Pracovný list – Reťazce

Úlohy na skúmanie

|  |  |
| --- | --- |
| **Úloha 1** | Otvorte program **14\_01\_retazec.py**.  slovo1 = 'zu' slovo2 = 'lienka' spolu = 2 \* slovo1 + slovo2  print(spolu) print(len(spolu))  if 'ie' in spolu:  print('ano')  else:  print('nie')  print(spolu[4])  Program spusťte viackrát. Skúšajte meniť reťazce/znaky, čísla, operácie s reťazcami a sledujte výpisy programu do konzoly. Na základe svojich pokusov odpovedzte na nasledujúce otázky:   1. Doposiaľ sme pri výpočtoch s číslami používali aritmetické operácie +, –, \* a /. Ktoré z týchto operácií môžeme používať aj pri práci s reťazcami/znakmi? Čo je výsledkom použitia týchto operácií? 2. Čo je výsledkom operácie retazec1 in retazec2? 3. Čo je návratovou hodnotou funkcie len(retazec)? 4. Čo vypíše program do konzoly príkazom v poslednom riadku print(spolu[4])? . |

|  |  |
| --- | --- |
| **Úloha 2** | Porovnajte výstupy nasledujúcich cyklov for – najprv odhadnite ich výstupy, svoje predpoklady overte spustením programu **14\_02\_retazec.py**:  retazec = 'Informatika'  for i in range(len(retazec)):  print(i)  for i in retazec:  print(i)  Prvý cyklus for vypíše:   1. môj tip 2. skutočnosť   Druhý cyklus for vypíše:   1. môj tip 2. skutočnosť |

Úlohy na precvičenie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Úloha 3** | Zuzka sa odsťahovala s celou rodinou do Kanady. So svojou najlepšou kamarátkou Katkou komunikujú písomne a na utajenie svojich správ si dohodli šifru: do textu správy vložia za každý znak ľubovoľný znak. Takto upravená správa vyzerá ako motanica nezmyselných slov, napr. text „Ahoj, Zuzka!“ po zašifrovaní vyzerá takto: „A\*huoXjj QZ8uyzKk+a,!(“.  Aby sa im správy ľahšie dešifrovali, obidve vytvorili vlastnú funkciu desifruj(), ktorej návratovou hodnotou je dešifrovaná správa. Každá z funkcií však vyzerá odlišne, dievčence sa nevedia dohodnúť, ktorá je správna. Pomôžte im pri rozhodovaní – určte, ktorá z funkcií plní danú úlohu.   |  |  | | --- | --- | | Zuzkina funkcia: | Katkina funkcia: | | **def** desifruj(s):  vysledok = ''  **for** i **in** range(0, len(s), 2):  vysledok = vysledok + s[i]   **return** vysledok | **def** desifruj(s):  vysledok = s[::2]   **return** vysledok | |

|  |  |
| --- | --- |
| **Úloha 4** | Pani učiteľka v rámci záverečného opakovania pripravuje na každú hodinu slovenského jazyka krátky diktát. Jeho náročnosť posudzuje najmä podľa počtu písmen i, í, I, Í, y, ý, Y, Ý v diktáte.  Vytvorte pre pani učiteľku program, ktorému zadá text diktátu a program spočíta a vypíše počet sledovaných samohlások i, í, I, Í, y, ý, Y, Ý v zadanom texte.  Napr. pri vstupnom texte „V našej peci myši pištia. Asi nie sú sýte.“ bude odpoveďou číslo 8.“ |

|  |  |
| --- | --- |
| **Úloha 5** | Tajomstvo komunikácie Zuzky a Katky odhalil Katkin brat Miško. Preto sa dievčatá rozhodli, že budú komunikovať po anglicky a zároveň budú používať Pig Latin – jazykovú hru, ktorá slúži na pobavenie, aj na utajenie komunikácie pred nepovolanými osobami. Princíp hry spočíva v úprave slov podľa týchto pravidiel   * Ak slovo začína spoluhláskou, táto sa presunie na koniec slova a za ňu sa pridá prípona –ay, napr. door => oorday, pen => enpay. * Ak slovo začína samohláskou, pridá sa len prípona –way, napr. apple =>appleway, old => oldway.   Vytvorte program **pig\_latin.py**, ktorý na vstupe dostane slovo (zapísané malými písmenami anglickej abecedy) a do konzoly vypíše toto slovo upravené podľa pravidiel jazykovej hry Pig Latin. |

Sebahodnotiaci test

|  |  |
| --- | --- |
| 1. | Nasledujúci program dešifruje vstupnú správu, ktorá vznikla podľa tohto pravidla šifrovania – pred a za každý znak správy bol vložený jeden náhodný znak (napr. správa „Pošli správu.“ je zašifrovaná v tvare „3Pxaob4šSbl4sils 4Is45pM0r7GáAAvmKuL8.4“).  Doplňte chýbajúcu časť kódu, aby bol program funkčný.  zasifrovana\_sprava = input(’Zašifrovaná správa: ’)  odsifrovana\_sprava = zasifrovana\_sprava[ : : ]  print(f’{zasifrovana\_sprava} => {odsifrovana\_sprava}’) |
| 2. | Vyberte z ponúknutých dvoch programov **A** a **B** ten, ktorý pre zadaný reťazec vypíše ordinálne hodnoty jeho znakov v tvare:  0 => ordinálna hodnota prvého znaku reťazca  1 => ordinálna hodnota druhého znaku reťazca  2 => ordinálna hodnota tretieho znaku reťazca  ...   |  |  | | --- | --- | | ***Program A*** | ***Program B*** | | retazec = input(’Vstupný reťazec: ’)  for i in range(len(retazec)):  print(f’{i} => {ord(retazec[i])}’) | retazec = input(’Vstupný reťazec: ’)  for i in retazec:  print(f’{i} => {ord(i)}’) |   Daný problém rieši program: |