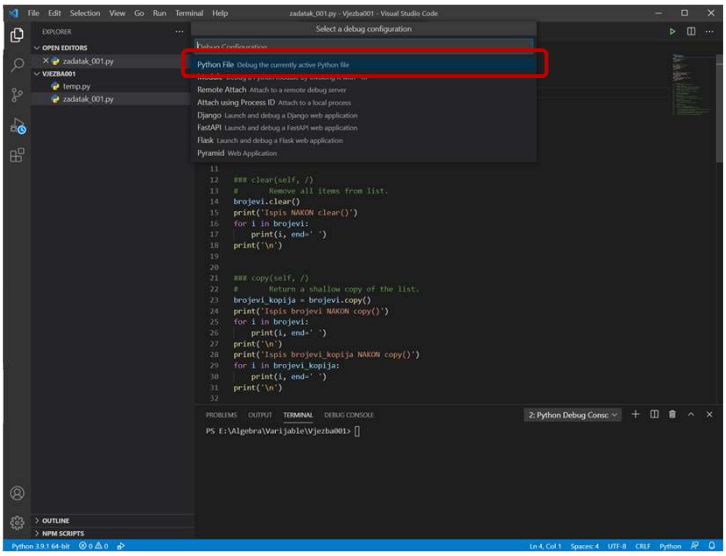
Debugging

1. Kliknite na stupac lijevo od linije kôda. U našem primjeru to je linija 6. Kada kliknete na taj stupac ispred te linije će se pojaviti točka.
   Ta točka se zove Break Point i to je točka u kojoj će se ZAUSTAVITI izvršavanje programa.
2. Pokrenite program u DEBUGGING načinu rada jednostavno tipkom F5 na tipkovnici ili iz izbornika
   Run -> Start Debugging



Debugging

1. Nakon klika na F5, VS Code će vam ponuditi što želite debugirati?
2. Odaberite "**Python File** Debug the currently active Python file"

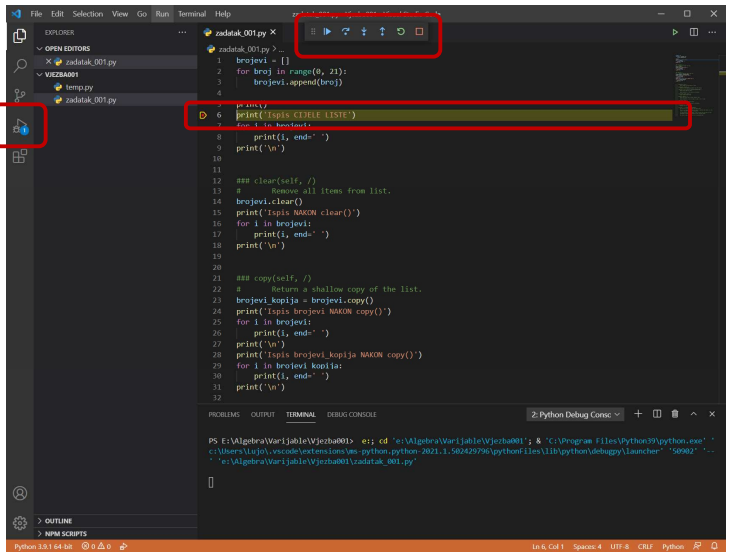


Debugging

1. Prvo kliknite na ikonu koja prikazuje aktivne programe.

Ostali crveni okviri prikazuju:

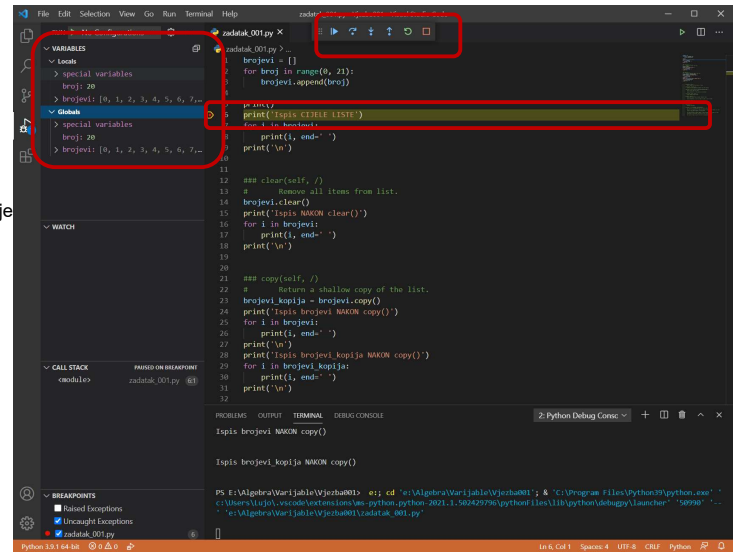
- Ikone za upravljanje pokretanjem programa. Opis na slijedećem slideu.
- Liniju u kojoj je program zaustavljen, a kada se pomoću upravljačkih ikona izvršavanje programa pokrene na slijedeću liniju, onda će ta slijedeća linija biti označena.
  - VAŽNO: Kôd u označenoj liniji NIJE se izvršio. To znači da je izvršavanje programa ZAUSTAVLJENO na POČETKU te linije.



Debugging

- Ikone za upravljanje pokretanjem programa. Opis s lijeva na desno:
  - Ikona s točkicama je za pozicioniranje ovog okvira - zanemarite
  - Continue (F5) – nastavi izvršavanje programa do kraja.
  - **Step Over (F10)** – ovo je ikona koju ćete najčešće koristiti. Pokreće izvršavanje programa red po red.
  - Step Into (F11)
  - Step Out (Shift + F11)
  - Restart (Ctrl + Shift + F5)
  - Stop (Shift + F5)

Stupac VARIABLES prikazuje trenutnu vrijednost lokalnih i globalnih varijabli. Ako nemate prikaz kao na slici, raširite prikaz klikom na > znak.



Lista – ostale naredbe

- *naziv_liste*.***clear()*** – naredba za brisanje svih elemenata liste
- *nova_lista* = *naziv_liste*.***copy()*** – naredba za kopiranje liste
- *broj_ponavljanja_u_listi* = *naziv_liste*.***count(element)*** – naredba koja prebrojava koliko se puta element pojavljuje u listi
- *naziv_liste*.***extend(nova_lista)*** – naredba koja proširuje postojeću listu novom listom
- *indeks_elementa* = *naziv_liste*.***index(element)*** – naredba koja dohvaća indeks pozicije na kojoj se nalazi element
- *naziv_liste*.***insert(indeks, element)*** – naredba za umetanje elementa na poziciju točno ispred pozicije označene navedenim indeksom
- *naziv_liste*.***sort()*** – naredba za sortiranje elemenata liste
- *naziv_liste*.***reverse()*** – naredba za sortiranje elemenata liste obrnutim redoslijedom

Zadatak – lista

- Napišite program koji kreira akorde na osnovu početnog tona, odnosno note.
  - POJAŠNJENJE
  - Akord se sastoji od tri tona koji se mogu ponavljati.
  - Durski akord čine: početni ton, 4. ton te 7. ton. Označava se samo velikim slovom početnog tona ili velikim slovom početnog tona uz dodatak dur
  - Molski akord čine: početni ton, 3. ton te 7. ton. Označava se samo malim slovom početnog tona ili malim slovom početnog tona uz dodatak mol
  - Glazbena abeceda počinje od C: C, C#, D, D#, E, F, F#, G, G#, A, A#, H
  - Engleska oznaka za H ton je B tako da oni imaju A B C D E F G tonove
  - Postoji pojašnjenje u teoriji glazbe zašto je prvi ton C, ali to sada nije važno.

Illustration by Freepik Storyset

1. Napišite program koji će u listu unositi ocjene učenika, a zatim izračunati njihov prosjek ocjena. Nije poznato unaprijed koliko učenika ima u razredu, pa se taj broj unosi na početku programa. Na kraju se ispisuje:
   `Prosjek ocjena ovih učenika iznosi ___ .`
2. U prethodnom zadatku dodati izračun i ispis koliko učenika ima ocjenu 5, a koliko ocjenu 1. ispis bi trebao biti:
   `Među ovim učenicima njih ____ima ocjenu 5, a _____ učenika ima ocjenu 1.`
3. Napišite program koji unosi cijene određenog broja proizvoda. Unosi se broj proizvoda i njihove cijene, koje se spremaju u listu. Zatim se ispisuje najviša i najniža cijena proizvoda, te njihova razlika.

Kolekcije podataka – Rječnik

- Dictionary ili Rječnik je kolekcija parova podataka.
- Rječnik koristi { } zagrade
- Svaki element Rječnika ima dva dijela:
  - Key ili Ključ
  - Value ili Vrijednost
- Key ili Ključ mora biti jedinstven u Rječniku. Zato jer Ključ mora biti jedinstven dopušten tipovi podataka za Ključ su:
  - String; Brojevi i N-terac
- Pomoću Ključa pristupamo drugom dijelu para, odnosno podacima. Ključ je isto kao i Indeks u listi.
- Value ili Vrijednost predstavlja sadržaj koji želimo pohraniti u kolekciju, odnosno Rječnik. Value može biti bilo koji tip podatka, a često je druga kolekcija kao lista ili drugi rječnik

Rad s Rječnikom / Dictionary

- Manipulacija podacima unutar Rječnika:
  - ***naziv_rjecnika[key]***
  - *naziv_rjecnika.***items()**
  - *naziv_rjecnika.***keys()**
  - *naziv_rjecnika.***values()**

Vježba – rječnik

- Kreirajte bazu s vozilima firme. ID svakog retka je cijeli broj, a podaci koji se čuvaju o svakom vozilu su: tip, proizvođač, registarska oznaka, godina prve registracije te cijena u eurima. Ispišite cijelu tablicu tako da ID odvojite od ostatka retka jednim TABom, a druge informacije formatirajte tako da prvi red tablice predstavlja naslovni red, a ostali redovi tablice predstavljaju podatke iz baze.

Illustration by Freepik Storyset

Vježba – rječnik

ID	Tip	Proizvođač	Registarska oznaka	Godina prve registracije	Cijena u EUR
1	Kamion	Iveco	OS 001 ZZ	2015	45.000,00 €
2	Kamion	Iveco	OS 002 ZZ	2015	47.000,00 €
3	Tegljač	MAN	RI 001 ZZ	2018	78.000,00 €
4	Tegljač	MAN	RI 002 ZZ	2020	97.000,00 €
5	Kombi	Mercedes Benz	ST 001 ZZ	2013	12.000,00 €
6	Kombi	Volkswagen	ST 002 ZZ	2021	35.000,00 €
7	Dostavno vozilo	Volkswagen	ZG 001 ZZ	2010	9.000,00 €
8	Dostavno vozilo	Volkswagen	ZG 002 ZZ	2010	9.300,00 €

Rad s Rječnikom / Dictionary

- Manipulacija podacima unutar Rječnika nastavak:
  - *naziv_rjecnika.***clear()**
  - *naziv_rjecnika.***pop(*key, default*)**
  - *naziv_rjecnika.***popitem()**

Kolekcije podataka – N-terac (tuple)

- Tuple koristi ( ) zagrade
- N-terac je Nepromjenjiv.
- Primjena
  - TUPLE se često koristi kao tip pohrane povezanih podataka unutar neke kolekcije. Zato jer je nepromjenjiv koristi se kao ključ u Rječnicima
  - Recimo podaci o nekoj osobi su pohranjeni u jednu TUPLE kolekciju, a onda te TUPLE kolekcije su pohranjene u neku listu
- Provjerite što vraćaju naredbe za Dictionary:
  - *naziv_rjecnika.***items()**
  - *naziv_rjecnika.***keys()**
  - *naziv_rjecnika.***values()**

ALGEBRA

86

87

Tok izvršavanja kôda

- Često koristimo uvjete kako bismo kontrolirati tijek događaja. Recimo, kada dođete u banku i želite podići neki iznos novca. Ako imate dovoljno na računu i odobren minus, onda ćete dobiti novac, a ako nemate, dobit ćete ispriku da nemate dovoljno novaca na računu i da vam ne mogu isplatiti trženi iznos.
- Uvjete koristimo i za kontrolu toka izvršavanja kôda. Za to koristimo petlje.
- Ponavljanje kôda i predefiniranom broju iteracija
  - FOR petlja – sve dok ima elemenata i kolekciji …
- Uvjetno izvršavanje
  - IF … ELSE; IF … ELIF … ELSE petlja – ako je uvjet ispunjen izvrši …
- Uvjetno ponavljanje kôda
  - WHILE petlja – sve dok je uvjet ispunjen izvršavaj …
- Kombinacija

Relacijski operatori

Operator	Opis	Primjer
>	Veće od – a je veće od b	a > b
<	Manje od – a je manje od b	a < b
==	Identično – a je identično b	a == b
!=	NIJE identično – a nije identično b	a != b
>=	Veće i jednako od – a je veće i jednako od b	a >= b
<=	Manje i jednako od – a je manje i jednako od b	a <= b

Tabela logičkih izraza

A	B	A and B	A or B	not B
✓ True	✓ True	✓ True	✓ True	✗ False
✓ True	✗ False	✗ False	✓ True	✓ True
✗ False	✓ True	✗ False	✓ True	✗ False
✗ False	✗ False	✗ False	✗ False	✓ True

ALGEBRA

90

UVJETNE NAREDBE: IF, IF ELSE, IF ELIF ELSE

- Analogno primjeru iz banke u vezi podizanja novca s računa, u programskim jezicima imamo naredbe koje ovisno o uvjetu određuju tijek izvršavanja programa.
- Ključne riječi koje sačinjavaju te naredbe su: IF, ELIF I ELSE
- AKO (IF) je uvjet ispunjen (njegova vrijednost je True), tada će se izvršiti blok instrukcija
- INAČE (ELSE) će se izvršiti drugi blok instrukcija
- Ukoliko postoji više mogućih opcija od dvije, koristi se ELIF, uvijek između IF i ELSE

ALGEBRA

91

IF i IF-ELSE UVJETNE NAREDBE

```
if uvjet:
```
izvrši instrukcije SAMO AKO je uvjet točan ili ima vrijednost True

nastavi dalje s izvođenjem programa

```
if uvjet:
```
izvrši instrukcije SAMO AKO je uvjet točan ili ima vrijednost True

```
else:
```
AKO uvjet NIJE ispunjen, odnosno vrijednost mu je FALSE

ALGEBRA

IF-ELIF-ELSE UVJETNA NAREDBA

```
if prvi_uvjet:
```
izvrši instrukcije SAMO AKO je prvi_uvjet točan ili ima vrijednost True

```
elif drugi_uvjet:
```
Ako prvi uvjet NIJE zadovoljen, znači da je prvi_uvjet NE točan ili ima vrijednost False

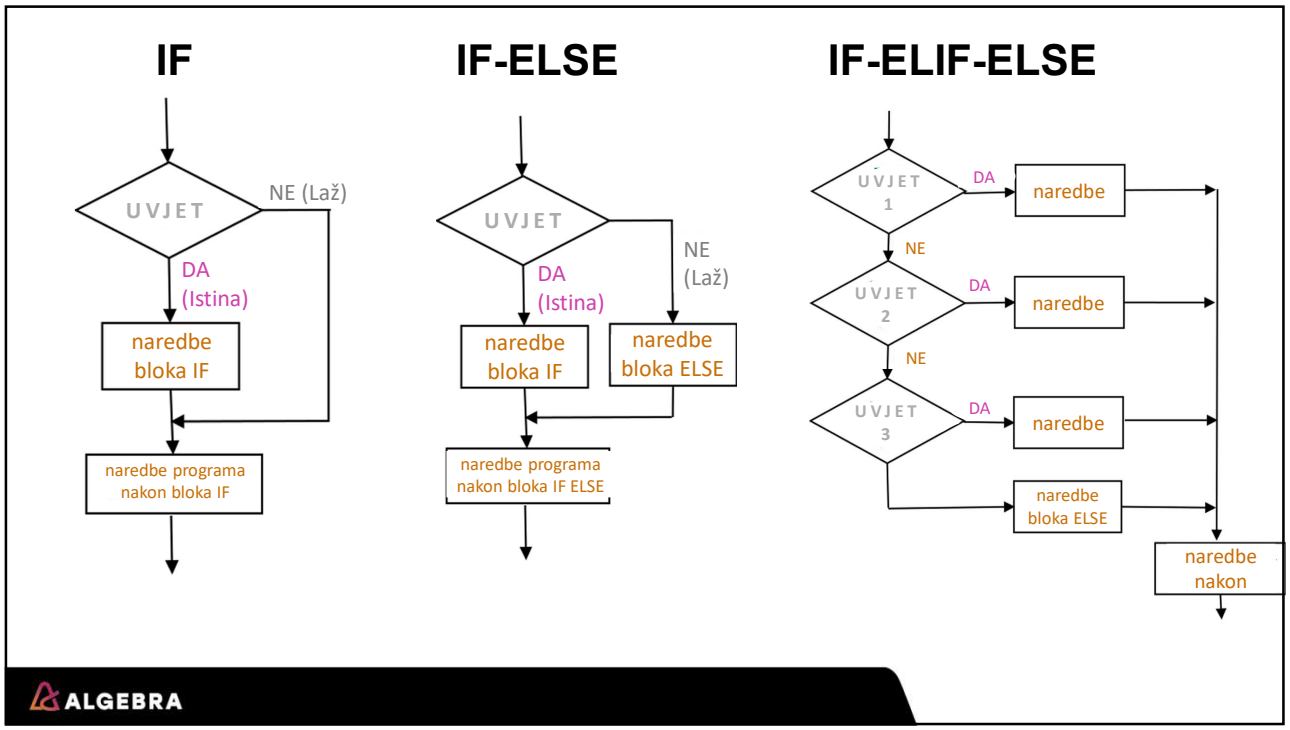
Tada je izvršavanje programa došlo do ove linije pa OPET slijedi provjera AKO je drugi_uvjet točan ili ima vrijednost True izvrši instrukcije u ovom bloku

```
elif treci_uvjet:
```
isto kao i za drugi uvjet i za četvrti i peti i … nema ograničenja u ELIF provjerama

```
else:
```
AKO niti jedan od uvjeta NIJE ispunjen, odnosno svi su FALSE

TADA BEZ obzira na UVJETE I VRIJEDNOSTI IZVRŠI aktivnosti iz ovog bloka

ALGEBRA



IF ELSE UVJETNA NAREDBA - Zadaci

- Kreirajte listu od 1 do broja 30. Ispišite sve brojeve koji su djeljivi s 3, 6 i 9
  - Provjera je li broj djeljiv s nekim drugim radimo pomoću % (modulo) operanda.
  - 15 % 3 NEMA ostatka, odnosno to je 0 pa je 15 djeljiv s 3.
  - 16 % 3 je 1, odnosno NIJE jednak 0 pa 16 NIJE djeljiv s 3.

- Napišite program koji provjerava pripada li unesena riječ vrsti riječi *palindrom*.
  - Palindrom je riječ koja se jednako piše (i čita) s lijeva na desno i s desna na lijevo

Vježba – rječnik ISPRAVITE ISPIS

- Kreirajte bazu s vozilima firme. ID svakog retka je cijeli broj, a podaci koji se čuvaju o svakom vozilu su: tip, proizvođač, registarska oznaka, godina prve registracije te cijena u eur. Ispišite cijelu tablicu tako da ID odvojite od ostatka retka jednim TABom, a druge informacije formatirajte tako da prvi red tablice predstavlja naslovni red, a ostali redovi tablice predstavljaju podatke iz baze.

ALGEBRA

Illustration by Freepik Storyset

96

IF ELSE NAREDBA – Zadaci tekst

- U generičkom tekstu 'Lorem ipsum ...' (https://www.lipsum.com/) pronađite koliko se puta pojavljuje neka riječ.
  - Probajte s Lorem.

ALGEBRA

97

WHILE petlja

- IF petlja je imala uvjet koji ako je ispunjen izvršava se blok programskog kôda te petlje SAMO JEDNOM.
- Nakon završetka bloka programskog kôda IF petlje, program se nastavlja izvršavati dalje, izvan IF petlje.
- WHILE petlja je slična. Isto tako ima uvjet koji ako je ispunjen, osigurava pokretanje bloka programskog kôda unutar WHILE petlje, ali nakon završetka tog bloka programskog kôda, WHILE petlja će ponoviti provjeru uvjeta.
- Ako je uvjet ispunjen, OPET će se pokrenuti isti blok programskog kôda.
- I to će se ponavljati SVE DOK (while) je uvjet ispunjen.
- WHILE petlju možemo usporediti s FOR petljom, samo što je uvjet kod WHILE petlje jasno iskazan, naveden, dok kod FOR petlje je uvjet skriven i vezan je uz postojanje elemenata u kolekciji.

WHILE petlja

`while uvjet:`

sve dok je `uvjet` točan izvrši instrukcije u ovom bloku.

Kada završiš s izvršavanjem zadnje instrukcije ponovi provjeru uvjeta te ako je ispunjen ponovi izvršavanje bloka instrukcija. To ponavljaj SVE DOK uvjet nije ispunjen.

- Prepravite zadatak „vozni park” tako da umjesto FOR, radi s WHILE petljom

break, continue, pass

- Kada želimo izvršavanje petlje završiti prije kraja - "nasilno izaći iz petlje" koristimo ključnu riječ BREAK
- Kada NE želimo izaći iz petlje, ali želimo da se jedan ili više ciklusa petlje NE izvrši do kraja, nego da se jedan dio instrukcija PRESKOČI te ako su ispunjeni uvjeti započne novi ciklus, koristimo naredbu CONTINUE
- Komentari se u kôdu ignoriraju, preskaču, ne postoje. Međutim, kada trebamo izvršiti naredbu koje na radi ništa koristimo naredbu PASS
  - PASS možemo smatrati kao komentar koji se izvršava. Program smatra da ima naredbu za izvršiti, samo što ta naredba ne radi ništa.

Zadaci

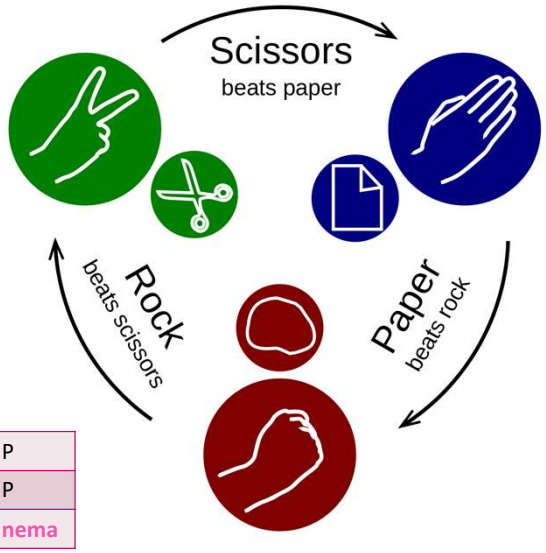
- Preraditi vježbe vezane uz FOR petlju tako da umjesto FOR petlje koriste WHILE petlju:
  - Je li riječ palindrom
  - Baza vozila firme
  - Generator akorda

Zadaci

- Napravite aplikaciju za prikaz tablice množenja. Korisnik treba moći unijeti do kojeg broja će se prikazati tablica.
- Sjećate se Računalnog razmišljanja i pogađanja broja između 1 i 100. Sada imate dovoljno znanja da napišete program koji će Vam omogućiti igranje igre pogađanje broja.
- Napravite aplikaciju za konverziju (u oba smjera):
  - km u milju – (1 km = 0.6214 milje)
  - °C u °F – (0°C = 32°F) obrnuto $T_{(°F)} = T_{(°C)} * (9/5) + 32$
  - kg u funtu (pounds) – 1 kg = 2.2046 pounds
  - Litra u US galon – 1l = 0.2642 US gal
  - kW (kilowatt) u ks (horsepoweer ili konjska snaga) – 1 kW = 1.3596
- Rezultate u svim zadacima je potrebno formatirano ispisati na ekran.
- U svim zadacima, nakon pokretanja programa, korisnik treba imati mogućnost izbora želi li nastaviti koristiti program ili želi završiti, odnosno izaći iz programa.

Kamen-Škare-Papir Pogodi broj

- Napravite program koji će vam omogućiti igranje igre protiv računala - kamen, škare, papir



K	K	K	Š	Š	Š	P	P	P
K	Š	P	K	Š	P	K	Š	P
nema	K	P	K	nema	Š	P	Š	nema

Rock-paper-scissors - Rock paper scissors - Wikipedia