Ručne bez počítača zisti, čo sa vypíše:

**>>>** x, y = 'Bratislava', 'Košice'

**>>>** y[1] + x[4] + y[3] + x[-4] + y[-5]

...

**>>>** x[5:8] + 3 \* x[3] + y[2:]

...

**>>>** y[:2] + x[-2:]

...

**>>>** x[1::2] + y[2::2] + x[2::3]

...

**>>>** x.replace('a', 'e') + y.replace('ic', 'm')

...

**>>>** (y + x).replace('i', '').replace('a', 'xa')

...

Potom to skontroluj pomocou Pythonu.

1. Prevod zadaného čísla z desiatkovej sústavy do sústavy so základom, ktorý zadá užívateľ a opačne. Smer prevodu bude možné vybrať v jednoduchom menu. V programe treba využiť vlastné funkcie s návratovou hodnotou.
2. Napíš funkciu sucet(retazec), ktorá dostáva znakový reťazec s dvomi celými číslami oddelenými znakom '+'. Funkcia vráti (nič nevypisuje) celé číslo, ktoré je súčtom dvoch čísel v reťazci. Použi rezy, metódu retazec.find('+') a funkciu int(). Napríklad:

**>>>** x = sucet('12+9')

**>>>** x

21

**>>>** sucet('987654321+99999')

987754320

1. Zovšeobecni funkciu sucet(retazec) z predchádzajúcej úlohy: vstupný reťazec obsahuje aspoň jedno číslo a keď ich je viac, sú oddelené znakom '+'. Funkcia vypočíta súčet. Napríklad:

**>>>** x = sucet('12+9')

**>>>** x

21

**>>>** sucet('1+2+3+4')

10

**>>>** sucet('1234')

1234

1. Napíš funkciu postupnost(start, koniec), ktorá vytvorí znakový reťazec z postupnosti čísel range(start, koniec). Čísla v tejto postupnosti budú oddelené znakom medzera ' '. Napríklad:

**>>>** p = postupnost(5, 13)

**>>>** p

'5 6 7 8 9 10 11 12'

Do funkcie pridaj ešte jeden parameter postupnost(start, koniec, krok=1) tak, aby fungovalo napríklad:

**>>>** p = postupnost(13, 5, -2)

**>>>** p

'13 11 9 7'

1. Napíš funkciu rozsekaj(text, sirka), ktorá vypíše zadaný text do viacerých riadkov, pričom každý (možno okrem posledného) má presne sirka znakov. Napríklad:

**>>>** ret = rozsekaj('Anicka dusicka, kde si bola', 10)

**>>>** ret

'Anicka dus\nicka, kde \nsi bola'

**>>>** print(ret)

Anicka dus

icka, kde

si bola

1. Napíš funkciu stvorec(n, znak), ktorá vráti jeden dlhý znakový reťazec. Znakový reťazec by po vypísaní pomocou print vytvoril obvod štvorca z daného znaku. Môžeš predpokladať, že n nebude menšie ako 2. Napríklad:

**>>>** r = stvorec(5, '#')

**>>>** r

'#####\n# #\n# #\n# #\n#####'

**>>>** print(r)

#####

# #

# #

# #

#####

Pokús sa to vyriešiť tak, že telo funkcie bude obsahovať jediný riadok return:

**def** stvorec(n, znak):

**return** ...

1. Napíš funkciu vyhod\_duplikaty(retazec), ktorá z daného reťazca vyhodí všetky za sebou idúce opakujúce sa znaky (nechá len jeden z nich). Napríklad:

**>>>** x = vyhod\_duplikaty('Braatisssllavaaaaa')

**>>>** x

'Bratislava'

1. Program, v ktorom treba implementovať algoritmus odstránenia podreťazca z reťazca znakov. Program dostane vstupný reťazec a podreťazec, následne spočíta a odstráni všetky výskyty podreťazca v reťazci bez použitia metód count a replace.
2. Napíš funkciu ozatvorkuj(retazec, podretazec), ktorá v zadanom reťazci retazec všetky výskyty daného podreťazca ozátvorkuje. Napríklad:

**>>>** b = ozatvorkuj('Bratislava', 'a')

**>>>** b 'Br(a)tisl(a)v(a)'

**>>>** ozatvorkuj('prospešné programovanie v prologu', 'pro')

'(pro)spešné (pro)gramovanie v (pro)logu'

1. Program, v ktorom užívateľ zadá vstupný reťazec; pomocou jednoduchého menu si vyberie, či chce reťazec náhodne premiešať alebo rozdeliť striedavo po znakoch na dva podreťazce.
2. Znakový reťazec vieme prevrátiť pomocou zápisu retazec[::-1]. Napíš funkciu prevrat(retazec), ktorá len pomocou cyklu a zreťazovania prevráti zadaný reťazec. Napríklad:

**>>>** x = prevrat('tseb eht si nohtyP')

**>>>** x

'Python is the best'

1. Program, ktorý v zadanej vete nájde najdlhšie slovo. Predpokladáme, že slová sú vo vete oddelené vždy jednou medzerou.
2. Napíš funkcie male(retazec, i) a velke(retazec, i), ktoré i-te písmeno v reťazci prerobia na malé (resp. veľké). Napríklad:

**>>>** r = male('PYTHON', 3)

**>>>** r

'PYThON'

**>>>** r = velke('python', 5)

**>>>** r

'pythoN'

1. Napíš funkciu riadky(retazec), ktorá vypíše daný viacriadkový reťazec, ale pritom každý riadok očísluje číslami od 1 do počet riadkov. Napríklad:

**>>>** riadky('prvy riadok**\n\n**treti je posledny')

1. prvy riadok

2.

3. treti je posledny

**>>>** riadky('len **\\**n jeden riadok')

1. len \n jeden riadok

BONUS

Napíš funkciu posun\_znak(znak, posun), ktorá posunie daný znak v abecede o p znakov vpravo (resp. vľavo, ak je záporné). Na konci abecedy sa pokračuje od začiatku. Funkcia posúva len písmená malej abecedy, ostatné znaky nemení. Napríklad:

**>>>** posun\_znak('c', 4)

'g'

**>>>** posun\_znak('g', -4)

'c'

**>>>** posun\_znak('x', 10)

'h'

**>>>** posun\_znak('A', 10)

'A'

Napíš funkciu zakoduj(text, posun), ktorá posunie v abecede všetky znaky (pomocou funkcie posun\_znak). Napríklad:

**>>>** x = zakoduj('pyThon', 10)

**>>>** x

'ziTryx'

**>>>** zakoduj(x, -10)

'pyThon'

**>>>** zakoduj(x, 16)

'pyThon'

Unicode 0x2654 a ďalších päť za ním sú obrázky šachových figúrok. Napíš program, ktorý do grafickej plochy nakresli vedľa seba všetkých 6 figúrok náhodnými farbami väčším fontom (napríklad 'arial 50'). Môžeš dostať takýto obrázok:

[](https://python.input.sk/_images/06_c01.png)