**Maturitné zadanie č. 1**

***Úloha č.1- Záznamy z meteorologických staníc***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**z=open('meteo\_stanice.txt','r')**

**teploty=[]**

**stanice=[]**

**merania=0**

**for i in z:**

**print(i)**

**teplota=i[21:26]**

**stanice.append(i[0:3])**

**teplota=teplota.replace(',','.')**

**teploty.append((float(teplota)))**

**merania += 1**

**print(f'Meranie sa spravilo {merania}')**

**print(f'Namerane teploty su {teploty}')**

**max=0**

**for i in teploty:**

**if i > max:**

**max = i**

**print(f'Najvacsia teplota bola {max}')**

**ktora=teploty.index(max)**

**print(f'Najviacsia teplota bola na stanici {stanice[ktora]}')**

**dokopy=0**

**for i in teploty:**

**dokopy += i**

**print(f'Priemerna teplota bola {dokopy/merania}')**

Tento program slúži na spracovanie dát z meteostaníc zo súboru "meteo\_stanice.txt". Tu je komentár k jednotlivým častiam programu:

1. Otvorenie súboru:
   * Súbor "meteo\_stanice.txt" sa otvára pomocou z=open('meteo\_stanice.txt','r') pre čítanie údajov o meteostaniciach.
2. Spracovanie údajov zo súboru:
   * Vytvára sa prázdny zoznam teploty, kde budú uložené namerané teploty.
   * Vytvára sa prázdny zoznam stanice, do ktorého budú uložené názvy meteostaníc.
   * Inicializuje sa premenná merania na 0, ktorá bude slúžiť na uchovanie počtu meraní.
   * Pre každý riadok v súbore z:
     + Vypíše sa riadok pomocou print(i).
     + Získava sa teplota z riadku na základe určitej pozície a uloží sa do premennej teplota.
     + Názov meteostanice (prvých 3 znaky v riadku) sa pridá do zoznamu stanice.
     + Teplota sa upraví (nahradenie ',' za '.') a prevedie sa na desatinné číslo (float).
     + Nameraná teplota sa pridá do zoznamu teploty.
     + Inkrementuje sa premenná merania o 1.
3. Výpis počtu meraní a nameraných teplôt:
   * Vypíše sa počet meraní pomocou print(f'Meranie sa spravilo {merania}').
   * Vypíše sa zoznam nameraných teplôt pomocou print(f'Namerane teploty su {teploty}').
4. Výpočet a výpis najvyššej teploty:
   * Inicializuje sa premenná max na hodnotu 0, ktorá bude slúžiť na uchovanie najvyššej teploty.
   * Pre každú nameranú teplotu v zozname teploty:
     + Ak je nameraná teplota väčšia ako aktuálna maximálna teplota (max), nastaví sa nová maximálna teplota (max = i).
   * Vypíše sa najvyššia teplota pomocou print(f'Najvacsia teplota bola {max}').
   * Zistí sa index najvyššej teploty v zozname teploty pomocou ktora=teploty.index(max).
   * Vypíše sa názov stanice, na ktorej bola nameraná najvyššia teplota pomocou print(f'Najviacsia teplota bola na stanici {stanice[ktora]}').
5. Výpočet a výpis priemernej teploty:
   * Inicializuje sa premenná dokopy na hodnotu 0, ktorá bude slúžiť na sčítanie všetkých teplôt.
   * Pre každú nameranú teplotu v zozname teploty:
     + Pripočíta sa nameraná teplota k premenná dokopy.
   * Vypíše sa priemerná teplota pomocou print(f'Priemerna teplota bola {dokopy/merania}'), kde sa výška všetkých teplôt (dokopy) delí počtom meraní (merania).

Tento program číta údaje o meteostaniciach zo súboru, získava teploty, určuje najvyššiu teplotu, vypočíta priemernú teplotu a vypisuje tieto informácie.

***Úloha č.2***

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Softvér a hardvér, Informačná spoločnosť.

***Pomôcky:*** súbor meteo\_stanice.txt pre úlohu č.1, súbor *mzdy.xls* pre úlohu č.2**Maturitné zadanie č.2**

***Úloha č.1 - Prvočísla***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**cislo = int(input('Zadaj číslo: '))**

**pocet = 0**

**for delitel in range(1, cislo+1):**

**if cislo % delitel == 0:**

**pocet += 1**

**if pocet == 2:**

**print(cislo, 'je prvočíslo')**

**else:**

**print(cislo, 'nie je prvočíslo')**

**cislo2=int(input('Zadajte druhe cislo: '))**

**delitele=[]**

**for delitel in range(1,cislo2+1):**

**if cislo2 % delitel == 0:**

**delitele.append(delitel)**

**print('Delitele cisla ',cislo2,'su',str(delitele))**

**od=int(input('Zadajte cisla od: '))**

**do=int(input('Zadajte cislo do: '))**

**for i in range(od,do+1):**

**pocet2=0**

**for delitel in range(1,i+1):**

**if i % delitel == 0:**

**pocet2 += 1**

**if pocet2 == 2:**

**print('Cislo',i,'je prvocislo')**

**else:**

**print('Cislo',i,'nie je prvocislo')**

Tento program je zameraný na prácu s číslami a overovanie, či sú prvočíslami. Tu je komentár k jednotlivým častiam programu:

1. Časť: Overenie, či zadané číslo je prvočíslo:
   * Používateľ zadáva číslo cez vstup pomocou funkcie input(), a táto hodnota sa konvertuje na celé číslo pomocou int().
   * Premenná pocet sa inicializuje na 0, táto premenná bude slúžiť na počítanie počtu deliteľov zadaného čísla.
   * S použitím for cyklu prechádzame všetky čísla od 1 do zadaného čísla (cislo+1).
   * V každom kroku cyklu sa kontroluje, či je cislo deliteľné bez zvyšku delitelom. Ak áno, zvýši sa hodnota premennej pocet o 1.
   * Na základe hodnoty premennej pocet sa rozhoduje, či je zadané číslo prvočíslo. Ak je pocet rovné 2, potom je číslo prvočíslo, inak nie je.
2. Časť: Získanie všetkých deliteľov zadaného čísla:
   * Používateľ zadáva druhé číslo cez vstup.
   * Vytvára sa prázdny zoznam delitele, do ktorého budú postupne pridávané všetky delitele.
   * Opäť s použitím for cyklu prechádzame všetky čísla od 1 do zadaného čísla (cislo2+1).
   * V každom kroku cyklu sa kontroluje, či je cislo2 deliteľné bez zvyšku delitelom. Ak áno, pridá sa delitel do zoznamu delitele.
   * Na konci sa vypíšu všetky delitele zadaného čísla.
3. Časť: Overenie, či čísla v zadanom rozsahu sú prvočísla:
   * Používateľ zadáva dolnú a hornú hranicu rozsahu cez vstup.
   * S použitím dvoch for cyklov prechádzame všetky čísla od od po do+1.
   * V každom kroku vnútorného cyklu sa kontroluje, či je aktuálne číslo deliteľné bez zvyšku delitelom. Ak áno, zvýši sa hodnota premennej pocet2 o 1.
   * Na základe hodnoty premennej pocet2 sa rozhoduje, či je aktuálne číslo prvočíslo. Výsledok sa vypisuje na obrazovku.

Celkovo program spracováva vstupy od používateľa a vykonáva overenia čísel na prvočíselnosť. Vypisuje výsledky na obrazovku.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Komunikácia a spolupráca, Informačná spoločnosť.

**Maturitné zadanie č. 3**

***Úloha č.1 – Náhodné skúšanie***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**import random**

**pocets=int(input('Zadajte pocet studentov'))**

**poceto=int(input('Zadajte pocet otazok'))**

**poradieotazokp=[]**

**poradieotazokn=[]**

**dokopy=[]**

**student=[]**

**for i in range(1,poceto+1):**

**if i%2==0:**

**poradieotazokp.append(i)**

**else:**

**poradieotazokn.append(i)**

**for i in range(pocets):**

**koklky=random.randint(1, pocets)**

**while koklky in student:**

**koklky=random.randint(1,pocets)**

**student.append(koklky)**

**random.shuffle(poradieotazokn)**

**random.shuffle(poradieotazokp)**

**print(poradieotazokp)**

**print(poradieotazokn)**

**print(student)**

**kolka=0**

**while kolka != pocets:**

**if kolka % 2 == 0:**

**print(f'Student {student[kolka]} si vytiahne otazku {poradieotazokp[kolka]}')**

**else:**

**print(f'Student {student[kolka]} si vytiahne otazku {poradieotazokn[kolka]}')**

**kolka += 1**

Tento program vykonáva žrebovanie otázok pre študentov na základe počtu študentov, počtu otázok a poradia otázok.

1. Importuje sa modul "random", ktorý umožňuje generovať náhodné čísla.
2. Študenti zadajú počet študentov a počet otázok.
3. Inicializujú sa prázdne zoznamy "poradieotazokp", "poradieotazokn", "dokopy" a "student".
4. V cykle "for" sa generujú poradia pre párne a nepárne otázky. Poradie pre párne otázky sa ukladá do zoznamu "poradieotazokp", a to od 1 po "otazky" s krokom 2. Poradie pre nepárne otázky sa ukladá do zoznamu "poradieotazokn", a to od 0 po "otazky" s krokom 2 (s výnimkou 0).
5. V ďalšom cykle "for" sa generuje náhodné poradie pre každého študenta. Náhodné číslo sa generuje v rozsahu od 1 do 30. Ak sa generované číslo už nachádza v zozname "student", generuje sa ďalšie číslo, kým sa nenájde unikátne číslo pre študenta. Vygenerované číslo sa pridáva do zoznamu "student".
6. Zoznamy "poradieotazokp" a "poradieotazokn" sa náhodne zamiešajú pomocou funkcie "random.shuffle".
7. Vypíšu sa zoznamy "poradieotazokp", "poradieotazokn" a "student" na obrazovku.
8. Inicializuje sa premenná "kolka" na hodnotu 0.

V cykle "while" sa opakuje nasledujúce:

a. Ak je hodnota premennej "kolka" deliteľná 2 bez zvyšku, vypíše sa text s číslom študenta a otázky s indexom "kolka" zo zoznamu "poradieotazokp".

b. V opačnom prípade (hodnota premennej "kolka" nie je deliteľná 2), vypíše sa text s číslom študenta a otázky s indexom "kolka" zo zoznamu "poradieotazokn".

1. c. Inkrementuje sa premenná "kolka" o 1.

Program vykonáva žrebovanie otázok pre študentov na základe zadaných parametrov. Generuje náhodné poradia otázok a študentov a následne ich vypisuje v zadanom formáte.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Informačná spoločnosť.

**Maturitné zadanie č. 4**

## *Úloha č.1 –Súťaž v behu*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## z=open('sutaz\_vbehu.txt.','r')

## zucastneny=0

## mena=[]

## casy=[]

## for i in z:

## medzera=i.find(' ')

## sutaziaci=i[:medzera]

## cas=i[medzera+1:]

## print(f'Sutaziaci s meno {sutaziaci} dobehol do ciela v case {cas}',end='')

## zucastneny += 1

## mena.append(sutaziaci)

## casy.append(cas)

## najlepsic=i[0]

## for i in casy:

## if i < najlepsic:

## najlepsic=i

## z.close()

## z=open('sutaz\_vbehu.txt.','r')

## meno=''

## for i in z:

## medzera=i.find(' ')

## cas=i[medzera+1:]

## sutaziaci=i[:medzera]

## if cas ==najlepsic:

## meno = sutaziaci

## najlepsic=int(najlepsic)

## hodiny=najlepsic//60

## sekund=najlepsic % 60

## print('')

## print(f'Zucastnilo sa {zucastneny} sutaziacich')

## print(f'Celkovy vitaz je {meno} s casom {hodiny} minut a {sekund} sekund')

Tento program pracuje so súborom "sutaz\_vbehu.txt", ktorý obsahuje údaje o súťažiacich a ich časoch v behu. Tu je komentár k jednotlivým častiam programu:

1. Časť: Načítanie údajov zo súboru a výpis výsledkov:
   * Otvára sa súbor "sutaz\_vbehu.txt" pre čítanie pomocou funkcie open().
   * Premenná zucastneny sa inicializuje na 0, táto premenná bude slúžiť na počítanie počtu súťažiacich.
   * Vytvárajú sa prázdne zoznamy mena a casy, do ktorých budú postupne pridávané mená a časy súťažiacich.
   * S použitím for cyklu sa prechádza každý riadok zo súboru.
   * Používa sa funkcia find() na nájdenie pozície medzery vo vstupnom riadku, ktorá oddeluje meno súťažiaceho od času.
   * Na základe pozície medzery sa rozdeľuje riadok na meno (sutaziaci) a čas (cas).
   * Vypisuje sa správa o súťažiacom a jeho čase pomocou formátovanej reťazcov print(f'...').
   * Zvyšuje sa hodnota premennej zucastneny o 1.
   * Mená súťažiacich sa pridávajú do zoznamu mena a časy do zoznamu casy.
2. Časť: Hľadanie najlepšieho času a mena víťaza:
   * Premenná najlepsic sa inicializuje prvým časom zo zoznamu casy.
   * S použitím for cyklu sa prechádza každý čas zo zoznamu casy.
   * Porovnáva sa každý čas s aktuálnou hodnotou premennej najlepsic.
   * Ak je porovnávaný čas menší ako najlepsic, premenná najlepsic sa aktualizuje na tento čas.
   * Súbor "sutaz\_vbehu.txt" sa opäť otvára pre čítanie.
   * Premenná meno sa inicializuje na prázdny reťazec.
   * Prechádza sa každý riadok zo súboru rovnakým spôsobom ako v predchádzajúcej časti.
   * Ak sa čas rovná najlepsic, meno súťažiaceho sa priradí do premennej meno.
3. Časť: Výpis výsledkov:
   * Premenná najlepsic sa konvertuje na celé číslo pomocou int() (predpokladá sa, že ide o čas v sekundách).
   * Vypočítava sa hodnota premenných hodiny a sekund zo získaného najlepšieho času.
   * Vypisuje sa prázdna riadka a potom sa vypisujú celkové štatistiky súťaže pomocou print().

Celkovo program načítava údaje zo súboru, spracováva ich a vypisuje výsledky na obrazovku. Vypočítava počet súťažiacich, hľadá najlepší čas a mená víťazov. Potom vypisuje celkové štatistiky súťaže, vrátane mena víťaza a jeho času.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér, Informačná spoločnosť.

***Pomôcky:*** súbor sutaz\_vbehu.txt pre úlohu č.1

**Maturitné zadanie č. 5**

***Úloha č. 1 - Krasokorčuľovanie***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**import random**

**rozhocovia=int(input('Zadajte pocet rozhodcov: '))**

**sutaziaci=int(input('Zadajte pocet sutaziacich: '))**

**body=[]**

**celkovebodu=0**

**vytaz=[]**

**for s in range(sutaziaci):**

**for i in range(rozhocovia):**

**cislo=random.randrange(1,7)**

**body.append(cislo)**

**print('Body sutaziaceho',s+1,'su',body)**

**print('Sutaziacemu',s+1,'udelili najvacsiu znamku',max(body))**

**print('Sutaziacemu',s+1,'udelili najmensiu znamku',min(body))**

**for z in body:**

**if z != max(body) and z != min(body):**

**celkovebodu += z**

**vytaz.append(celkovebodu)**

**print('Dokopy mal sutaziaci',s+1,celkovebodu,'bodov')**

**celkovebodu=0**

**body=[]**

**vytazfin=max(vytaz)**

**vytazfinnal=vytaz.index(vytazfin)**

**print('Celkovy vytaz je',vytazfinnal+1)**

Tento program vykonáva simuláciu súťaže medzi súťažiacimi a rozhodcami a určuje celkové skóre a výhercu súťaže.

1. Importuje sa modul "random", ktorý umožňuje generovať náhodné čísla.
2. Štandardným vstupom sa zadáva počet rozhodcov (premenná "rozhocovia") a počet súťažiacich (premenná "sutaziaci").
3. Inicializuje sa prázdny zoznam "body", premenná "celkovebodu" s hodnotou 0 a zoznam "vytaz".
4. V prvom cykle "for" sa prechádza každý súťažiaci (premenná "s") a pre každého súťažiaceho sa generuje skóre od rozhodcov.
5. V druhom vnoreném cykle "for" sa prechádza každý rozhodca (premenná "i") a generuje sa náhodné číslo medzi 1 a 6 (vrátane). Toto číslo sa pridáva do zoznamu "body".
6. Vypíše sa skóre pre daného súťažiaceho.
7. Vypíše sa najvyššie a najnižšie skóre, ktoré boli pridelené súťažiacemu.
8. V treťom vnoreném cykle "for" sa prechádza každé skóre (premenná "z") v zozname "body". Ak sa skóre nerovná najvyššiemu ani najnižšiemu skóre, pridá sa k "celkovebodu".
9. Celkové skóre sa pridáva do zoznamu "vytaz".
10. Vypíše sa celkové skóre súťažiaceho.
11. Premenná "celkovebodu" sa resetuje na hodnotu 0 a zoznam "body" sa vyprázdňuje pre ďalšieho súťažiaceho.
12. Cyklus sa opakuje pre každého súťažiaceho.
13. Zoznam "vytaz" sa prechádza a určuje sa najvyššie skóre zo zoznamu a jeho index (víťaz).
14. Vypíše sa číslo víťaza.

Program generuje náhodné skóre pre súťažiacich od rozhodcov a vypočítava celkové skóre pre každého súťažiaceho. Nakoniec určuje a vypisuje víťaza na základe najvyššieho celkového skóre.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér, Komunikácia a spolupráca.

**Maturitné zadanie č. 6**

### *Úloha č.1 - Hlasovanie*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## z=open('hlasovanie\_1.txt','r')

## 

## sutaziaci=[5220,5221,5222,5223,5224,5225,5226,5227,5228,5229]

## hlasy=[0]\*10

## print(hlasy)

## hlas=0

## 

## for i in z:

## cisloa=(int(i))

## kde=sutaziaci.index(cisloa)

## hlasy[kde] += 1

## hlas += 1

## 

## print(hlasy)

## print(hlas)

## 

## min=hlasy[0]

## 

## for i in hlasy:

## if i < min:

## min = i

## 

## ktory=(hlasy.index(min))

## print(sutaziaci[ktory])

Tento program pracuje so súborom "hlasovanie\_1.txt", ktorý obsahuje čísla súťažiacich, ktorí dostali hlasy. Tu je komentár k jednotlivým častiam programu:

1. Časť: Načítanie údajov zo súboru a inicializácia premenných:
   * Otvára sa súbor "hlasovanie\_1.txt" pre čítanie pomocou funkcie open().
   * Vytvára sa zoznam sutaziaci, do ktorého sa ručne zadávajú identifikačné čísla súťažiacich.
   * Vytvára sa zoznam hlasy s počiatočnou hodnotou 0 pre každého súťažiaceho.
   * Vypisuje sa obsah zoznamu hlasy pomocou print().
2. Časť: Spracovanie hlasov a aktualizácia počtu hlasov pre jednotlivých súťažiacich:
   * S použitím for cyklu prechádza každý riadok zo súboru.
   * Každý riadok sa konvertuje na celé číslo pomocou int() a priradí sa do premennej cisloa.
   * S použitím metódy index() sa hľadá pozícia čísla cisloa v zozname sutaziaci.
   * Hodnota na tejto pozícii v zozname hlasy sa zvyšuje o 1.
   * Inkrementuje sa premenná hlas, ktorá slúži na zaznamenanie celkového počtu hlasov.
3. Časť: Výpis výsledkov:
   * Vypisuje sa obsah zoznamu hlasy pomocou print(), čo zobrazuje počet hlasov pre každého súťažiaceho.
   * Vypisuje sa hodnota premennej hlas, čo zobrazuje celkový počet hlasov.
4. Časť: Hľadanie súťažiaceho s najmenším počtom hlasov:
   * Inicializuje sa premenná min na prvú hodnotu v zozname hlasy.
   * S použitím for cyklu prechádza každú hodnotu v zozname hlasy.
   * Porovnáva sa každá hodnota s aktuálnou hodnotou premennej min.
   * Ak je porovnávaná hodnota menšia ako min, premenná min sa aktualizuje na túto hodnotu.
   * S použitím metódy index() sa hľadá pozícia hodnoty min v zozname hlasy.
   * Na tejto pozícii sa nachádza identifikačné číslo súťažiaceho v zozname sutaziaci, ktoré sa vypisuje pomocou print().

Celkovo program spracováva údaje zo súboru a zaznamenáva počet hlasov pre každého súťažiaceho. Potom vypisuje tieto výsledky na obrazovku a určuje súťažiaceho s najmenším počtom hlasov.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Reprezentácie a nástroje, Softvér a hardvér, Komunikácia a spolupráca.

***Pomôcky:*** súbor *hlasovanie.txt* pre úlohu č.1**Maturitné zadanie č. 7**

## *Úloha č.1 –Vírus*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## import random

## z=open('virus.txt','r')

## nove=[]

## koncovy=[]

## for riadok in z:

## nove=riadok.split()

## novy\_riadok=''

## for i in nove:

## nahoda1=random.randint(1,2)

## if nahoda1 == 1:

## i=i[::-1]

## novy\_riadok = novy\_riadok + i + ' '

## else:

## novy\_riadok = novy\_riadok + i + ' '

## koncovy.append(novy\_riadok)

## 

## koncovy2=[]

## print(koncovy)

## novy\_riadok=''

## for riadok in koncovy:

## nahodne2=random.randint(1,2)

## if nahodne2 == 1:

## nove2=riadok.split()

## random.shuffle(nove2)

## novy\_riadok=''

## for i in nove2:

## novy\_riadok = novy\_riadok + i + ' '

## print(novy\_riadok)

## koncovy2.append(novy\_riadok)

## else:

## print(riadok)

## koncovy2.append(riadok)

## print(koncovy2)

## nahodne3=random.randint(1,2)

## if nahodne3 == 1:

## random.shuffle(koncovy2)

## z.close()

## z=open('konecnyvirus.txt','w')

## for i in koncovy2:

## z.write(i)

## z.write('\n')

Tento program pracuje so súborom 'virus.txt' a vykonáva transformácie nad obsahom súboru.

1. Importuje sa modul "random", ktorý umožňuje generovať náhodné čísla.
2. Otvára sa súbor 'virus.txt' v režime čítania.
3. Inicializujú sa prázdne zoznamy "nove" a "koncovy".

Pre každý riadok v súbore sa vykoná nasledujúce:

a. Riadok sa rozdelí na slová pomocou metódy "split()" a uloží sa do zoznamu "nove".

b. Inicializuje sa prázdny reťazec "novy\_riadok".

1. c. Pre každé slovo "i" v zozname "nove" sa generuje náhodné číslo medzi 1 a 2 (vrátane).
   * Ak je vygenerované číslo rovné 1, slovo "i" sa prehádza (mení poradie znakov) a pridáva sa do "novy\_riadok".

Ak je vygenerované číslo iné ako 1, slovo "i" sa pridáva do "novy\_riadok" bez zmeny.

* + d. Nový riadok "novy\_riadok" sa pridáva do zoznamu "koncovy".

1. Inicializuje sa prázdny zoznam "koncovy2".
2. Vypíše sa obsah zoznamu "koncovy".

Pre každý riadok v zozname "koncovy" sa vykoná nasledujúce:

a. Generuje sa náhodné číslo medzi 1 a 2 (vrátane).

1. b. Ak je vygenerované číslo rovné 1, vykoná sa nasledujúce:
   * Riadok sa rozdelí na slová pomocou metódy "split()" a zoznam "nove2" dostane tieto slová.
   * Slová v zozname "nove2" sa náhodne zamiešajú pomocou funkcie "random.shuffle".
   * Inicializuje sa prázdny reťazec "novy\_riadok".

Pre každé slovo "i" v zozname "nove2" sa pridáva do "novy\_riadok".

c. Ak je vygenerované číslo iné ako 1, riadok sa vypíše bez zmeny.

* + d. Nový riadok "novy\_riadok" sa pridáva do zoznamu "koncovy2".

1. Vypíše sa obsah zoznamu "koncovy2".
2. Generuje sa náhodné číslo medzi 1 a 2 (vrátane) a uloží sa do premennej "nahodne3".
3. Ak je vygenerované číslo rovné 1, zoznam "koncovy2" sa náhodne zamieša pomocou funkcie "random.shuffle".
4. Súbor 'virus.txt' sa zatvorí.
5. Otvára sa súbor 'konecnyvirus.txt' v režime zápisu.

13. Pre každý riadok v zozname "koncovy2" sa vykoná nasledujúce:

a. Riadok "i" sa zapíše do súboru.

b. Na koniec riadku sa pridá znak nového riadku '\n'.

Program vykonáva rôzne transformácie nad obsahom súboru 'virus.txt'. Pre každý riadok sa vykoná zamiešanie a zmena poradia slov, a potom sa výsledok zapíše do súboru 'konecnyvirus.txt'.

## *Úloha č.2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér, Komunikácia a spolupráca.

***Pomôcky:*** súbor *virus.txt* pre úlohu č.1**Maturitné zadanie č.8**

## *Úloha č.1 – Evidencia krúžku*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**subor=open('rodnec.txt','r')**

**dievcata=0**

**clenov=0**

**starsi=0**

**for i in subor:**

**print(i,end='')**

**clenov += 1**

**if i[2] == '5' or i[2] == '6':**

**dievcata += 1**

**print('V kruzku je clenov: ',clenov)**

**print('V kruzku je dievcat: ',dievcata)**

**subor.close**

**subor=open('rodnec.txt','a')**

**subor.write('\n')**

**subor.write(f'Dievcat je{dievcata}')**

Tento program pracuje so súborom "rodnec.txt", ktorý obsahuje záznamy o členoch kruzku. Tu je komentár k jednotlivým častiam programu:

1. Časť: Načítanie údajov zo súboru a spracovanie členov kruzku:
   * Otvára sa súbor "rodnec.txt" pre čítanie pomocou funkcie open().
   * Inicializujú sa premenné dievcata, clenov a starsi na hodnotu 0. Tieto premenné budú slúžiť na počítanie počtu dievčat, počtu členov a počtu starších členov.
   * S použitím for cyklu prechádza sa každý riadok zo súboru.
   * Vypisuje sa obsah riadku pomocou print(i, end='').
   * Zvyšuje sa hodnota premennej clenov o 1.
   * Ak číslo na pozícii indexu 2 riadku (tretí znak) je '5' alebo '6', zvyšuje sa hodnota premennej dievcata o 1.
2. Časť: Výpis výsledkov:
   * Vypisuje sa počet členov kruzku pomocou print('V kruzku je clenov: ', clenov).
   * Vypisuje sa počet dievčat v kruzku pomocou print('V kruzku je dievcat: ', dievcata).
3. Časť: Uzavretie súboru a zápis informácie o dievčatách:
   * Súbor "rodnec.txt" sa uzatvára pomocou subor.close().
   * Súbor sa opäť otvára, tentoraz v režime pridávania ('a' - append) pomocou subor=open('rodnec.txt','a').
   * Na nový riadok sa zapíše oddeľovač pomocou subor.write('\n').
   * Do súboru sa zapíše informácia o počte dievčat pomocou formátovanej reťazcov subor.write(f'Dievcat je {dievcata}').

Celkovo program spracováva údaje zo súboru, počíta počet členov a dievčat v kruzku a vypisuje tieto informácie na obrazovku. Potom do súboru "rodnec.txt" pridáva informáciu o počte dievčat.

***Úloha č. 2***

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér, Komunikácia a spolupráca, Informačná spoločnosť.

***Pomôcky:*** súbor *rodne\_cisla.txt* pre úlohu č.1**Maturitné zadanie č. 9**

### *Úloha č. 1 – Tajná tabuľka*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## veta=input("zadaj vetu:")

## for znak in veta:

## if znak==" ":

## print("0",end=" ")

## else:

## oc=ord(znak) - 65

## pol=(oc//3)+1

## pocet=(oc%3)+1

## print(pocet\*str(pol),end=" ")

Tento program pracuje s vetou zadanou používateľom a vykonáva šifrovanie znakov na základe ich poradia v abecede.

1. Štandardným vstupom sa zadáva veta (premenná "veta").

Pre každý znak ("znak") vo vete sa vykoná nasledujúce:

a. Ak je znak rovný medzere, vypíše sa "0" a na konci sa pridá medzera (" ", nastavením argumentu "end" na " ").

1. b. Inak, sa vykoná nasledujúce:
   * Znak "znak" sa prevádza na jeho kód ASCII pomocou funkcie "ord" a odčíta sa hodnota 65, aby sa získal index v abecede.
   * Výsledok sa ukladá do premennej "oc".
   * Vypočíta sa hodnota polovice celočíselného delenia "oc" s 3, ktorá určuje v akom poli v šifre sa znak nachádza.
   * Vypočíta sa hodnota zvyšku po delení "oc" s 3, ktorá určuje koľkokrát sa táto hodnota v šifre opakuje.
   * Vypíše sa "pocet" krát reťazec "pol" (číslo pola v šifre) a na konci sa pridá medzera (" ", nastavením argumentu "end" na " ").

Program šifruje znaky vo vete na základe ich polohy v abecede a vypisuje šifrované hodnoty. Medzery sú nahradené číslom 0.

## 

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Komunikácia a spolupráca, Informačná spoločnosť.

**Maturitné zadanie č. 10**

## *Úloha č.1 - Lotéria*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**import random**

**import time**

**nase=[]**

**loteria=[]**

**for i in range(6):**

**tip=(int(input('Zadajte svoj tip')))**

**while tip in nase:**

**tip=(int(input('Zadajte svoj tip ale iny nez ten ktory uz je v loterii')))**

**nase.append(tip)**

**for i in range(6):**

**tip=random.randint(1,10)**

**while tip in loteria:**

**tip=random.randint(1,10)**

**print(i+1,'cislo ktore sa vylosovalo je',tip)**

**loteria.append(tip)**

**time.sleep(1)**

**for i in loteria:**

**if i in nase:**

**print('Uhadli ste cislo',i)**

**print(loteria)**

**print(nase)**

Tento program je simuláciou lotérie, v ktorej používateľ zadáva svoje tipy a potom sa vylosovaním náhodných čísel generuje výsledok. Tu je komentár k jednotlivým častiam programu:

1. Importovanie modulov:
   * Importuje sa modul random, ktorý slúži na generovanie náhodných čísel.
   * Importuje sa modul time, ktorý slúži na oneskorenie (pauzu) medzi jednotlivými krokmi lotérie.
2. Časť: Zadávanie používateľských tipov:
   * Inicializuje sa prázdny zoznam nase, do ktorého budú ukladané tipy používateľa.
   * V cykle for sa opakuje 6-krát:
     + Používateľ zadáva svoj tip cez funkciu input().
     + Skontroluje sa, či je tento tip už obsiahnutý v zozname nase. Ak áno, používateľ je vyzvaný, aby zadal iný tip.
     + Ak je tip unikátny, pridá sa do zoznamu nase.
3. Časť: Generovanie výsledkov lotérie:
   * Inicializuje sa prázdny zoznam loteria, do ktorého budú ukladané vylosované čísla.
   * V cykle for sa opakuje 6-krát:
     + Generuje sa náhodné číslo medzi 1 a 10 pomocou funkcie random.randint().
     + Skontroluje sa, či je vygenerované číslo už obsiahnuté v zozname loteria. Ak áno, generuje sa nové číslo.
     + Vypisuje sa poradové číslo a vylosované číslo.
     + Pridá sa vylosované číslo do zoznamu loteria.
     + Použije sa funkcia time.sleep(1) na vytvorenie jednosekundovej pauzy medzi každým vylosovaným číslom.
4. Časť: Kontrola zhody tipov s vylosovanými číslami:
   * V cykle for sa prechádza každé vylosované číslo i v zozname loteria.
   * Skontroluje sa, či sa toto číslo nachádza aj v zozname nase. Ak áno, vypíše sa, že používateľ uhádol dané číslo.
5. Časť: Výpis výsledkov:
   * Vypisuje sa obsah zoznamu loteria, čo zobrazuje vylosované čísla lotérie.
   * Vypisuje sa obsah zoznamu nase, čo zobrazuje tipy používateľa.

Celkovo program umožňuje používateľovi zadať svoje tipy, vygeneruje výsledky lotérie, porovná tipy s vylosovanými číslami a vypíše výsledky na obrazovku.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Komunikácia a spolupráca, Informačná spoločnosť.

***Pomôcky:***  súbor *loteria.txt* pre úlohu č.1**Maturitné zadanie č. 11**

***Úloha č. 1 - Výplaty zamestnancov***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## z=open('platy.txt','r')

## platy=[]

## zamestanec=0

## 

## for i in z:

## kde=i.find(',')

## plat=i[kde+1:]

## plat=plat.strip()

## platy.append(int(plat))

## zamestanec += 1

## 

## print(platy)

## 

## najmenej=platy[0]

## 

## for i in platy:

## if i < najmenej:

## najmenej = i

## 

## print(f'Firma ma {zamestanec} zamestnancov')

## z.close()

## 

## z=open('platy.txt','r')

## kto=''

## for y in z:

## kde=y.find(',')

## plat=y[kde+1:]

## plat=plat.strip()

## plat=int(plat)

## if plat == najmenej:

## kto=y[:kde]

## 

## print(f'Najmesi plat je {najmenej} a ma ho {kto}')

Tento program pracuje so súborom 'platy.txt' obsahujúcim zoznam zamestnancov a ich platov.

1. Otvára sa súbor 'platy.txt' v režime čítania.
2. Inicializuje sa prázdny zoznam "platy" a premenná "zamestanec" na hodnotu 0.

Pre každý riadok "i" v súbore sa vykoná nasledujúce:

a. Hľadá sa pozícia čiarky pomocou metódy "find(',')".

b. Zistí sa hodnota platu ("plat") zo substringu od pozície čiarky + 1 až do konca riadku.

c. Odstráni sa prípadný medzerový znak na začiatku alebo konci platu pomocou metódy "strip()".

d. Hodnota platu sa prevádza na celé číslo a pridáva sa do zoznamu "platy".

1. e. Zvýši sa hodnota premennej "zamestanec" o 1.
2. Vypíše sa obsah zoznamu "platy".
3. Inicializuje sa premenná "najmenej" na hodnotu prvého platu v zozname "platy".

Pre každý plat "i" v zozname "platy" sa vykoná nasledujúce:

1. a. Ak je plat "i" menší ako hodnota premennej "najmenej", nastaví sa "najmenej" na hodnotu platu "i".
2. Vypíše sa správa obsahujúca počet zamestnancov "zamestanec" vo firme.
3. Súbor 'platy.txt' sa zatvorí.
4. Súbor 'platy.txt' sa opäť otvára v režime čítania.
5. Inicializuje sa prázdny reťazec "kto".

Pre každý riadok "y" v súbore sa vykoná nasledujúce:

a. Hľadá sa pozícia čiarky pomocou metódy "find(',')".

b. Zistí sa hodnota platu ("plat") zo substringu od pozície čiarky + 1 až do konca riadku.

c. Odstráni sa prípadný medzerový znak na začiatku alebo konci platu pomocou metódy "strip()".

d. Plat sa prevádza na celé číslo.

1. e. Ak je plat rovný hodnote "najmenej", uloží sa do premennej "kto" meno zo substringu od začiatku riadku do pozície čiarky.
2. Vypíše sa správa obsahujúca najmenší plat "najmenej" a kto ho má.

Program načíta platy zamestnancov zo súboru 'platy.txt' a vypíše ich. Potom nájde najmenší plat a zistí, kto ho má.

## *b)*

## z=open('platy.txt','r')

## mena=[]

## platy=[]

## for i in z:

## hladat=i.find(',')

## meno=(i[:hladat])

## plat=(i[hladat+1:])

## plat=plat.strip()

## plat=int(plat)

## mena.append(meno)

## platy.append(plat)

## 

## 

## print(mena,platy)

## 

## max=0

## 

## for i in platy:

## if i > max:

## max = i

## 

## print(max)

## najviac=platy.index(max)

## ktory=mena[najviac]

## print(f'Najviac si zarobil {ktory} ciastku: {max}')

Tento program pracuje so súborom 'platy.txt', kde sa nachádzajú mená zamestnancov a ich príslušné platy.

1. Otvára sa súbor 'platy.txt' v režime čítania.
2. Inicializujú sa prázdne zoznamy "mena" a "platy".

Pre každý riadok "i" v súbore sa vykoná nasledujúce:

a. Hľadá sa pozícia čiarky pomocou metódy "find(',')".

b. Meno zamestnanca sa získava zo substringu od začiatku riadku do pozície čiarky.

c. Plat zamestnanca sa získava zo substringu od pozície čiarky + 1 do konca riadku.

d. Odstráni sa prípadný medzerový znak na začiatku alebo konci platu pomocou metódy "strip()".

e. Plat sa prevádza na celé číslo a pridáva sa do zoznamu "platy".

1. f. Meno sa pridáva do zoznamu "mena".
2. Vypíšu sa zoznamy "mena" a "platy".
3. Inicializuje sa premenná "max" na hodnotu 0.

Pre každý plat "i" v zozname "platy" sa vykoná nasledujúce:

1. a. Ak je plat "i" väčší ako hodnota premennej "max", nastaví sa "max" na hodnotu platu "i".
2. Vypíše sa hodnota premennej "max", čo predstavuje najvyšší plat.
3. Zistí sa index zoznamu "platy", kde sa nachádza najvyšší plat ("max").
4. Zistí sa meno zamestnanca, ktorý má najvyšší plat, získaním hodnoty z rovnakého indexu v zozname "mena".
5. Vypíše sa správa obsahujúca meno zamestnanca a jeho najvyšší plat.

Program načíta mená a platy zamestnancov zo súboru 'platy.txt' a vypíše ich. Potom zistí, kto zamestnancov zarobil najviac a vypíše jeho meno a príslušný plat.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Softvér a hardvér, Komunikácia a spolupráca, Informačná spoločnosť.

***Pomôcky:*** súbor *firma.txt* pre úlohu č.1

**Maturitné zadanie č. 12**

## *Úloha č.1 – Mapa krajiny*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**import zlib**

**z=open('krajina.txt','r')**

**moria=[]**

**mm=0**

**niziny=[]**

**nn=0**

**pohoria=[]**

**pp=0**

**for i in z:**

**if int(i) == 0:**

**moria.append(int(i))**

**mm += 1**

**elif int(i) > 0 and int(i) < 300:**

**niziny.append(int(i))**

**nn += 1**

**elif int(i) > 300:**

**pohoria.append(int(i))**

**pp += 1**

**print(moria,niziny,pohoria)**

**print('Mori je:',mm)**

**print('Nizin je:',nn)**

**print('Pohorii je:',pp)**

**maxp=0**

**for i in pohoria:**

**if i > maxp:**

**maxp = i**

**print('Najvacsie pohorie je',maxp)**

**minn=niziny[0]**

**for i in range(1,len(niziny)):**

**if niziny[i] < minn:**

**minn = niziny[i]**

**print('Najnizsia nizina ma vysku',minn)**

Tento program pracuje so súborom "krajina.txt", v ktorom sú uložené výšky rôznych oblastí. Tu je komentár k jednotlivým častiam programu:

1. Importovanie modulu:
   * Importuje sa modul zlib, ktorý poskytuje funkcie na kompresiu a dekompresiu dát. V tomto prípade sa však zdá, že modul nie je potrebný, pretože sa v programe nepoužíva.
2. Časť: Načítanie údajov zo súboru a triedenie do príslušných kategórií:
   * Otvára sa súbor "krajina.txt" pre čítanie pomocou funkcie open().
   * Inicializujú sa prázdne zoznamy moria, niziny a pohoria, do ktorých budú zoradené príslušné výšky.
   * Inicializujú sa počítadlá mm, nn a pp na hodnotu 0, ktoré budú slúžiť na počítanie počtu výšok v jednotlivých kategóriách.
   * S použitím for cyklu prechádza sa každý riadok zo súboru.
   * Ak je výška na riadku rovná 0, pridá sa do zoznamu moria a zvýši sa počítadlo mm.
   * Ak je výška na riadku väčšia ako 0 a menšia ako 300, pridá sa do zoznamu niziny a zvýši sa počítadlo nn.
   * Ak je výška na riadku väčšia ako 300, pridá sa do zoznamu pohoria a zvýši sa počítadlo pp.
3. Časť: Výpis výsledkov:
   * Vypisujú sa obsahy zoznamov moria, niziny a pohoria, čo zobrazuje výšky v jednotlivých kategóriách.
   * Vypisuje sa počet prvkov v zozname moria pomocou premenného mm.
   * Vypisuje sa počet prvkov v zozname niziny pomocou premenného nn.
   * Vypisuje sa počet prvkov v zozname pohoria pomocou premenného pp.
4. Časť: Vyhľadanie najväčšieho pohoria:
   * Inicializuje sa premenná maxp na hodnotu 0, ktorá bude slúžiť na uchovávanie najväčšej výšky pohoria.
   * V cykle for sa prechádza každá výška i v zozname pohoria.
   * Ak je výška i väčšia ako maxp, maxp sa aktualizuje na hodnotu i.
5. Časť: Vyhľadanie najnižšej nížiny:
   * Inicializuje sa premenná minn na hodnotu prvej výšky v zozname niziny.
   * V cykle for sa prechádzajú výšky v zozname niziny od druhého prvku po posledný.
   * Ak je výška i menšia ako minn, minn sa aktualizuje na hodnotu i.
6. Vypisuje sa najväčšia výška pohoria (maxp) a najnižšia výška nížiny (minn).

Celkovo program triedi výšky do rôznych kategórií a potom vypisuje tieto kategórie a počty prvkov v nich. Taktiež nájde najväčšiu výšku v pohoriach a najnižšiu výšku v nížinách.

***Úloha č.2***

Prierezová úloha, ktorá zasahuje do tematických okruhov: Reprezentácie a nástroje, Softvér a hardvér, Komunikácia a spolupráca.

***Pomôcky:*** súbor krajina.txt pre úlohu č.1**Maturitné zadanie č.13**

***Úloha č.1 - Skoky do výšky***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**ziakov=int(input('Kolko ziakov skakalo: '))**

**skoky=[]**

**for i in range(ziakov):**

**skocil=int(input(f'Ziak {i+1} skocil kolko: '))**

**skoky.append(skocil)**

**maxskos=0**

**pocetmax=0**

**for i in skoky:**

**if i > maxskos:**

**maxskos = i**

**pocetmax = 0**

**if i==maxskos:**

**pocetmax += 1**

**print('Najdlhsi skok bol',maxskos)**

**print('Najdlhsi skok bol skoceny',pocetmax)**

**print('Prvy kto skocil najdalej je cislo',skoky.index(maxskos)+1)**

**#kolky=1**

**#for i in skoky:**

**#if i == maxskos:**

**#break**

**#kolky += 1**

**#print(kolky)**

Tento program zbiera informácie o skokoch žiakov a vykoná ich analýzu.

1. Načíta sa počet žiakov, ktorí skákali, prostredníctvom funkcie "input".
2. Inicializuje sa prázdny zoznam "skoky", do ktorého budú uložené hodnoty skokov žiakov.

Pre každého žiaka v rozsahu od 0 po (ziakov-1) sa vykoná nasledujúce:

a. Načíta sa hodnota skoku žiaka "skocil" prostredníctvom funkcie "input".

1. b. Hodnota skoku sa pridá do zoznamu "skoky".
2. Inicializujú sa premenné "maxskos" a "pocetmax" na hodnotu 0.

Pre každý skok "i" v zozname "skoky" sa vykoná nasledujúce:

a. Ak je skok "i" väčší ako hodnota premennej "maxskos", nastaví sa "maxskos" na hodnotu skoku "i" a "pocetmax" na 0.

1. b. Ak je skok "i" rovnaký ako hodnota premennej "maxskos", zvýši sa hodnota premennej "pocetmax" o 1.
2. Vypíše sa správa obsahujúca najdlhší skok.
3. Vypíše sa správa obsahujúca, koľkokrát bol dosiahnutý najdlhší skok.
4. Vypíše sa správa obsahujúca číslo žiaka, ktorý dosiahol najdlhší skok, pričom sa zistí jeho index v zozname "skoky" a pridá sa 1.

Program zbiera informácie o skokoch žiakov a vypisuje výsledky, ako je najdlhší skok, počet žiakov, ktorí dosiahli najdlhší skok, a prvého žiaka, ktorý dosiahol najdlhší skok.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Komunikácia a spolupráca, Informačná spoločnosť.

**Maturitné zadanie č. 14.**

***Úloha č.1 - Šetrič obrazovky***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**import tkinter**

**import random**

**canvas=tkinter.Canvas(bg='black', width=1200, height=800)**

**canvas.pack()**

**p=1**

**def setric():**

**if p==1:**

**sirka = random.randint(1, 1200)**

**vyska = random.randint(1, 800)**

**x = random.randint(1, 370)**

**y = random.randint(1, 260)**

**farba=f'#{random.randrange(256\*\*3):06x}'**

**canvas.create\_oval(x, y, x + sirka, y + vyska, fill=farba)**

**sirka2=random.randint(10, 1200)**

**vyska2=random.randint(10, 800)**

**x = random.randint(1, 370)**

**y = random.randint(1, 260)**

**farb=f'#{random.randrange(256\*\*3):06x}'**

**canvas.create\_rectangle(x, y, x + sirka, y + vyska, fill=farb)**

**canvas.after(800, setric)**

**def stop(event):**

**global p**

**if p==1:**

**p=0**

**elif p==0:**

**p=1**

**canvas.after(50, setric)**

**setric()**

**canvas.bind\_all('p',stop)**

Tento program využíva knižnicu tkinter na vytvorenie grafického okna a vykresľovanie tvarov v ňom. Tu je komentár k jednotlivým častiam programu:

1. Importovanie modulov:
   * Importuje sa modul tkinter, ktorý poskytuje nástroje pre tvorbu grafického rozhrania.
   * Importuje sa modul random, ktorý poskytuje funkcie pre generovanie náhodných čísel.
2. Vytvorenie plátna:
   * Vytvára sa plátno s čiernym pozadím a rozmermi 1200x800 pixelov pomocou tkinter.Canvas().
   * Plátno sa zobrazuje pomocou metódy pack().
3. Inicializácia premenných:
   * Inicializuje sa premenná p s hodnotou 1. Táto premenná bude slúžiť na spúšťanie a zastavovanie animácie.
4. Funkcia setric():
   * Definuje sa funkcia setric(), ktorá vykresľuje náhodné tvary na plátne.
   * Ak je premenná p nastavená na 1, vykoná sa nasledujúci kód.
   * Generujú sa náhodné hodnoty pre šírku, výšku, súradnice x a y, a farbu tvarov.
   * Na plátno sa vykreslí ovál so zadanými parametrami.
   * Opäť sa generujú náhodné hodnoty pre šírku, výšku, súradnice x a y a farbu pre obdĺžnik.
   * Na plátno sa vykreslí obdĺžnik so zadanými parametrami.
   * Pomocou canvas.after() sa volá funkcia setric() s oneskorením 800 milisekúnd.
5. Funkcia stop():
   * Definuje sa funkcia stop(), ktorá zastavuje a opätovne spúšťa animáciu.
   * Ak je stlačená klávesa "p" a premenná p je nastavená na 1, premenná p sa zmení na 0.
   * Ak je premenná p nastavená na 0, premenná p sa zmení na 1 a pomocou canvas.after() sa volá funkcia setric() s oneskorením 50 milisekúnd.
6. Spustenie animácie:
   * Volá sa funkcia setric(), ktorá začína vykresľovať tvary na plátne.
   * Plátno sa viaže na udalosť stlačenia klávesy "p" pomocou canvas.bind\_all().
   * Pri stlačení klávesy "p" sa vyvolá funkcia stop(), ktorá zastaví alebo spustí animáciu v závislosti od hodnoty premennej p.

Celkovo program vytvára grafické okno s animáciou, v ktorej sa zobrazujú náhodné tvary (ovály a obdĺžniky) na plátne. Animácia sa spúšťa a zastavuje stlačením klávesy "p".

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér, Komunikácia a spolupráca, Informačná spoločnosť.

***Pomôcky:*** súbor *stranka.html* pre úlohu č.1**Maturitné zadania č.15**

***Úloha č.1 - Pošta***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## z=open('telegram.txt','r')

## cena=0

## vety=0

## slov=0

## for riadok in z:

## for znak in riadok:

## if znak == ' ':

## cena += 10

## slov += 1

## elif znak =='.':

## cena += 10

## slov += 1

## vety += 1

## if vety >= 10 and vety <= 20:

## cena += 100

## elif vety > 20:

## vety += 150

## elif vety < 10:

## cena += 60

## print(cena,vety,slov)

Tento program pracuje so súborom 'telegram.txt', kde sa nachádza telegrafická správa. Vykonáva analýzu správy a vypočíta jej cenu, počet viet a počet slov.

1. Otvára sa súbor 'telegram.txt' v režime čítania.
2. Inicializuje sa premenná "cena" na hodnotu 0, ktorá bude slúžiť na uchovávanie ceny správy.
3. Inicializujú sa premenné "vety" a "slov" na hodnotu 0, ktoré budú slúžiť na uchovávanie počtu viet a počtu slov v správe.

Pre každý riadok "riadok" v súbore sa vykoná nasledujúce:

1. a. Pre každý znak "znak" v riadku sa vykoná nasledujúce:
   * Ak je znak medzera (' '), zvýši sa cena o 10 a zvýši sa počet slov o 1.
   * Ak je znak bodka ('.'), zvýši sa cena o 10, zvýši sa počet slov o 1 a zvýši sa počet viet o 1.
2. Podmienky sú použité na pridanie príplatku k cene v závislosti od počtu viet:
   * Ak je počet viet medzi 10 a 20 (vrátane), pridá sa k cene 100.
   * Ak je počet viet väčší ako 20, pridá sa k cene 150.
   * Ak je počet viet menší ako 10, pridá sa k cene 60.
3. Vypíše sa cena správy, počet viet a počet slov.

Program analýzuje telegrafickú správu zo súboru 'telegram.txt' a vypočíta jej cenu na základe počtu viet, slov a príplatkových podmienok. Následne vypíše výsledky.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Komunikácia a spolupráca, Informačná spoločnosť.

***Pomôcky:*** súbor *Telegram.txt* pre úlohu č.1 a *zamestnanci.xls* pre úlohu č.2**Maturitné zadanie č.16**

## *Úloha č.1 – Tabuľka početnosti*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## z=open('tabulka\_pocetnosti.txt','r')

## kolko=[0]\*26

## for riadok in z:

## for pismeno in riadok:

## pismeno=pismeno.upper()

## if 'A' <= pismeno <= 'Z':

## kolko[ord(pismeno)-65] += 1

## print(kolko)

## for i in range(26):

## if kolko[i] != 0:

## print(f'Pismeno {chr(i+65)} je v texte {kolko[i]} krat')

## elif kolko[i] == 0:

## print(f'Pismeno {chr(65 + i)} v texte bolo 0 krat')

Tento program číta obsah súboru 'tabulka\_pocetnosti.txt' a zisťuje početností písmen v texte. Tu je komentár k jednotlivým častiam programu:

1. Otvorenie súboru:
   * Súbor 'tabulka\_pocetnosti.txt' sa otvára v režime na čítanie pomocou open('tabulka\_pocetnosti.txt', 'r').
   * Otvorený súbor je priradený k premenným z.
2. Inicializácia premenných:
   * Vytvára sa prázdny zoznam kolko s dĺžkou 26, ktorý bude slúžiť na uchovávanie početností písmen.
   * Každá položka v zozname je inicializovaná na hodnotu 0 pomocou [0]\*26.
3. Prechádzanie riadkami súboru:
   * Cyklus for prechádza každý riadok v súbore z.
   * Vnorený cyklus for prechádza jednotlivé písmená v každom riadku.
   * Každé písmeno sa prevedie na veľké písmeno pomocou pismeno.upper().
   * Ak je písmeno v rozsahu 'A' až 'Z', zvýši sa príslušný počet v zozname kolko o 1.
   * Početnosti písmen sa ukladajú na indexe odpovedajúcom poradovému číslu písmena v abecede.
4. Výpis početností písmen:
   * Vypíše sa zoznam kolko, ktorý obsahuje početnosti písmen v texte.
   * V cykle for sa prechádza každá položka v rozsahu 0 až 25 (pre 26 písmen v abecede).
   * Ak početnosť písmena nie je 0, vypíše sa správa o početnosti písmena s použitím print().
   * Ak početnosť písmena je 0, vypíše sa správa o tom, že písmeno sa v texte nevyskytuje.

Celkovo program analyzuje textový súbor a vypočítava početnosti písmen v texte. Výsledky sú zaznamenané v zozname kolko a vypisované na výstup.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér, Komunikácia a spolupráca.

***Pomôcky:*** súbor *známky.xls* pre úlohu č.2, súbor *tabulka\_pocetnosti.txt* pre úlohu č.1**Maturitné zadanie č. 17**

## *Úloha č. 1- Zašifrovaný text*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## text=open('vstupny\_text.txt','r')

## kolko=int(input('Zadajte o kolko by sa text mal posunut: '))

## novy=''

## for riadok in text:

## for znak in riadok:

## znak=znak.lower()

## if 'a' <= znak <= 'z':

## znak=ord(znak)+kolko

## if znak > 122:

## znak= 97 + (znak % 122)

## znak=chr(znak)

## novy+= znak

## else:

## znak=chr(znak)

## novy+= znak

## elif znak == ' ':

## novy += ' '

## 

## print(novy)

Tento program pracuje so súborom 'vstupny\_text.txt' obsahujúcim vstupný text. Požaduje od používateľa vstup, koľko miest by sa mal text posunúť. Následne vykonáva posunutie písmen v texte o zadaný počet miest a vypisuje výsledný posunutý text.

1. Otvára sa súbor 'vstupny\_text.txt' v režime čítania.
2. Používateľovi sa zobrazí otázka na vstup, koľko miest by sa mal text posunúť. Hodnota je uložená do premennej "kolko".
3. Inicializuje sa prázdny reťazec "novy", do ktorého budú postupne pridávané posunuté znaky.

Pre každý riadok "riadok" vo vstupnom texte sa vykoná nasledujúce:

1. a. Pre každý znak "znak" v riadku sa vykoná nasledujúce:
   * Znak sa prevádza na malé písmeno pomocou funkcie "lower()".
   * Ak je znak písmeno od 'a' po 'z', vykoná sa posunutie znaku o hodnotu "kolko".
     + Ak je výsledná hodnota väčšia ako 122 (ord('z')), znak sa presunie na začiatok abecedy.
     + Vypočíta sa nová hodnota znaku a prevedie sa späť na znak pomocou funkcie "chr()".
     + Nový znak sa pridá do reťazca "novy".
   * Ak je znak medzera (' '), ponechá sa nezmenený a pridá sa do reťazca "novy".
2. Vypíše sa výsledný posunutý text uložený v reťazci "novy".

Program vykonáva posunutie písmen v texte o zadaný počet miest (cyklicky v rámci abecedy) a vypisuje výsledný posunutý text.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Reprezentácie a nástroje, Softvér a hardvér, Informačná spoločnosť.

## *Pomôcky:* súbor *vstupny\_text.txt* pre úlohu č.1

## 

## 

## 

## 

## 

## *Maturitne zadanie c.18*

## *Úloha č. 1 – BMI index*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**z=open('BMI.txt','r')**

**kolko=0**

**for i in z:**

**kde=i.find(';')**

**vaha=(int(i[:kde]))**

**vyska=(int(i[kde+1:]))**

**bmi=(vaha/(vyska/100)\*\*2)**

**if bmi >= 18.5 and bmi < 25:**

**print('Ziak ma optymalnu vahu')**

**elif bmi < 18.5:**

**kolko += 1**

**print('Ziak ma podvahu')**

**elif bmi >= 18.5:**

**print('Ziak ma nadvahu')**

**print(f'Povahu ma {kolko} ziakov')**

Tento program číta obsah súboru 'BMI.txt' a vypočítava hodnoty BMI (Body Mass Index) pre každého žiaka. Tu je komentár k jednotlivým častiam programu:

1. Otvorenie súboru:
   * Súbor 'BMI.txt' sa otvára v režime na čítanie pomocou open('BMI.txt', 'r').
   * Otvorený súbor je priradený k premennej z.
2. Inicializácia premenných:
   * Premenná kolko je inicializovaná na hodnotu 0 a slúži na počítanie žiakov s podváhou.
3. Prechádzanie riadkami súboru:
   * Cyklus for prechádza každý riadok v súbore z.
   * Na každom riadku sa vyhľadá znak ';' pomocou i.find(';'), ktorý určuje oddelenie váhy a výšky.
   * Hodnota váhy sa získava zo začiatku riadku a prevedie sa na celé číslo pomocou int(i[:kde]).
   * Hodnota výšky sa získava za znakom ';' a rovnako sa prevedie na celé číslo pomocou int(i[kde+1:]).
   * Výpočet hodnoty BMI sa vykonáva pomocou vzorca vaha / (vyska / 100) \*\* 2, kde výška sa prevádza na metre.
4. Vyhodnotenie hodnoty BMI:
   * Podľa hodnoty BMI sa vypisuje správa o stavbe tela žiaka.
   * Ak je BMI medzi 18.5 a 25, vypíše sa správa "Ziak ma optimálnu váhu".
   * Ak je BMI menšie ako 18.5, zvýši sa hodnota premennej kolko o 1 a vypíše sa správa "Ziak ma podváhu".
   * Ak je BMI väčšie alebo rovné 25, vypíše sa správa "Ziak ma nadváhu".
5. Výpis počtu žiakov s podváhou:
   * Na konci programu sa vypíše počet žiakov, ktorí majú podváhu, pomocou f-stringu f'Povahu ma {kolko} žiakov'.

Celkovo program spracováva údaje o váhe a výške žiakov, vypočítava BMI a vypisuje správy o stavbe tela. Na záver vypíše počet žiakov s podváhou.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Softvér a hardvér, Informačná spoločnosť.

***Pomôcky:*** súbor *BMI.txt* pre úlohu č.1**Maturitné zadanie č. 19**

***Úloha č. 1 – Delenie***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**a=int(input('Citatel: '))**

**b=int(input('Menovatel: '))**

**vysledok=a/b**

**aa=[]**

**bb=[]**

**for i in range(1,a+1):**

**if a % i == 0:**

**aa.append(i)**

**for i in range(1,b+1):**

**if b % i == 0:**

**bb.append(i)**

**aa=aa[::-1]**

**bb=bb[::-1]**

**print(aa,bb)**

**max=0**

**for i in aa:**

**if i in bb:**

**max=i**

**break**

**print(max)**

**print(vysledok)**

Tento program pracuje s dvoma vstupnými číslami citatel (a) a menovatel (b), a vypočítava ich podiel. Následne vyhľadáva najväčší spoločný deliteľ medzi číslami a vypisuje ho spolu s výsledkom podielu.

1. Používateľovi sa zobrazujú otázky na vstup, pričom si zisťuje hodnoty citatela a menovateľa. Hodnoty sa uložia do premenných "a" a "b".
2. Vypočíta sa výsledok podielu čísel a/b a uloží sa do premennej "vysledok".
3. Vytvárajú sa prázdne zoznamy "aa" a "bb", do ktorých budú postupne pridávané delitele citatela a menovateľa.

Pre každé číslo "i" od 1 po "a" sa vykoná nasledujúce:

1. a. Ak je citatel deliteľné číslom "i" (a % i == 0), číslo "i" sa pridá do zoznamu "aa".

Pre každé číslo "i" od 1 po "b" sa vykoná nasledujúce:

1. a. Ak je menovateľ deliteľné číslom "i" (b % i == 0), číslo "i" sa pridá do zoznamu "bb".
2. Zoznamy "aa" a "bb" sa reverzujú (obrátený poriadok).
3. Vypíše sa obsah zoznamov "aa" a "bb" (delitele citatela a menovateľa).
4. Inicializuje sa premenná "max" na hodnotu 0.

Pre každé číslo "i" v zozname "aa" sa vykoná nasledujúce:

1. a. Ak číslo "i" sa nachádza aj v zozname "bb", nastaví sa premenná "max" na hodnotu "i" a ukončí sa cyklus.
2. Vypíše sa hodnota premennej "max", ktorá predstavuje najväčší spoločný deliteľ.
3. Vypíše sa hodnota výsledku podielu uložená v premennej "vysledok".

Program vyhľadáva najväčší spoločný deliteľ medzi číslami citatel a menovatel, a vypisuje ho spolu s výsledkom podielu týchto čísel.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Softvér a hardvér, Komunikácia a spolupráca, Informačná spoločnosť.

**Maturitné zadanie č. 20.**

***Úloha č.1 - Dvojková sústava***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**cislo=int(input('Dvojkova sustava'))**

**dvojkova=''**

**while cislo != 0:**

**print(cislo//2)**

**zvysok=cislo % 2**

**print(zvysok)**

**cislo=cislo // 2**

**dvojkova += str(zvysok)**

**print(dvojkova[::-1])**

Tento program prevádza zadané číslo z desiatkovej sústavy na dvojkovú sústavu. Tu je komentár k jednotlivým častiam programu:

1. Vstup čísla:
   * Používateľ je vyzvaný, aby zadal číslo v desiatkovej sústave pomocou input('Dvojková sústava').
   * Zadaná hodnota je prevedená na celé číslo a priradená do premennej cislo.
2. Inicializácia premenných:
   * Premenná dvojkova je inicializovaná ako prázdny reťazec. Táto premenná bude slúžiť na ukladanie číslice v dvojkovej sústave.
3. Prevod na dvojkovú sústavu:
   * Cyklus while pokračuje, kým je hodnota premennej cislo nenulová.
   * V každom kroku cyklu sa vykonajú nasledujúce kroky:
     + Vypíše sa celočíselné delenie cislo po dvojke (cislo // 2), čo predstavuje postupné delenie čísla pre konverziu do dvojkovej sústavy.
     + Získaný zvyšok po delení dvoma (cislo % 2) je priradený do premennej zvysok a vypíše sa.
     + Hodnota premennej cislo sa aktualizuje delením dvojkou (cislo = cislo // 2).
     + Zvyšok zvysok sa pridá na koniec reťazca dvojkova pomocou dvojkova += str(zvysok).
   * Tento postup sa opakuje, pričom číslo je postupne delené dvojkou, až kým nedosiahne hodnotu 0.
4. Výpis výsledku:
   * Výsledný reťazec dvojkova obsahuje postupnosť číslic v dvojkovej sústave v opačnom poradí.
   * Preto je použitá operácia dvojkova[::-1], ktorá vracia reťazec v opačnom poradí.
   * Výsledok je vypísaný pomocou print(dvojkova[::-1]).

Celkovo program prechádza postupne cez jednotlivé kroky konverzie z desiatkovej na dvojkovú sústavu a vypisuje výsledný výsledok.

***Úloha č. 2***

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér, Komunikácia a spolupráca, Informačná spoločnosť.

***Pomôcky:***  súbor *sustava.txt* pre úlohu č.1**Maturitné zadanie č. 21**

***Úloha č. 1 – Čiarový kód***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## import tkinter

## 

## z=open('ciarovy\_kod\_1.txt','r')

## 

## canvas=tkinter.Canvas(width=500,height=500)

## canvas.pack()

## 

## for p in z:

## x=50

## y=50

## ciara=0

## xx=x+15

## yy=85

## p=p.strip()

## vstup=p

## for y in vstup:

## y=int(y)

## canvas.create\_text(xx,yy,text=y)

## xx += 7

## for i in vstup:

## i=int(i)

## print(i)

## if i != 0:

## if ciara == 0:

## canvas.create\_rectangle(x,y,x+i,y+80,fill='black')

## elif ciara == 7:

## canvas.create\_rectangle(x+5,y,x+i+5,y+80,fill='black')

## else:

## canvas.create\_rectangle(x,y,x+i,y+70,fill='black')

## x += 10

## ciara += 1

## 

## 

## 

## tkinter.mainloop()

Tento program využíva knižnicu tkinter na vytvorenie grafického okna a vykreslenie čiarového kódu na plátno.

1. Importuje sa modul tkinter.
2. Otvára sa súbor "ciarovy\_kod\_1.txt" na čítanie.
3. Vytvára sa plátno (canvas) s rozmermi 500x500 a umiestňuje sa na obrazovku.

Pre každý riadok (p) v súbore sa vykoná nasledujúce:

a. Inicializuje sa premenná x na hodnotu 50 a y na hodnotu 50. Tieto hodnoty určujú počiatočnú pozíciu vykreslovania čiarového kódu.

b. Inicializujú sa premenné ciara na hodnotu 0, xx na hodnotu x+15 a yy na hodnotu 85.

c. Odstráni sa prípadné medzery zo začiatku a konca riadku (p).

d. Vytvorí sa premenná vstup, do ktorej sa uloží hodnota riadku.

1. e. Pre každý znak y v vstup sa vykoná nasledujúce:
   * Konvertuje sa znak y na celé číslo (int(y)).
   * Na plátno sa vytvorí textový objekt s hodnotou y na pozícii (xx, yy).

Hodnota xx sa zvýši o 7, aby sa posunula o ďalší textový objekt.

* + f. Pre každé číslo i v vstup sa vykoná nasledujúce:
  + Konvertuje sa číslo i na celé číslo (int(i)).
  + Ak i nie je nulové:
    - Ak je ciara rovné 0, vytvorí sa na plátne obdĺžnik od bodu (x, y) po bod (x+i, y+80) so farbou 'black'.
    - Ak je ciara rovné 7, vytvorí sa na plátne obdĺžnik od bodu (x+5, y) po bod (x+i+5, y+80) so farbou 'black'.
    - V opačnom prípade sa vytvorí na plátne obdĺžnik od bodu (x, y) po bod (x+i, y+70) so farbou 'black'.
  + Hodnota x sa zvýši o 10, aby sa posunula na ďalšiu pozíciu vykreslovania.
  + Hodnota ciara sa zvýši o 1.

1. Spúšťa sa nekonečná smyčka tkinter.mainloop(), ktorá zabezpečuje, že okno s plátnom je zobrazené a reaguje na udalosti.

Program načíta číselný reťazec zo súboru a vykreslí čiarový kód na plátno pomocou obdĺžnikov a textových objektov.

## 

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér.

***Pomôcky:*** súbor ciarovy\_kod.txt pre úlohu č.1, súbor stretnutie.txt pre úlohu č.2**Maturitné zadanie č. 22**

***Úloha č.1 – Zobrazenie zátvoriek***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**import tkinter**

**vstup=input('Zadajte priklad')**

**kontrola=0**

**zle=0**

**for znak in vstup:**

**if znak == '(':**

**kontrola += 1**

**elif znak == ')':**

**kontrola -= 1**

**if kontrola < 0:**

**zle += 1**

**if zle == 0:**

**print('Zatvorky su dobre')**

**farby=['blue','red','orange','purple','green']**

**canvas=tkinter.Canvas(width=500,height=100)**

**canvas.pack()**

**farba = -1**

**x=100**

**for znak in vstup:**

**if znak == '(':**

**farba += 1**

**if znak == '(' or znak == ')':**

**canvas.create\_text(x,50,text=znak,fill=farby[farba])**

**else:**

**canvas.create\_text(x,50,text=znak,fill='black')**

**if znak == ')':**

**farba -= 1**

**x += 10**

**else:**

**print('Zatvorky su zle')**

**tkinter.mainloop()**

Tento program kontroluje správnosť zátvoriek v zadanom matematickom výraze a zobrazuje ho v grafickom okne s využitím modulu tkinter. Tu je komentár k jednotlivým častiam programu:

1. Vstup výrazu:
   * Používateľ je vyzvaný, aby zadal matematický výraz pomocou input('Zadajte priklad').
   * Zadaný výraz je priradený do premennej vstup.
2. Kontrola zátvoriek:
   * Premenná kontrola je inicializovaná na hodnotu 0, slúži na kontrolu správnej rovnováhy zátvoriek.
   * Premenná zle je inicializovaná na hodnotu 0, slúži na počítanie nesprávne uzavretých zátvoriek.
   * V cykle for prechádzame cez každý znak vo vstupnom výraze.
   * Ak je znak (, inkrementuje sa kontrola o 1, čo značí otvorenú zátvorku.
   * Ak je znak ), dekrementuje sa kontrola o 1, čo značí uzavretú zátvorku.
   * Ak kontrola klesne pod 0, znamená to, že boli uzavreté zátvorky bez predchádzajúcej otvorenej zátvorky, a inkrementuje sa zle o 1.
3. Výpis výsledku a zobrazenie v grafickom okne:
   * Ak je hodnota premennej zle rovná 0, t.j. neexistujú nesprávne uzavreté zátvorky, vypíše sa správa 'Zátvorky sú dobre'.
   * Inicializuje sa premenná farby s rôznymi farbami pre zobrazenie zátvoriek v grafe.
   * Vytvára sa grafické okno pomocou tkinter.Canvas(width=500, height=100) a je zobrazené pomocou canvas.pack().
   * Premenná farba je inicializovaná na hodnotu -1, slúži na indexovanie farieb pre každú otvorenú zátvorku.
   * Premenná x je inicializovaná na hodnotu 100, slúži na určenie pozície x pre každý znak v grafe.
   * V cykle for prechádzame znak po znaku vo vstupnom výraze.
   * Ak je znak (, inkrementuje sa farba o 1 a vytvorí sa textový objekt s daným znakom a príslušnou farbou.
   * Ak je znak ( alebo ), vytvorí sa textový objekt s daným znakom a čiernou farbou.
   * Ak je znak ), dekrementuje sa farba o 1.
   * Hodnota premennej x sa zvyšuje o 10, aby sa znaky zobrazovali postupne vedľa seba.
4. Výpis výsledku pre nesprávne uzavreté zátvorky:
   * Ak je hodnota premennej zle nenulová, t.j. existujú nesprávne uzavreté zátvorky, vypíše sa správa 'Zátvorky sú zle'.
5. Hlavná slučka pre zobrazenie grafického okna:
   * Slučka tkinter.mainloop() spúšťa hlavný cyklus pre zobrazenie grafického okna a spracovanie udalostí.

Celkovo program kontroluje správnosť zátvoriek v zadanom matematickom výraze a zobrazuje ho v grafickom okne s využitím modulu tkinter.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Reprezentácie a nástroje, Softvér a hardvér, Komunikácia a spolupráca.

**Maturitné zadanie č. 23**

***Úloha č. 1- Teploty***

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**rano=[]**

**obed=[]**

**vecer=[]**

**dni=int(input('Zadajte pocet dni: '))**

**for i in range(dni):**

**priemernat=0**

**teplotar=int(input('Zadajte teplotu rano: '))**

**teplotao=int(input('Zadajte teplotu poobede: '))**

**teplotav=int(input('Zadajte teplotu vecer: '))**

**priemernat=(teplotar+teplotao+teplotav)/3**

**rano.append(teplotar)**

**obed.append(teplotao)**

**vecer.append(teplotav)**

**print(f'Priemer za {i+1} den je {priemernat}')**

**celkover=0**

**for i in rano:**

**celkover +=i**

**print(f'Priemer za rano bol {celkover/dni}')**

**celkoveo=0**

**for i in obed:**

**celkoveo +=i**

**print(f'Priemer za poobede bol {celkoveo/dni}')**

**celkovev=0**

**for i in vecer:**

**celkovev +=i**

**print(f'Priemer za vecer bol {celkovev/dni}')**

Tento program zaznamenáva teploty ráno, poobede a večer počas určitého počtu dní a vypočíta priemerné teploty pre jednotlivé časti dňa a celý deň.

1. Inicializujú sa prázdne zoznamy rano, obed a vecer, do ktorých budú uložené teploty.
2. Počet dní sa načíta zo vstupu a uloží sa do premennej dni.

Pre každý deň v rozsahu dni sa vykoná nasledujúce:

a. Inicializuje sa premenná priemernat na hodnotu 0.

b. Zo vstupu sa načítajú teploty ráno, poobede a večer a uložia sa do premenných teplotar, teplotao a teplotav.

c. Vypočíta sa priemer teplôt pre daný deň (priemernat = (teplotar + teplotao + teplotav) / 3).

d. Teploty sa pridajú do príslušných zoznamov rano, obed a vecer.

1. e. Vypíše sa priemer pre daný deň.

Vypočíta sa celkový priemer teplôt pre časový úsek ráno:

a. Inicializuje sa premenná celkover na hodnotu 0.

b. Pre každú teplotu i v zozname rano sa pripočíta k celkover.

1. c. Vypíše sa priemer pre ráno (celkover / dni).

Vypočíta sa celkový priemer teplôt pre časový úsek poobede:

a. Inicializuje sa premenná celkoveo na hodnotu 0.

b. Pre každú teplotu i v zozname obed sa pripočíta k celkoveo.

1. c. Vypíše sa priemer pre poobede (celkoveo / dni).

Vypočíta sa celkový priemer teplôt pre časový úsek večer:

a. Inicializuje sa premenná celkovev na hodnotu 0.

b. Pre každú teplotu i v zozname vecer sa pripočíta k celkovev.

1. c. Vypíše sa priemer pre večer (celkovev / dni).

Program umožňuje zaznamenávať teploty a vypočítať priemery pre jednotlivé časti dňa a celý deň na základe zadaného počtu dní.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér, Komunikácia a spolupráca.

***Pomôcky:*** súbor *termin.xlsx* pre úlohu č.2

## Maturitné zadanie č. 24

## *Úloha č. 1- Ako rastie strom*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## import random

## 

## vysoky=0

## vyska=[]

## kontrola=0

## 

## for i in range(20):

## a=random.randrange(5,20)

## vysoky += a

## vyska.append(vysoky)

## print(vyska)

## 

## kedy=int(input('Zadajte vysku: '))

## 

## for i in vyska:

## if i==kedy:

## print(f'Strom dosiahol danu vysku v roku {vyska.index(i)+1}')

## break

## elif i > kedy and i < max(vyska):

## print(f'Strom dosiahol danu vysku v rokocho {vyska.index(i)} a {vyska.index(i)+1} ')

## break

## 

## while kedy > max(vyska):

## a=random.randrange(5,20)

## vysoky += a

## vyska.append(vysoky)

## 

## print(vyska)

## kedyd=kedy

## 

## for i in vyska:

## if i == kedyd:

## print(f'Strom dosiahol danu vysku v roku {vyska.index(i)+1}')

## break

## elif i > kedyd:

## print(f'Strom dosiahol danu vysku v rokocho {vyska.index(i)} a {vyska.index(i)+1} ')

## break

Tento program simuluje rast stromu a kontroluje, kedy dosiahol zadanú výšku. Tu je komentár k jednotlivým častiam programu:

1. Generovanie rastu stromu:
   * Premenná vysoky je inicializovaná na hodnotu 0, slúži na sledovanie celkového rastu stromu.
   * Vytvára sa prázdny zoznam vyska, do ktorého sa postupne pridávajú hodnoty rastu stromu.
   * V cykle for sa iteruje 20-krát:
     + Generuje sa náhodné číslo a medzi 5 a 20, ktoré reprezentuje rast stromu v danom roku.
     + Hodnota a sa pripočíta k premennej vysoky, čím sa aktualizuje celkový rast stromu.
     + Aktuálna hodnota vysoky sa pridáva do zoznamu vyska.
   * Vypisuje sa zoznam vyska, ktorý obsahuje postupné hodnoty rastu stromu.
2. Vstup a kontrola výšky:
   * Používateľ je vyzvaný, aby zadal želanú výšku stromu pomocou int(input('Zadajte vysku: ')).
   * V cykle for prechádzame hodnotami v zozname vyska.
   * Ak je hodnota i rovná zadanému kedy, vypíše sa správa o roku, kedy strom dosiahol danú výšku a cyklus sa ukončí pomocou break.
   * Ak je hodnota i väčšia ako kedy a menšia ako maximálna hodnota v vyska, vypíše sa správa o dvoch rokoch, medzi ktorými strom dosiahol danú výšku, a cyklus sa ukončí pomocou break.
3. Rozšírenie rastu stromu:
   * V cykle while sa opakuje, kým zadaná výška kedy je väčšia ako maximálna hodnota v vyska.
   * Opakuje sa rovnaký proces generovania rastu stromu ako v prvom kroku.
   * Nové hodnoty rastu sa pridávajú do zoznamu vyska.
4. Kontrola výšky po rozšírení:
   * Premenná kedyd je inicializovaná na hodnotu kedy, slúži na zachovanie pôvodnej hodnoty.
   * V cykle for prechádzame hodnotami v zozname vyska.
   * Ak je hodnota i rovná zadanému kedyd, vypíše sa správa o roku, kedy strom dosiahol danú výšku, a cyklus sa ukončí pomocou break.
   * Ak je hodnota i väčšia ako kedyd, vypíše sa správa o dvoch rokoch, medzi ktorými strom dosiahol danú výšku, a cyklus sa ukončí pomocou break.

Celkovo program generuje rast stromu, umožňuje zadať výšku a kontroluje, v ktorých rokoch strom dosiahol túto výšku.

## 

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Softvér a hardvér, Komunikácia a spolupráca, Informačná spoločnosť.

**Maturitné zadanie č. 25**

## *Úloha č.1 – Koncoročný výlet*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**z=open('prihlasenie.txt','r')**

**riadky = 0**

**osoby=0**

**vek=[]**

**for riadok in z:**

**if riadky % 2 == 1:**

**vek.append(int(riadok))**

**osoby += 1**

**riadky += 1**

**print(f'Pocet osob je {osoby}')**

**print(vek)**

**menej=0**

**cena=int(input('Kolko je cena: '))**

**celkovacena=0**

**for cislo in vek:**

**if cislo < 15:**

**menej += 1**

**celkovacena += cena/2**

**else:**

**celkovacena += cena**

**print(menej)**

**print(celkovacena)**

Tento program načíta údaje zo súboru "prihlasenie.txt" a vykoná niekoľko výpočtov na základe týchto údajov.

Prvá časť programu otvára súbor "prihlasenie.txt" pre čítanie a inicializuje premenné. Premenná "riadky" je počítadlo riadkov v súbore, premenná "osoby" je počítadlo záznamov o osobách a premenná "vek" je prázdne pole, do ktorého budú uložené veky.

Ďalšia časť programu prechádza cez každý riadok v súbore. Ak je aktuálny riadok nepárne číslo riadku (keďže indexovanie začína od 0), vykoná sa nasledujúci kód: vek sa konvertuje na celé číslo a pridá sa do poľa "vek", zvýši sa počítadlo osôb o 1 a zvýši sa počítadlo riadkov o 1.

Po dokončení tohto cyklu program vypíše počet osôb a pole "vek".

Následne program pokračuje v časti, kde sa používateľovi zobrazí otázka na cenu a táto hodnota sa uloží do premennej "cena". Následne sa inicializujú premenné "menej" a "celkovacena". Premenná "menej" bude slúžiť na počítanie počtu osôb s vekom nižším ako 15, a premenná "celkovacena" bude slúžiť na výpočet celkovej ceny.

Ďalší cyklus prechádza každým číslom v poli "vek". Ak je dané číslo menšie ako 15, zvýši sa počítadlo "menej" o 1 a k hodnote "celkovacena" sa pridá polovica ceny. V opačnom prípade sa k hodnote "celkovacena" pridá plná cena.

Nakoniec program vypíše počet osôb s vekom nižším ako 15 a celkovú cenu.

Celkovo vzaté, tento program načíta údaje o veku z textového súboru, vypočíta počet osôb, počet osôb s vekom nižším ako 15 a celkovú cenu na základe zadaného veku a ceny.

# *Úloha č.2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér, Komunikácia a spolupráca.

***Pomôcky:*** súbor *prihlaseni.txt* pre úlohu č.2

**Maturitné zadanie č. 26**

## *Úloha č. 1 –Slovník*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## z=open('slovnik.txt','r')

## slovenske=[]

## anglicke=[]

## riadok=0

## for i in z:

## print(i,end='')

## i=i.strip()

## if riadok % 2 == 0:

## slovenske.append(i)

## else:

## anglicke.append(i)

## riadok += 1

## 

## print(slovenske)

## print(anglicke)

## 

## z.close

## 

## subor=open('slovnik.txt','a')

## slovo=input('Zadajte slovo ktore chcete vediet: ')

## while subor !='':

## if slovo in slovenske:

## ktore=slovenske.index(slovo)

## kde=anglicke[ktore]

## print(kde)

## break

## elif slovo in anglicke:

## ktore=anglicke.index(slovo)

## kde=slovenske[ktore]

## print(kde)

## break

## else:

## subor.write('\n')

## subor.write(slovo)

## preklad=input('Aky je preklad tohoto slova: ')

## subor.write('\n')

## subor.write(preklad)

## break

Tento program pracuje so slovníkom uloženým v súbore a umožňuje vyhľadávanie a pridávanie slov. Tu je komentár k jednotlivým častiam programu:

1. Načítanie slovníka zo súboru:
   * Súbor slovnik.txt sa otvára v režime čítania pomocou z=open('slovnik.txt','r').
   * Vytvárajú sa prázdne zoznamy slovenske a anglicke, do ktorých sa budú ukladať slovenské a anglické slová zo súboru.
   * Premenná riadok slúži na sledovanie aktuálneho riadku v súbore.
   * V cykle for prechádzame riadky súboru:
     + Vypisuje sa aktuálny riadok pomocou print(i,end='').
     + Odstraňuje sa biely priestor zo začiatku a konca riadku pomocou i=i.strip().
     + Ak je riadok deliteľný 2 bez zvyšku, pridáva sa slovenské slovo do zoznamu slovenske, inak sa pridáva anglické slovo do zoznamu anglicke.
     + Inkrementuje sa premenná riadok o 1.
   * Vypisujú sa zoznamy slovenske a anglicke, ktoré obsahujú načítané slová zo súboru.
2. Vyhľadávanie slova v slovníku:
   * Súbor slovnik.txt sa zatvára pomocou z.close.
   * Súbor slovnik.txt sa otvára v režime pridávania na koniec pomocou subor=open('slovnik.txt','a').
   * Používateľ je vyzvaný, aby zadal slovo, ktoré chce vyhľadať, pomocou input('Zadajte slovo ktore chcete vediet: ') a toto slovo sa ukladá do premennej slovo.
   * V cykle while sa opakuje, kým súbor nie je prázdny (subor != ''):
     + Ak je slovo obsiahnuté v zozname slovenske, získa sa jeho index pomocou ktore=slovenske.index(slovo).
     + Z anglického zoznamu anglicke sa získa preklad slova na rovnakej pozícii ako slovo pomocou kde=anglicke[ktore].
     + Vypisuje sa preklad pomocou print(kde) a cyklus sa ukončuje pomocou break.
     + Ak slovo nie je obsiahnuté v slovenske, skúsi sa vyhľadať v zozname anglicke a získať príslslušný preklad pomocou ktore=anglicke.index(slovo) a kde=slovenske[ktore].
     + Vypisuje sa preklad pomocou print(kde) a cyklus sa ukončuje pomocou break.
     + Ak slovo nie je obsiahnuté ani v slovenske ani v anglicke, pridáva sa do súboru nový riadok so slovom pomocou subor.write('\n') a subor.write(slovo).
     + Používateľ je vyzvaný, aby zadal preklad tohto slova pomocou preklad=input('Aky je preklad tohoto slova: ') a tento preklad sa tiež pridáva do súboru.
     + Cyklus sa ukončuje pomocou break, čím sa zabezpečí, že sa do súboru pridá iba jedno nové slovo.

Celkovo program načítava slovník zo súboru, umožňuje vyhľadávanie slov a pridávanie nových slov s ich prekladmi do slovníka.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér.

# *Pomôcky:* súbor *slovnik.txt* pre úlohu č.1Maturitné zadanie č. 27

# *Úloha č. 1 – Spokojnosť zákazníkov*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## z=open('spokojnost\_1.txt','r')

## vyjadrenia=0

## hodinyspokojny=[0]\*24

## hodinynespkojny=[0]\*24

## spolu=[0]\*24

## for i in z:

## hodina=i[0:2]

## hodina=int(hodina)

## spokojnost=i[6:9]

## if spokojnost == 'áno':

## hodinyspokojny[hodina] += 1

## spolu[hodina] += 1

## elif spokojnost == 'nie':

## hodinynespkojny[hodina] += 1

## spolu[hodina] += 1

## vyjadrenia += 1

## print(f'Celkovy pocet vyjadreni je {vyjadrenia}')

## print(f'Najviac spokojny zakaznici boli v {max(hodinyspokojny)} hodine')

## print(f'Najviac nespokojny zakaznici boli v {max(hodinynespkojny)} hodine')

## print(hodinyspokojny)

## print(spolu)

## for i in spolu:

## if i == 0:

## continue

## else:

## kde=(spolu.index(i))

## dobry=hodinyspokojny[kde]

## percento=(dobry/spolu[kde]) \* 100

## print(f'V hodine {kde} bolo spokojnych {percento} % zakaznikov')

Tento program načíta údaje zo súboru "spokojnost\_1.txt" a vykoná analýzu spokojnosti zákazníkov v závislosti od času.

Po otvorení súboru pre čítanie sa inicializujú premenné. Premenná "vyjadrenia" bude slúžiť na počítanie celkového počtu vyjadrení, premenné "hodinyspokojny", "hodinynespokojny" a "spolu" sú zoznamy o dĺžke 24, ktoré budú slúžiť na počítanie počtu spokojných, nespokojných a celkových vyjadrení v jednotlivých hodinách.

Nasleduje cyklus, ktorý prechádza každý riadok v súbore. Z každého riadku sa extrahuje hodnota hodiny a spokojnosť. Hodina sa konvertuje na celé číslo. Ak je spokojnosť označená ako "áno", príslušnému indexu v zozname "hodinyspokojny" a "spolu" sa pridá 1. Ak je spokojnosť označená ako "nie", príslušnému indexu v zozname "hodinynespokojny" a "spolu" sa pridá 1. Zároveň sa zvýši počítadlo "vyjadrenia".

Po dokončení cyklu program vypíše celkový počet vyjadrení, hodinu s najväčším počtom spokojných zákazníkov, hodinu s najväčším počtom nespokojných zákazníkov, zoznam "hodinyspokojny" a zoznam "spolu".

Následne nasleduje ďalší cyklus, ktorý prechádza každý prvok v zozname "spolu". Ak je hodnota prvku rovná 0, program pokračuje ďalej. Ak je hodnota nenulová, program vypočíta percento spokojných zákazníkov pre danú hodinu a vypíše ho spolu s číslom hodiny.

Celkovo vzaté, tento program analyzuje spokojnosť zákazníkov v závislosti od času a poskytuje rôzne štatistiky a výstupy, vrátane celkového počtu vyjadrení, hodín s najväčším počtom spokojných a nespokojných zákazníkov, percenta spokojných zákazníkov pre jednotlivé hodiny a zoznamov počtu spokojných zákazníkov a celkového počtu vyjadrení pre jednotlivé hodiny.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Softvér a hardvér, Informačná spoločnosť.

***Pomôcky:*** súbor *spokojnost.txt* pre úlohu č.1**Maturitné zadanie č. 28**

# *Úloha č. 1 – Puknutý tanier*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

## import tkinter

## import random

## canvas=tkinter.Canvas(height=500,width=500)

## canvas.pack()

## taniera=['A','B','C','D','E','F','G','H','I','J']

## puknuty=random.choice(taniera)

## x=75

## y=150

## zle=[]

## for i in taniera:

## kolko=0

## canvas.create\_oval(x,y,x+30,y+30,fill='blue')

## canvas.create\_text(x+15,y+15,text=i)

## if i == puknuty:

## puknutyx=x

## puknutyy=y

## print(puknutyx,puknutyy,puknuty)

## else:

## nepuknutyx=x

## nepuknutyy=y

## x += 35

## def klik(event):

## kolko=0

## xx= event.x

## yy= event.y

## if xx >= puknutyx and xx <= puknutyx+30 and yy >= puknutyy and yy <= puknutyy+30:

## canvas.delete('all')

## canvas.create\_text(250,250,font='Arial 50',text='Vyhrali ste')

## canvas.create\_text(400,250,font='Arial 10',text=zle)

## else:

## if xx >= nepuknutyx and xx <= nepuknutyx+30 and yy >= nepuknutyy and yy <= nepuknutyy+30:

## kolko += 1

## if kolko > 2:

## zle.append(i)

## canvas.bind('<ButtonPress>', klik)

## tkinter.mainloop()

Tento program je jednoduchá hra implementovaná pomocou knižnice Tkinter v jazyku Python. Tu je komentár k jednotlivým častiam programu:

1. Import knižnice tkinter:
   * Importuje sa knižnica tkinter pre vytvorenie grafického rozhrania.
2. Vytvorenie plátna:
   * Vytvára sa plátno s výškou 500 pixelov a šírkou 500 pixelov pomocou canvas=tkinter.Canvas(height=500,width=500).
   * Plátno sa umiestňuje pomocou canvas.pack().
3. Definícia tanierov:
   * Vytvára sa zoznam taniera, ktorý obsahuje jednotlivé označenia tanierov.
   * Náhodne sa vyberie jedno označenie taniera pomocou puknuty=random.choice(taniera).
4. Umiestnenie tanierov na plátno:
   * Inicializujú sa premenné x a y, ktoré slúžia na určenie pozície taniera na plátno.
   * Pre každý tanier v zozname taniera:
     + Vykreslí sa ovál reprezentujúci tanier na pozícii (x, y) pomocou canvas.create\_oval(x,y,x+30,y+30,fill='blue').
     + Vykreslí sa text s označením taniera na stredovú pozíciu taniera pomocou canvas.create\_text(x+15,y+15,text=i).
     + Ak je aktuálny tanier rovnaký ako puknuty, zapamätá sa jeho pozícia puknutyx a puknutyy a vypíše sa označenie taniera puknuty.
     + V opačnom prípade sa zapamätá pozícia taniera nepuknutyx a nepuknutyy.
     + Inkrementuje sa premenná x o 35, aby sa tanier posunul na nasledujúcu pozíciu.
5. Funkcia pre spracovanie kliknutia:
   * Definuje sa funkcia klik, ktorá bude spracovávať udalosť kliknutia myšou.
   * V rámci tejto funkcie sa spracováva pozícia kliknutia pomocou premenných xx a yy.
   * Ak pozícia kliknutia sa nachádza v oblasti taniera označeného ako puknuty:
     + Vymaže sa všetko na plátne pomocou canvas.delete('all').
     + Vykreslí sa text "Vyhrali ste" na stred plátna pomocou canvas.create\_text(250,250,font='Arial 50',text='Vyhrali ste').
     + Vykreslí sa text so zoznamom zlých označení tanierov na danej pozícii pomocou canvas.create\_text(400,250,font='Arial 10',text=zle).
   * V opačnom prípade, ak pozícia kliknutia sa nachádza v oblasti taniera označeného ako nepuknuty:
     + Inkrementuje sa premenná kolko.
     + Ak kolko je väčšie ako 2, pridá sa označenie taniera i do zoznamu zle.
6. Pripojenie funkcie k udalosti kliknutia:
   * Plátno sa pripája na udalosť stlačenia tlačidla myši pomocou canvas.bind('<ButtonPress>', klik).
7. Spustenie hlavného cyklu udalostí:
   * Spustí sa hlavný cyklus udalostí pomocou tkinter.mainloop(), ktorý čaká na vstup od používateľa a spracováva udalosti.

Tento program vytvára jednoduchú interaktívnu hru, kde používateľ má za úlohu kliknúť na správny tanier a vyhrať. Ak klikne na správny tanier, zobrazí sa hlásenie o výhre. Ak klikne na nesprávny tanier viac ako 2-krát, jeho označenie sa pridá do zoznamu zlých označení tanierov.

## *Úloha č. 2*

Prierezová úloha, ktorá zasahuje do tematických okruhov: Reprezentácie a nástroje, Softvér a hardvér, Informačná spoločnosť.

**Maturitné zadanie č. 29**

## *Úloha č. 1 – Skoky*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**z=open('skokyvyska.txt','r')**

**skoky=[]**

**for i in z:**

**skoky.append(int(i))**

**print(skoky)**

**max=0**

**for i in skoky:**

**if i > max:**

**max = i**

**kolko=0**

**for i in skoky:**

**if max == i:**

**kolko += 1**

**print(f'Najdlhsi skok je {max} a bol skoceny {kolko} krat')**

Tento program načíta údaje zo súboru "skokyvyska.txt", ktorý obsahuje hodnoty výšok skokov, a vykoná analýzu týchto hodnôt.

Po otvorení súboru pre čítanie sa inicializuje prázdne pole s názvom "skoky".

Nasleduje cyklus, ktorý prechádza každý riadok v súbore. Každý riadok sa konvertuje na celé číslo a pridá sa do poľa "skoky" pomocou metódy append().

Po dokončení tohto cyklu program vypíše pole "skoky", ktoré obsahuje všetky načítané hodnoty.

Ďalšia časť programu hľadá najväčšiu hodnotu v poli "skoky". Premenná "max" je inicializovaná na hodnotu 0. Nasleduje cyklus, ktorý prechádza každú hodnotu v poli "skoky". Ak je aktuálna hodnota väčšia ako hodnota "max", nastaví sa "max" na túto hodnotu. Týmto spôsobom sa nájde najväčšia hodnota v poli.

Ďalší cyklus prechádza znova každú hodnotu v poli "skoky" a počíta, koľkokrát sa táto hodnota zhoduje s maximálnou hodnotou "max". Každýkrát, keď sa nájde zhoda, inkrementuje sa premenná "kolko".

Na záver program vypíše najväčšiu hodnotu "max" a koľkokrát sa táto hodnota vyskytla v poli "skoky".

Celkovo vzaté, tento program načíta hodnoty výšok skokov zo súboru, nájde najväčšiu hodnotu a spočíta, koľkokrát sa táto hodnota vyskytuje. Následne vypíše najdlhší skok a koľkokrát bol tento skok uskutočnený.

**b)**

**z=open('skoky.txt','r')**

**skoko=0**

**for i in z:**

**skoky=i.find(' ')**

**skokan=(i[:skoky])**

**for x in i:**

**if x == ' ':**

**skoko += 1**

**print(skokan,' ',skoko)**

**skoko = 0**

Tento program načíta údaje zo súboru "skoky.txt" a vykoná analýzu týchto údajov.

Po otvorení súboru pre čítanie sa inicializuje premenná "skoko" na hodnotu 0.

Nasleduje cyklus, ktorý prechádza každý riadok v súbore. V každom riadku sa pomocou metódy find() hľadá prvý výskyt medzery. Výsledok tejto operácie sa priradí do premennej "skoky". Následne sa zo začiatku riadku extrahuje časť reprezentujúca meno skokana a priradí sa do premennej "skokan".

Ďalší vnorený cyklus prechádza každý znak v riadku. Ak je aktuálny znak medzera, inkrementuje sa premenná "skoko". Týmto spôsobom sa počíta počet medzier (a teda aj počet skokov) v riadku.

Program potom vypíše meno skokana a príslušný počet skokov. Premenná "skoko" sa potom nastaví na hodnotu 0, aby bola pripravená pre ďalší riadok.

Celý proces sa opakuje pre každý riadok v súbore, a teda sa vypíšu mená skokanov a príslušné počty skokov pre každý riadok.

Celkovo vzaté, tento program načíta údaje zo súboru, kde každý riadok obsahuje meno skokana a jeho skoky oddelené medzerou. Program spočíta počet skokov pre každého skokana a vypíše ich spolu s menom skokana.

***Úloha č. 2***

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Softvér a hardvér.

# *Pomôcky:* súbor *skoky.txt* pre úlohu č.1, súbor *chyba\_v\_datach.xlsx* pre úlohu č.2 Maturitné zadanie č. 30

## *Úloha č.1 – Rodinné financie*

Vytvorenie programu v programovacom jazyku Python, žiak rieši zadaný algoritmický problém, tematický okruh Algoritmické riešenie problémov.

**z=open('uroky.txt','r')**

**percenta=[]**

**for i in z:**

**po=i.find('%')**

**percenta.append(int(i[:po]))**

**max=0**

**for i in percenta:**

**if i > max:**

**max = i**

**print(max)**

**vklad=int(input('Zadajte vklad'))**

**for i in range(10):**

**vklad= ((vklad/100)\*max)+vklad**

**print(vklad)**

Tento program slúži na výpočet výnosu zo vkladu na základe úrokových sadzieb uložených v súbore "uroky.txt". Tu je komentár k jednotlivým častiam programu:

1. Otvorenie súboru:
   * Súbor "uroky.txt" sa otvára pomocou z=open('uroky.txt','r') pre čítanie úrokových sadzieb.
2. Získanie percentuálnych hodnôt:
   * Vytvára sa prázdny zoznam percenta, do ktorého budú uložené percentuálne hodnoty zo súboru.
   * Pre každý riadok v súbore z:
     + Hľadá sa pozícia znaku '%' v riadku pomocou po=i.find('%').
     + Úroková sadzba (hodnota pred znakom '%') sa konvertuje na celé číslo a pridá sa do zoznamu percenta pomocou percenta.append(int(i[:po])).
3. Výpočet maximálnej úrokovej sadzby:
   * Inicializuje sa premenná max na hodnotu 0, ktorá bude slúžiť na uchovanie maximálnej úrokovej sadzby.
   * Pre každú úrokovú sadzbu v zozname percenta:
     + Ak je úroková sadzba väčšia ako aktuálna maximálna sadzba (max), nastaví sa nová maximálna sadzba (max = i).
4. Výpis maximálnej úrokovej sadzby:
   * Vypíše sa maximálna úroková sadzba pomocou print(max).
5. Vklad a výpočet výnosu:
   * Používateľ je požiadaný o zadanie vkladu pomocou vklad=int(input('Zadajte vklad')).
   * Pre 10 iterácií:
     + Výška vkladu je aktualizovaná podľa vzorca pre výpočet výnosu: vklad = ((vklad/100) \* max) + vklad.
6. Výpis konečnej výšky vkladu:
   * Vypíše sa konečná výška vkladu po výpočte výnosu pomocou print(vklad).

Tento program načíta úrokové sadzby zo súboru, nájde maximálnu sadzbu, a potom na základe tejto sadzby vypočíta výnos zo vkladu.

***Úloha č. 2***

Prierezová úloha, ktorá zasahuje do tematických okruhov: Algoritmické riešenie problémov, Reprezentácie a nástroje, Komunikácia a spolupráca.

***Pomôcky :*** súbor *urok.txt* pre úlohu č.1, plan\_mesta.xlsx pre úlohu č.2