

# 14. praktiskais darbs. 2. semestris

## 1. uzdevums

Sekmju žurnāla izveide. Vienas skolas visu skolēnu sekmju uzskaite glabājas vārdnīca, kurā katrs skolēns tiek identificēts pēc tā personas koda (11 ciparu virkne).

Par katru skolēnu tiek uzkrāta šāda informācija:

1. Vārds

2. Uzvārds

3. Klase

4. Mācību priekšmeti un tajos saņemtās atzīmes (mācību priekšmetu un tajos saņemto atzīmju skaitu katrai klasei nosaka programmas autors)

Pēc skolas izveides un atzīmju ievades jāparedz, ka tiek izveidots "baltais žurnāls", kur redzamas katra skolēna gada atzīmes katrā mācību priekšmetā, kā arī "liecības" izveidi, ierakstot informāciju teksta datnē ar nosaukumu VardsUzvards.txt

Testējot programmu, jāparedz, kā skolā ir vismaz 10 skolēni, kas mācās vismaz 3 dažādās klasēs un katrā klasē ir vismaz 3 mācību priekšmeti, kuros izliek vismaz 3 atzīmes. (5 punkti).

### Kods:

# Programmas nosaukums: Skolas sekmju uzskaite

# 1. uzdevums (1MPR14\_Vladislavs\_Babaņins)

# Uzdevuma formulējums: Sekmju žurnāla izveide. Vienas skolas visu skolēnu sekmju uzskaite glabājas vārdnīca, kurā katrs skolēns tiek identificēts pēc tā personas koda (11 ciparu virkne).

# Par katru skolēnu tiek uzkrāta šāda informācija:

# 1. Vārds

# 2. Uzvārds

# 3. Klase

# 4. Mācību priekšmeti un tajos saņemtās atzīmes (mācību priekšmetu un tajos saņemto atzīmju skaitu katrai klasei nosaka programmas autors)

# Pēc skolas izveides un atzīmju ievades jāparedz, ka tiek izveidots "baltais žurnāls", kur redzamas katra skolēna gada atzīmes katrā mācību priekšmetā, kā arī

# "liecības" izveidi, ierakstot informāciju teksta datnē ar nosaukumu VardsUzvards.txt

# Testējot programmu, jāparedz, kā skolā ir vismaz 10 skolēni, kas mācās vismaz 3 dažādās klasēs un katrā klasē ir vismaz 3 mācību priekšmeti, kuros izliek vismaz 3 atzīmes. (5 punkti).

# Programmas autors: Vladislavs Babaņins

# Versija 1.0

""""

ĒRTĪBAS PĒC PROGRAMMA TIKA IZVEIDOTI DIVI FAILI

.CSV FAILS, KURU AIZPILD PATS LIETOTĀJS AR ROKU

.TXT FAILS, KUR TIEK VEIDOTAS LIECĪBAS UN KURU IZVEIDO PROGRAMMA

TIKA REALIZĒTS TIEŠI TĀ, JO TĀ IR ĒRTĀK LIETOTĀJAM, NEVISS IEVĀDĪT PROGRAMMA ĻOTI DAUDZ SKOLĒNU VĀRDU, UZVĀRDU, PERSONAS ID UN ATZĪMES.

TĀPĒC PROGRAMMA NAV PAREDZĒTA FUNCKIJA KAS PAPRASA LIETOTĀJAM IEVĀDĪT SĒKMES VAI SKOLĒNUS.

""""

import csv

def read\_student\_data(file\_name):

# Šī funkcija nolasa skolēna datus no CSV faila un sakārto tos vārdnīcā.

# file\_name - ceļš līdz failām (.csv datne).

# Piemēram:

# file\_name = "C:\\Users\\User\\Desktop\\student\_data.csv"

# Tas atgriež skolēna datu vārdnīcu.

# return student\_data

student\_data = {} # Definējam tukšu vārdnīcu

with open(file\_name, 'r', newline='', encoding='utf-8-sig') as file:

# Datu izvilkšana no CSV rindām un kārtošana vārdnīcā

# Katrs skolēns tiek identificēts ar unikālu skolēna kodu

```

reader = csv.DictReader(file, delimiter=';')

for row in reader:

    # Izvelcam attiecīgos datus no katras CSV faila rindas un sakārtojam tos vārdnīcā

    student_code = row['Personas kods'] # Izvelcam skolēna unikālo ID identifikatoru
    (Personas kodu) no kolonnas "Personas kods"

    student_name = row['Vārds'] # Izvelcam skolēna uzvārdu no rindas "Vārds".

    student_last_name = row['Uzvārds'] # Izvelcam skolēna uzvārdu no rindas "Uzvārds".

    student_class = row['Klase'] # Izvelcam skolēna klasi no rindas "Klase".

    math_grades = [int(grade) for grade in row['Matemātika'].split()] # Izņemam
    matemātikas atzīmes no rindas "Matemātika" un pārveidojam "int" skaitļos

    science_grades = [int(grade) for grade in row['Dabaszinības'].split()] # Izņemam
    dabaszinības atzīmes no rindas "Dabaszinības" un pārveidojam "int" skaitļos

    history_grades = [int(grade) for grade in row['Vēsture'].split()] # Izņemam vēstures
    atzīmes no rindas "Vēsture" un pārveidojam "int" skaitļos

    geo_grades = [int(grade) for grade in row['Ģeogrāfija'].split()] # Izņemam ģeogrāfijas
    atzīmes no rindas "Ģeogrāfija" un pārveidojam "int" skaitļos

    computer_grades = [int(grade) for grade in row['Datorika'].split()] # Izņemam
    datorika atzīmes no rindas "Datorika" un pārveidojam "int" skaitļos

    # Izveido vārdnīcu ar nosaukumu subject_grades, lai saglabātu katra priekšmeta
    atzīmes. Vārdnīca ir strukturēta šādi:

    # Priekšmets: Atzīmē priekšmeta nosaukumu ('Matemātika', 'Dabaszinības', 'Vēsture',
    'Ģeogrāfija', 'Datorika').

    # Atzīmes: atspoguļo attiecīgajam mācību priekšmetam atbilstošo atzīmju sarakstu.

    subject_grades = {

        'Matemātika': math_grades,

        'Dabaszinības': science_grades,

        'Vēsture': history_grades,

        'Ģeogrāfija': geo_grades,

        'Datorika': computer_grades

    }

```

```

        # Pievienojam skolēna informāciju un atzīmes vārdnīcai student_data

        student_data[student_code] = {'Vārds': student_name, 'Uzvārds': student_last_name,
        'Klase': student_class, 'Marks': subject_grades}

    return student_data

def print_student_grades_by_code(student_data, student_code):
    # Šī funkcija izdrukā skolēna atzīmes, kas identificētas pēc skolēna koda.
    # student_data: vārdnīca, kurā ir skolēna dati.
    # studenta_kods: skolēna unikālais identifikators (Skolēna ID (Personas kods)).
    # Atgriež: None

    if student_code in student_data: # Pārbaudam, vai skolēna datu vārdnīcā ir šāds skolēna
    kods (vai tāds skolēns eksistē)

        student_info = student_data[student_code] # Iegūvam informāciju par doto skolēna
        kodu

        student_name = student_info['Vārds'] # Iegūvam informāciju par doto skolēna vārdu

        student_surname = student_info['Uzvārds'] # Iegūvam informāciju par doto skolēna
        uzvārdu

        student_grades = student_info['Marks'] # Iegūvam informāciju par doto skolēna
        atzīmēm

        print(f"Atzīmes skolēnam: {student_name} {student_surname}") # Uzrakstām
        lietotājam skolēna atzīmes.

        for subject, grades in student_grades.items(): # Ejām ciklā caur katru priekšmetu un
        tām atbilstošām atzīmēm un izdrukām tos

            print(f"Priekšmets: {subject}, Atzīmes: {grades}") # Izdrukājam priekšmeta
            nosaukumu un tā atzīmes

    else: # Izdrukājam paziņojumu, kas norāda, ka skolēns ar norādīto kodu netika atrasts, ja
    netika atrasts šāds personas kods.

        print(f"Skolēns ar personas kodu '{student_code}' nav atrasts.")

```

```

def calculate_average(grades):
    # Aprēķina vidējo atzīmi no atzīmju saraksta.
    # Tas atgriež vidējo atzīmi kā noapaļotu līdz veselam skaitlim.
    # (Ja nevajag apoļot, tad return average)
    # Ja atzīmju saraksts ir tukšs, tas atgriež 0.

    total = sum(grades)
    count = len(grades)
    if count == 0:
        return 0 # Gadījums, kad nav atzīmēs.
    average = total / count
    return round(average)

def generate_white_journal(student_data):
    # Ģenerē baltu žurnālu, pamatojoties uz skolēna datiem.
    # Tas atgriež vārdnīcu, kurā ir priekšmeti kā atslēgas, un skolēnu vārdu un vidējo atzīmju
    vārdnīcu kā vērtības.
    # Atgriež baltu žurnālu.

    white_journal = {} # Inicializējam tukšu vārdnīcu, lai saglabātu balto žurnālu.

    for student_name, data in student_data.items(): # Iterējam caur katru skolēnu
        student_datu vārdnīcā.
        student_class = data['Klase'] # Iegūvam skolēna klasi
        subject_grades = data['Marks'] # Iegūvam skolēna atzīmes

        for subject, grades in subject_grades.items(): # Iterējam caur katru priekšmetu un tā
            atbilstošās atzīmem.

```

```
    average_grade = calculate_average(grades) # Aprēķinam katra priekšmeta vidējo  
    atzīmi un saglabājam to baltajā žurnālā.
```

```
    if subject not in white_journal: # Ja priekšmets vēl nav iekļauts vārdnīcā  
    white_journal, pievienojam to.
```

```
        white_journal[subject] = {}
```

```
        white_journal[subject][student_name] = average_grade # Pievienojam skolēna vārdu  
    un vidējo atzīmi attiecīgajam priekšmetam baltajā žurnālā.
```

```
    return white_journal # Atgriež baltu žurnālu.
```

```
def generate_testimony(student_data):
```

```
    # Šī funkcija ģenerē liecību, pamatojoties uz skolēna datiem.
```

```
    # Tas atgriež simbolu virkni, kas satur liecību.
```

```
    # Ja skolēnam vidēja atzīme ir 0 (visas atzīmes ir 0), tad tas priekšmēts neparādīsies (vajag  
    ierākstīt 0 priekšmētos kurus nav skolēnam).
```

```
    testimony = ""
```

```
    for student_name, data in student_data.items():
```

```
        student_class = data['Klase']
```

```
        subject_grades = data['Marks']
```

```
        testimony = testimony + f"Liecība ID: {student_name} (Klase: {student_class}):\n"
```

```
        for subject, grades in subject_grades.items():
```

```
            average_grade = calculate_average(grades)
```

```
            if average_grade != 0:
```

```
                testimony = testimony + f"Priekšmets: {subject}, Vidējā atzīme: {average_grade}\n"
```

```
        testimony = testimony + "\n"
```

```
    return testimony
```

```

def write_testimony_to_file(testimony, file_name):
    # Ieraksta ģenerēto liecību .txt failā (file_name)
    # Tas neatgriež nekādu vērtību (atgriež None)

    with open(file_name, 'w', encoding='utf-8') as file:
        file.write(testimony)

# -----
# Galvenā programmas daļa
# -----

file_name = "C:\\Users\\User\\Desktop\\student_data.csv" # Aizvietojiet ar sava CSV faila
nosaukumu
student_data = read_student_data(file_name)

# TESTĒŠANAI -----
# VAR PĀRBAUDĪT KATRU SKOLENU UN VIŅAS VISAS ATZĪMES
student_code = '10000000000' # Aizstāt ar skolēna kodu, kuram vēlaties piekļūt
print_student_grades_by_code(student_data, student_code)
# TESTĒŠANAI -----

# Izveidot un izdrukāt balto žurnālu
white_journal = generate_white_journal(student_data)

# Izveidot liecību un ierakstīt to failā
testimony = generate_testimony(student_data)
file_name = "C:\\Users\\User\\Desktop\\VardsUzvards.txt" # Aizvietojiet ar sava TXT faila
nosaukumu
write_testimony_to_file(testimony, file_name)

```

```
print("\nLiecības tika uzģenerētas un ierakstītas failā VardsUzvards.txt") # Informācija lietotājam
```

## Testa piemēri:

1)

```
Atzīmes skolēnam: Jānis Balodis
Priekšmets: Matemātika, Atzīmes: [1, 1, 1, 1, 1]
Priekšmets: Dabaszinības, Atzīmes: [7, 6, 7, 7, 6]
Priekšmets: Vēsture, Atzīmes: [10, 10, 10]
Priekšmets: Ģeogrāfija, Atzīmes: [5, 6, 7]
Priekšmets: Datorika, Atzīmes: [4, 4, 4]

Liecības tika uzģenerētas un ierakstītas failā VardsUzvards.txt
```

2)

Liecība ID: 10000000000 (Klase: 10A):

Priekšmets: Matemātika, Vidējā atzīme: 1

Priekšmets: Dabaszinības, Vidējā atzīme: 7

Priekšmets: Vēsture, Vidējā atzīme: 10

Priekšmets: Ģeogrāfija, Vidējā atzīme: 6

Priekšmets: Datorika, Vidējā atzīme: 4

Liecība ID: 10000000001 (Klase: 11B):

Priekšmets: Matemātika, Vidējā atzīme: 5

Priekšmets: Dabaszinības, Vidējā atzīme: 6

Priekšmets: Vēsture, Vidējā atzīme: 9

Priekšmets: Ģeogrāfija, Vidējā atzīme: 8

Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000002 (Klase: 1A):

Priekšmets: Matemātika, Vidējā atzīme: 2

Priekšmets: Dabaszinības, Vidējā atzīme: 2

Priekšmets: Vēsture, Vidējā atzīme: 6



Liecība ID: 10000000003 (Klase: 6A):

Priekšmets: Matemātika, Vidējā atzīme: 4

Priekšmets: Dabaszinības, Vidējā atzīme: 5

Priekšmets: Vēsture, Vidējā atzīme: 6

Priekšmets: Datorika, Vidējā atzīme: 8

Liecība ID: 10000000004 (Klase: 6A):

Priekšmets: Matemātika, Vidējā atzīme: 4

Priekšmets: Dabaszinības, Vidējā atzīme: 7

Priekšmets: Vēsture, Vidējā atzīme: 7

Priekšmets: Datorika, Vidējā atzīme: 7

Liecība ID: 10000000005 (Klase: 1A):

Priekšmets: Matemātika, Vidējā atzīme: 1

Priekšmets: Dabaszinības, Vidējā atzīme: 2

Priekšmets: Vēsture, Vidējā atzīme: 2

Liecība ID: 10000000006 (Klase: 10A):

Priekšmets: Matemātika, Vidējā atzīme: 10

Priekšmets: Dabaszinības, Vidējā atzīme: 9

Priekšmets: Vēsture, Vidējā atzīme: 6

Priekšmets: Ģeogrāfija, Vidējā atzīme: 5

Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000007 (Klase: 12A):

Priekšmets: Matemātika, Vidējā atzīme: 10

Priekšmets: Dabaszinības, Vidējā atzīme: 10

Priekšmets: Vēsture, Vidējā atzīme: 10

Priekšmets: Ģeogrāfija, Vidējā atzīme: 10

Priekšmets: Datorika, Vidējā atzīme: 10

Liecība ID: 10000000008 (Klase: 12A):

Priekšmets: Matemātika, Vidējā atzīme: 9

Priekšmets: Dabaszinības, Vidējā atzīme: 4

Priekšmets: Vēsture, Vidējā atzīme: 9

Priekšmets: Ģeogrāfija, Vidējā atzīme: 7

Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000009 (Klase: 6A):

Priekšmets: Matemātika, Vidējā atzīme: 4

Priekšmets: Dabaszinības, Vidējā atzīme: 9

Priekšmets: Vēsture, Vidējā atzīme: 5

Priekšmets: Datorika, Vidējā atzīme: 9

3)

Liecība ID: 10000000000 (Klase: 10A):  
Priekšmets: Matemātika, Vidējā atzīme: 1  
Priekšmets: Dabaszinības, Vidējā atzīme: 7  
Priekšmets: Vēsture, Vidējā atzīme: 10  
Priekšmets: Ģeogrāfija, Vidējā atzīme: 6  
Priekšmets: Datorika, Vidējā atzīme: 4

Liecība ID: 10000000001 (Klase: 11B):  
Priekšmets: Matemātika, Vidējā atzīme: 5  
Priekšmets: Dabaszinības, Vidējā atzīme: 6  
Priekšmets: Vēsture, Vidējā atzīme: 9  
Priekšmets: Ģeogrāfija, Vidējā atzīme: 8  
Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000002 (Klase: 1A):  
Priekšmets: Matemātika, Vidējā atzīme: 2  
Priekšmets: Dabaszinības, Vidējā atzīme: 2  
Priekšmets: Vēsture, Vidējā atzīme: 6

Liecība ID: 10000000003 (Klase: 6A):  
Priekšmets: Matemātika, Vidējā atzīme: 4  
Priekšmets: Dabaszinības, Vidējā atzīme: 5  
Priekšmets: Vēsture, Vidējā atzīme: 6  
Priekšmets: Datorika, Vidējā atzīme: 8

Liecība ID: 10000000004 (Klase: 6A):  
Priekšmets: Matemātika, Vidējā atzīme: 4  
Priekšmets: Dabaszinības, Vidējā atzīme: 7  
Priekšmets: Vēsture, Vidējā atzīme: 7  
Priekšmets: Datorika, Vidējā atzīme: 7

Liecība ID: 10000000005 (Klase: 1A):  
Priekšmets: Matemātika, Vidējā atzīme: 1  
Priekšmets: Dabaszinības, Vidējā atzīme: 2  
Priekšmets: Vēsture, Vidējā atzīme: 2

Liecība ID: 10000000006 (Klase: 10A):  
Priekšmets: Matemātika, Vidējā atzīme: 10  
Priekšmets: Dabaszinības, Vidējā atzīme: 9  
Priekšmets: Vēsture, Vidējā atzīme: 6  
Priekšmets: Ģeogrāfija, Vidējā atzīme: 5  
Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000007 (Klase: 12A):  
Priekšmets: Matemātika, Vidējā atzīme: 10  
Priekšmets: Dabaszinības, Vidējā atzīme: 10  
Priekšmets: Vēsture, Vidējā atzīme: 10  
Priekšmets: Ģeogrāfija, Vidējā atzīme: 10  
Priekšmets: Datorika, Vidējā atzīme: 10

Liecība ID: 10000000008 (Klase: 12A):  
Priekšmets: Matemātika, Vidējā atzīme: 9  
Priekšmets: Dabaszinības, Vidējā atzīme: 4  
Priekšmets: Vēsture, Vidējā atzīme: 9  
Priekšmets: Ģeogrāfija, Vidējā atzīme: 7  
Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000009 (Klase: 6A):  
Priekšmets: Matemātika, Vidējā atzīme: 4  
Priekšmets: Dabaszinības, Vidējā atzīme: 9  
Priekšmets: Vēsture, Vidējā atzīme: 5  
Priekšmets: Datorika, Vidējā atzīme: 9

# PU1. uzdevums

Papildināt 1.programmu, kas atļauj skolēnu atzīmju ievadi un rediģēšanu pa vienai atzīmei vienam skolēnam. (5 punkti).

## Kods:

```
# Programmas nosaukums: Skolas sekmju uzskaitē ar skolēnu atzīmju ievadi un rediģēšanu
```

```
# PU1. uzdevums (1MPR14_Vladislavs_Babaņins)
```

```
# Uzdevuma formulējums: Papildināt 1.programmu, kas atļauj skolēnu atzīmju ievadi un rediģēšanu pa vienai atzīmei vienam skolēnam. (5 punkti).
```

```
# Programmas autors: Vladislavs Babaņins
```

```
# Versija 1.0
```

```
"""
```

Programma tika papildināta ar rediģēšanas iespēju.

Lietotājs ievada Personas ID un tiek parādīts visu atzīmju saraksts, lietotājas ievada noteikto priekšmetu un pēc tam viņš ievada jaunās atzīmes.

Atzīmes tika mainītas .csv failā un pēc tam tika izveidots .txt fails.

ĒRTĪBAS PĒC PROGRAMMA TĪKA IZVEIDOTI DIVI FAILI

.CSV FAILS, KURU AIZPILD PATS LIETOTĀJS AR ROKU

.TXT FAILS, KUR TĪEK VEIDOTAS LĪECĪBAS UN KURU IZVEIDO PROGRAMMA

TĪKA REALIZĒTS TĪEŠĪ TĀ, JO TĀ IR ĒRTĀK LIETOTĀJAM, NEVISS IEVĀDĪT PROGRAMMA ĻOTI DAUDZ SKOLĒNU VĀRDU, UZVĀRDU, PERSONAS ID UN ATZĪMES.

TĀPĒC PROGRAMMA NAV PAREDZĒTA FUNCKĪJA KAS PAPRASA LIETOTĀJAM IEVĀDĪT SĒKMES VAI SKOLĒNUS.

```
"""
```

```
import csv
```

```

def read_student_data(file_name):

    # Šī funkcija nolasa skolēna datus no CSV faila un sakārto tos vārdnīcā.

    # file_name - ceļš līdz failam (.csv datne).

    # Piemēram:

    # file_name = "C:\\Users\\User\\Desktop\\student_data.csv"

    # Tas atgriež skolēna datu vārdnīcu.

    # return student_data


student_data = {} # Definējam tukšu vārdnīcu


with open(file_name, 'r', newline="", encoding='utf-8-sig') as file:

    # Datu izvilkšana no CSV rindām un kārtošana vārdnīcā

    # Katrs skolēns tiek identificēts ar unikālu skolēna kodu


    reader = csv.DictReader(file, delimiter=';')

    for row in reader:

        # Izvelcam attiecīgos datus no katras CSV faila rindas un sakārtojam tos vārdnīcā


        student_code = row['Personas kods'] # Izvelcam skolēna unikālo ID identifikatoru
        (Personas kodu) no kolonnas "Personas kods"

        student_name = row['Vārds'] # Izvelcam skolēna uzvārdu no rindas "Vārds".

        student_last_name = row['Uzvārds'] # Izvelcam skolēna uzvārdu no rindas "Uzvārds".

        student_class = row['Klase'] # Izvelcam skolēna klasi no rindas "Klase".

        math_grades = [int(grade) for grade in row['Matemātika'].split()] # Izņemam
        matemātikas atzīmes no rindas "Matemātika" un pārveidojam "int" skaitļos

        science_grades = [int(grade) for grade in row['Dabaszinības'].split()] # Izņemam
        dabaszinības atzīmes no rindas "Dabaszinības" un pārveidojam "int" skaitļos

        history_grades = [int(grade) for grade in row['Vēsture'].split()] # Izņemam vēstures
        atzīmes no rindas "Vēsture" un pārveidojam "int" skaitļos

        geo_grades = [int(grade) for grade in row['Ģeogrāfija'].split()] # Izņemam ģeogrāfijas
        atzīmes no rindas "Ģeogrāfija" un pārveidojam "int" skaitļos

```

```
computer_grades = [int(grade) for grade in row['Datorika'].split()] # Izņemam datorika atzīmes no rindas "Datorika" un pārveidojam "int" skaitļos
```

```
# Izveido vārdnīcu ar nosaukumu subject_grades, lai saglabātu katra priekšmeta atzīmes. Vārdnīca ir strukturēta šādi:
```

```
# Priekšmets: Atzīmē priekšmeta nosaukumu ('Matemātika', 'Dabaszinības', 'Vēsture', 'Ģeogrāfija', 'Datorika').
```

```
# Atzīmes: atspoguļo attiecīgajam mācību priekšmetam atbilstošo atzīmju sarakstu.
```

```
subject_grades = {  
    'Matemātika': math_grades,  
    'Dabaszinības': science_grades,  
    'Vēsture': history_grades,  
    'Ģeogrāfija': geo_grades,  
    'Datorika': computer_grades  
}
```

```
# Pievienojam skolēna informāciju un atzīmes vārdnīcai student_data
```

```
student_data[student_code] = {'Vārds': student_name, 'Uzvārds': student_last_name,  
'Klase': student_class, 'Marks': subject_grades}
```

```
return student_data
```

```
def print_student_grades_by_code(student_data, student_code):
```

```
    # Šī funkcija izdrukā skolēna atzīmes, kas identificētas pēc skolēna koda.
```

```
    # student_data: vārdnīca, kurā ir skolēna dati.
```

```
    # studenta_kods: skolēna unikālais identifikators (Skolēna ID (Personas kods)).
```

```
    # Atgriež: None
```

```
    if student_code in student_data: # Pārbaudam, vai skolēna datu vārdnīcā ir šāds skolēna kods (vai tāds skolēns eksistē)
```

```
        student_info = student_data[student_code] # Iegūvam informāciju par doto skolēna kodu
```

```
student_name = student_info['Vārds'] # Iegūvam informāciju par doto skolēna vārdu  
student_surname = student_info['Uzvārds'] # Iegūvam informāciju par doto skolēna  
uzvārdu
```

```
student_grades = student_info['Marks'] # Iegūvam informāciju par doto skolēna  
atzīmi
```

```
print(f"Atzīmes skolēnam: {student_name} {student_surname}") # Uzrakstām  
lietotājam skolēna atzīmes.
```

```
for subject, grades in student_grades.items(): # Ejām ciklā caur katru priekšmetu un  
tām atbilstošām atzīmēm un izdrukām tos
```

```
    print(f"Priekšmets: {subject}, Atzīmes: {grades}") # Izdrukājam priekšmeta  
nosaukumu un tā atzīmes
```

```
else: # Izdrukājam paziņojumu, kas norāda, ka skolēns ar norādīto kodu netika atrasts, ja  
netika atrasts šāds personas kods.
```

```
    print(f"Skolēns ar personas kodu '{student_code}' nav atrasts.")
```

```
def calculate_average(grades):
```

```
    # Aprēķina vidējo atzīmi no atzīmju saraksta.
```

```
    # Tas atgriež vidējo atzīmi kā noapaļotu līdz veselam skaitlim.
```

```
    # (Ja nevajag apaļot, tad return average)
```

```
    # Ja atzīmju saraksts ir tukšs, tas atgriež 0.
```

```
    total = sum(grades)
```

```
    count = len(grades)
```

```
    if count == 0:
```

```
        return 0 # Gadījums, kad nav atzīmēs.
```

```
    average = total / count
```

```
    return round(average)
```

```

def generate_white_journal(student_data):

    # Ģenerē baltu žurnālu, pamatojoties uz studenta datiem.

    # Tas atgriež vārdnīcu, kurā ir priekšmeti kā atslēgas, un skolēnu vārdu un vidējo atzīmju
    vārdnīcu kā vērtības.

    # Atgriež baltu žurnālu.


    white_journal = {} # Inicializējam tukšu vārdnīcu, lai saglabātu balto žurnālu.


    for student_name, data in student_data.items(): # Iterējam caur katru skolēnu
    student_datu vārdnīcā.

        student_class = data['Klase'] # Iegūvam skolēna klasi

        subject_grades = data['Marks'] # Iegūvam skolēna atzīmes


        for subject, grades in subject_grades.items(): # Iterējam caur katru priekšmetu un tā
        atbilstošās atzīmem.

            average_grade = calculate_average(grades) # Aprēķinām katra priekšmeta vidējo
            atzīmi un saglabājam to baltajā žurnālā.

            if subject not in white_journal: # Ja priekšmets vēl nav iekļauts vārdnīcā
            white_journal, pievienojiet to.

                white_journal[subject] = {}

                white_journal[subject][student_name] = average_grade # Pievienojam skolēna vārdu
                un vidējo atzīmi attiecīgajam priekšmetam baltajā žurnālā.


    return white_journal # Atgriež baltu žurnālu.


def generate_testimony(student_data):

    # Šī funkcija ģenerē liecību, pamatojoties uz skolēna datiem.

    # Tas atgriež simbolu virkni, kas satur liecību.

    # Ja skolēnam vidēja atzīme ir 0 (visas atzīmes ir 0), tad tas priekšmēts neparādīsies (vajag
    ierakstīt 0 priekšmētos kurus nav skolēnam).


    testimony = ""

```

```

for student_name, data in student_data.items():
    student_class = data['Klase']
    subject_grades = data['Marks']

    testimony = testimony + f"Liecība ID: {student_name} (Klase: {student_class}):\n"
    for subject, grades in subject_grades.items():
        average_grade = calculate_average(grades)
        if average_grade != 0:
            testimony = testimony + f"Priekšmets: {subject}, Vidējā atzīme: {average_grade}\n"
    testimony = testimony + "\n"

return testimony

```

```

def write_testimony_to_file(testimony, file_name):
    # Ieraksta ģenerēto liecību .txt failā (file_name)
    # Tas neatgriež nekādu vērtību (atgriež None)

    with open(file_name, 'w', encoding='utf-8') as file:
        file.write(testimony)

```

```

def change_student_marks(file_name, student_id, subject, new_marks):
    # Izmainā skolēna atzīmes noteiktā priekšmeta uz jaunajām.
    # Atgriež True, ja netika atrāsta neviena kļūda.
    # Atgriež False, ja tika atrāsta kļūda.
    # file_name - .csv failā ceļš
    # student_id - skolēna personas kods
    # subject - skolas priekšmets
    # new_marks - jaunās atzīmes

```



```

student_data = read_student_data(file_name) # Nolasa skolēnu datus no CSV faila

if student_id in student_data: # Pārbauda, vai skolēna ID ir iekšā student_data
    # Iegūst skolēna informāciju un atzīmes
    student_info = student_data[student_id]
    student_grades = student_info['Marks']

    # Pārbauda, vai priekšmets pastāv skolēna atzīmēs
    if subject in student_grades:
        # Atjaunina atzīmes ar jaunajām atzīmēm
        student_grades[subject] = new_marks

    # Ieraksta atjauninātos skolēna datus atpakaļ CSV failā
    with open(file_name, 'w', newline="", encoding='utf-8-sig') as file:
        kolonnas_nosaukumi = ['Personas kods', 'Vārds', 'Uzvārds', 'Klase', 'Matemātika',
                              'Dabaszinības', 'Vēsture', 'Ģeogrāfija', 'Datorika']

        writer = csv.DictWriter(file, fieldnames=kolonnas_nosaukumi, delimiter=';')
        writer.writeheader()

        for student_code, student_info in student_data.items():
            # Ieraksta skolēna informāciju un atzīmes CSV failā

            writer.writerow({
                'Personas kods': student_code, # Ierakstam skolēna unikālo identifikatoru
                kolonnā "Personas kods".

                'Vārds': student_info['Vārds'], # Ailē 'Vārds' ierakstiet skolēna vārdu

                'Uzvārds': student_info['Uzvārds'], # Ailē 'Uzvārds' ierakstiet skolēna uzvārdu

                'Klase': student_info['Klase'], # Ierakstam skolēna klasi kolonnā "Klase".

                'Matemātika': ' '.join(str(grade) for grade in
student_info['Marks']['Matemātika']), # Konvertējam un savienojam visas matemātikas
atzīmes ar atstarpi atdalītā simbolu virknē un ierakstīsim to rindiņā "Matemātika"

```

```

        'Dabaszinibas': ' '.join(str(grade) for grade in
student_info['Marks']['Dabaszinibas']), # Konvertējam un savienojam visas dabaszinibas
atzīmes ar atstarpi atdalītā simbolu virknē un ierakstīsim to rindiņā "Dabaszinibas"

        'Vesture': ' '.join(str(grade) for grade in student_info['Marks']['Vesture']), #
Konvertējam un savienojam visas vēstures atzīmes ar atstarpi atdalītā simbolu virknē un
ierakstīsim to rindiņā "Vesture"

        'Ģeogrāfija': ' '.join(str(grade) for grade in student_info['Marks']['Ģeogrāfija']),
# Konvertējam un savienojam visas ģeogrāfijas atzīmes ar atstarpi atdalītā simbolu virknē un
ierakstīsim to rindiņā "Ģeogrāfija"

        'Datorika': ' '.join(str(grade) for grade in student_info['Marks']['Datorika']) #
Konvertējam un savienojam visas datorikas atzīmes ar atstarpi atdalītā simbolu virknē un
ierakstīsim to rindiņā "Datorika"

    })

```

```

    return True # Atgriež True, lai norādītu uz veiksmīgu skolēna atzīmju izmaiņšanu

```

```

else:

```

```

    print(f"Skolēns ar personas kodu '{student_id}' nav atrasts priekšmetā '{subject}'")

```

```

else:

```

```

    print(f"Skolēns ar personas kodu '{student_id}' nav atrasts.")

```

```

    return False # Atgriež False, lai norādītu uz neveiksmīgu skolēnu atzīmju izmaiņšanu

```

```

# -----

```

```

# Galvenā programmas daļa

```

```

# -----

```

```

file_name = "C:\\Users\\User\\Desktop\\student_data.csv"

```

```

student_data = read_student_data(file_name)

```

```

student_code = ""

```

```

while student_code.lower() != "nē": # Turpinām ciklu, līdz ievade ir "nē".

```

```

# Lūdzam lietotājam ievadīt skolēna personas kodu vai "nē", lai izietu no cikla

student_code = input("Lūdzu, ievadiet skolēna personas kodu, kuras atzīmes gribāt
izmainīt, ja negribat, tad ierakstiet \"Nē\" ==> ")

if student_code.lower() != "nē" and student_code in student_data: # Ja ievade nav "Nē"
(neatkarīgi no reģistra) un tāds skolēns eksistē, tad...

    print_student_grades_by_code(student_data, student_code) # ... izsaucām funkciju, lai
izdrukātu skolēna atzīmes, pamatojoties uz norādīto Personas kodu


subject = input("Lūdzu, ievadiet priekšmetu: ") # Lūdziet lietotājam skolas priekšmētu

if subject in student_data[student_code]['Marks']: # Ja priekšmets pastāv skolēna
"atzīmju ierakstos"

    # Lūdzam lietotājam ievadīt jaunās atzīmes, atdalot tās ar atstarpēm

    new_marks = input("Lūdzu, ievadiet jaunās atzīmes atdalītas ar atstarpi: ")

    # Pārvēršam atzīmju ievades virkni par veselu skaitļu sarakstu

    new_marks = [int(grade) for grade in new_marks.split()]

    if change_student_marks(file_name, student_code, subject, new_marks): # Izsaucām
funkciju, lai mainītu skolēna atzīmes norādītajā priekšmetā

        pass

    else:

        print("Notika neparedzēta kļūda!") # Testēšanai

    else:

        print(f"Skolēnam ar personas kodu '{student_code}' nav atzīmes par priekšmētu
'{subject}'.") # Ja netika atrasts

    else:

        if student_code.lower() == "nē":

            pass

        else:

            print("Kļūda! Tāds skolēns neeksistē!")

        break


file_name = "C:\\Users\\User\\Desktop\\student_data.csv" # Aizvietojiet ar sava CSV faila
nosaukumu

```

```

student_data = read_student_data(file_name)

# Ģenerējam un izdrukājam balto žurnālu
white_journal = generate_white_journal(student_data)

# Izveidojam liecību un ierakstiet to failā
testimony = generate_testimony(student_data)

file_name = "C:\\Users\\User\\Desktop\\VardsUzvards.txt" # Aizvietojiet ar sava TXT faila
nosaukumu

write_testimony_to_file(testimony, file_name)

print("\nLiecības tika uzģenerētas un ierakstītas failā VardsUzvards.txt")

```

## Testa piemēri:

1)

```

Lūdzu, ievadiet skolēna personas kodu, kuras atzīmes gribāt izmainīt, ja negribat, tad ierakstiet "Nē" ==> 10000000000
Atzīmes skolēnam: Jānis Balodis
Priekšmets: Matemātika, Atzīmes: [1, 1, 1, 1, 1]
Priekšmets: Dabaszinības, Atzīmes: [7, 6, 7, 7, 6]
Priekšmets: Vēsture, Atzīmes: [10, 10, 10]
Priekšmets: Ģeogrāfija, Atzīmes: [5, 6, 7]
Priekšmets: Datorika, Atzīmes: [4, 4, 4]
Lūdzu, ievadiet priekšmetu: Matemātika
Lūdzu, ievadiet jaunās atzīmes atdalītas ar atstarpi: 10 10 10 8 4
Lūdzu, ievadiet skolēna personas kodu, kuras atzīmes gribāt izmainīt, ja negribat, tad ierakstiet "Nē" ==> Nē

Liecības tika uzģenerētas un ierakstītas failā VardsUzvards.txt

```

Liecība ID: 10000000000 (Klase: 10A):

Priekšmets: Matemātika, Vidējā atzīme: 8

Priekšmets: Dabaszinības, Vidējā atzīme: 7

Priekšmets: Vēsture, Vidējā atzīme: 10

Priekšmets: Ģeogrāfija, Vidējā atzīme: 6

Priekšmets: Datorika, Vidējā atzīme: 4

Liecība ID: 10000000001 (Klase: 11B):

Priekšmets: Matemātika, Vidējā atzīme: 5

Priekšmets: Dabaszinības, Vidējā atzīme: 6

Priekšmets: Vēsture, Vidējā atzīme: 9

Priekšmets: Ģeogrāfija, Vidējā atzīme: 8

Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000002 (Klase: 1A):

Priekšmets: Matemātika, Vidējā atzīme: 2

Priekšmets: Dabaszinības, Vidējā atzīme: 2

Priekšmets: Vēsture, Vidējā atzīme: 6

Liecība ID: 10000000003 (Klase: 6A):

Priekšmets: Matemātika, Vidējā atzīme: 4

Priekšmets: Dabaszinības, Vidējā atzīme: 5

Priekšmets: Vēsture, Vidējā atzīme: 6

Priekšmets: Datorika, Vidējā atzīme: 8

Liecība ID: 10000000004 (Klase: 6A):

Priekšmets: Matemātika, Vidējā atzīme: 4

Priekšmets: Dabaszinības, Vidējā atzīme: 7

Priekšmets: Vēsture, Vidējā atzīme: 7

Priekšmets: Datorika, Vidējā atzīme: 7

Liecība ID: 10000000005 (Klase: 1A):

Priekšmets: Matemātika, Vidējā atzīme: 1

Priekšmets: Dabaszinības, Vidējā atzīme: 2

Priekšmets: Vēsture, Vidējā atzīme: 2

Liecība ID: 10000000006 (Klase: 10A):

Priekšmets: Matemātika, Vidējā atzīme: 10

Priekšmets: Dabaszinības, Vidējā atzīme: 9

Priekšmets: Vēsture, Vidējā atzīme: 6

Priekšmets: Ģeogrāfija, Vidējā atzīme: 5

Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000007 (Klase: 12A):

Priekšmets: Matemātika, Vidējā atzīme: 10

Priekšmets: Dabaszinības, Vidējā atzīme: 10

Priekšmets: Vēsture, Vidējā atzīme: 10

Priekšmets: Ģeogrāfija, Vidējā atzīme: 10

Priekšmets: Datorika, Vidējā atzīme: 10

Liecība ID: 10000000008 (Klase: 12A):

Priekšmets: Matemātika, Vidējā atzīme: 9

Priekšmets: Dabaszinības, Vidējā atzīme: 4

Priekšmets: Vēsture, Vidējā atzīme: 9

Priekšmets: Ģeogrāfija, Vidējā atzīme: 7

Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000009 (Klase: 6A):

Priekšmets: Matemātika, Vidējā atzīme: 4

Priekšmets: Dabaszinības, Vidējā atzīme: 9

Priekšmets: Vēsture, Vidējā atzīme: 5

Priekšmets: Datorika, Vidējā atzīme: 9

1	Personas kods;Vārds;Uzvārds;Klase;Matemātika;Dabaszinības;Vēsture;Ģeogrāfija;Datorika
2	10000000000;Jānis;Balodis;10A;10 10 10 8 4;7 6 7 7 6;10 10 10;5 6 7;4 4 4
3	10000000001;Johanna;Rozenberga;11B;4 6 5 5 5;6 8 4 5;10 9 7;8 8 7;5 6 4
4	10000000002;Viktors;Katjuščiks;1A;1 2 2 2;1 4 3 1;4 8 5;0;0
5	10000000003;Jurijs;Ribņikovs;6A;5 2 1 1 10;4 5 6 6;6 6 6;0;10 10 5
6	10000000004;Soņa;Marmeladova;6A;5 6 5 1 2;8 7 6 6;6 9 6;0;4 7 10
7	10000000005;Poligrāfs;Šarikovs;1A;1 1 1 1;1 2 2 3;1 2 4;0;0
8	10000000006;Pāvels;Čičikovs;10A;10 10 10 10;9 8 10 10 7;4 6 7;4 4 8;5 6 4
9	10000000007;Boriss;Bojaršinovs;12A;10 10 10 10;10 10 10 10;10 10 10 10;10 10 10 10
10	10000000008;Antuāns;Rokatens;12A;10 9 8 9;4 4 4 4;9 9 9 10;6 6 7 8;4 4 5 6
11	10000000009;Stepans;Pļuškins;6A;5 5 5 3 4;8 9 9 9;8 1 5;0;10 7 9

2)

```

Lūdzu, ievadiet skolēna personas kodu, kuras atzīmes gribāt izmainīt, ja negribat, tad ierakstiet "Nē" ==> 10000000000
Atzīmes skolēnam: Jānis Balodis
Priekšmets: Matemātika, Atzīmes: [10, 10, 10, 8, 4]
Priekšmets: Dabaszinības, Atzīmes: [7, 6, 7, 7, 6]
Priekšmets: Vēsture, Atzīmes: [10, 10, 10]
Priekšmets: Ģeogrāfija, Atzīmes: [5, 6, 7]
Priekšmets: Datorika, Atzīmes: [4, 4, 4]
Lūdzu, ievadiet priekšmetu: Datorika
Lūdzu, ievadiet jaunās atzīmes atdalītas ar atstarpi: 10 10 10
Lūdzu, ievadiet skolēna personas kodu, kuras atzīmes gribāt izmainīt, ja negribat, tad ierakstiet "Nē" ==> 10000000000
Atzīmes skolēnam: Jānis Balodis
Priekšmets: Matemātika, Atzīmes: [10, 10, 10, 8, 4]
Priekšmets: Dabaszinības, Atzīmes: [7, 6, 7, 7, 6]
Priekšmets: Vēsture, Atzīmes: [10, 10, 10]
Priekšmets: Ģeogrāfija, Atzīmes: [5, 6, 7]
Priekšmets: Datorika, Atzīmes: [4, 4, 4]
Lūdzu, ievadiet priekšmetu: Matemātika
Lūdzu, ievadiet jaunās atzīmes atdalītas ar atstarpi: 1 1 1 1 1
Lūdzu, ievadiet skolēna personas kodu, kuras atzīmes gribāt izmainīt, ja negribat, tad ierakstiet "Nē" ==> nē

Liecības tika uzģenerētas un ierakstītas failā VardsUzvards.txt

```

(izmaiņas netiek parādītas uzreiz lietotājam, jo fails mainīsies kad tiek uzrakstīts "nē")

```

1 Personas kods;Vārds;Uzvārds;Klase;Matemātika;Dabaszinības;Vēsture;Ģeogrāfija;Datorika
2 10000000000;Jānis;Balodis;10A;1 1 1 1 1;7 6 7 7 6;10 10 10;5 6 7;10 10 10
3 10000000001;Johanna;Rozenberga;11B;4 6 5 5 5;6 8 4 5;10 9 7;8 8 7;5 6 4
4 10000000002;Viktors;Katjuščiks;1A;1 2 2 2;1 4 3 1;4 8 5;0;0
5 10000000003;Jurijs;Ribņikovs;6A;5 2 1 1 10;4 5 6 6;6 6 6;0;10 10 5
6 10000000004;Soņa;Marmeladova;6A;5 6 5 1 2;8 7 6 6;6 9 6;0;4 7 10
7 10000000005;Poligrāfs;Šarikovs;1A;1 1 1 1;1 2 2 3;1 2 4;0;0
8 10000000006;Pāvels;Čičikovs;10A;10 10 10 10;9 8 10 10 7;4 6 7;4 4 8;5 6 4
9 10000000007;Boriss;Bojaršinovs;12A;10 10 10 10;10 10 10 10;10 10 10 10;10 10 10 10
10 10000000008;Antuāns;Rokatens;12A;10 9 8 9;4 4 4 4;9 9 9 10;6 6 7 8;4 4 5 6
11 10000000009;Stepans;Pluškīns;6A;5 5 5 3 4;8 9 9 9;8 1 5;0;10 7 9

```

Liecība ID: 10000000000 (Klase: 10A):  
 Priekšmets: Matemātika, Vidējā atzīme: 1  
 Priekšmets: Dabaszinības, Vidējā atzīme: 7  
 Priekšmets: Vēsture, Vidējā atzīme: 10  
 Priekšmets: Ģeogrāfija, Vidējā atzīme: 6  
 Priekšmets: Datorika, Vidējā atzīme: 10

3)

```

Lūdzu, ievadiet skolēna personas kodu, kuras atzīmes gribāt izmainīt, ja negribat, tad ierakstiet "Nē" ==> nē

Liecības tika uzģenerētas un ierakstītas failā VardsUzvards.txt

```

4)

```

Lūdzu, ievadiet skolēna personas kodu, kuras atzīmes gribāt izmainīt, ja negribat, tad ierakstiet "Nē" ==> Labi
Kļūda! Tāds skolēns neeksistē!

Liecības tika uzģenerētas un ierakstītas failā VardsUzvards.txt

```

5)

```

Lūdzu, ievadiet skolēna personas kodu, kuras atzīmes gribāt izmainīt, ja negribat, tad ierakstiet "Nē" ==> 10000000009
Atzīmes skolēnam: Stepans Pluškīns
Priekšmets: Matemātika, Atzīmes: [5, 5, 5, 3, 4]
Priekšmets: Dabaszinības, Atzīmes: [8, 9, 9, 9]
Priekšmets: Vēsture, Atzīmes: [8, 1, 5]
Priekšmets: Ģeogrāfija, Atzīmes: [0]
Priekšmets: Datorika, Atzīmes: [10, 7, 9]
Lūdzu, ievadiet priekšmetu: Datorika
Lūdzu, ievadiet jaunās atzīmes atdalītas ar atstarpi: 10 10 10
Lūdzu, ievadiet skolēna personas kodu, kuras atzīmes gribāt izmainīt, ja negribat, tad ierakstiet "Nē" ==> Nē

Liecības tika uzģenerētas un ierakstītas failā VardsUzvards.txt

```

1	Personas kods;Vārds;Uzvārds;Klase;Matemātika;Dabaszinības;Vēsture;Ģeogrāfija;Datorika
2	10000000000;Jānis;Balodis;10A;1 1 1 1 1;7 6 7 7 6;10 10 10;5 6 7;10 10 10
3	10000000001;Johanna;Rozenberga;11B;4 6 5 5 5;6 8 4 5;10 9 7;8 8 7;5 6 4
4	10000000002;Viktors;Katjuščiks;1A;1 2 2 2;1 4 3 1;4 8 5;0;0
5	10000000003;Jurijs;Ribņikovs;6A;5 2 1 1 10;4 5 6 6;6 6 6;0;10 10 5
6	10000000004;Soņa;Marmeladova;6A;5 6 5 1 2;8 7 6 6;6 9 6;0;4 7 10
7	10000000005;Poligrāfs;Šarikovs;1A;1 1 1 1;1 2 2 3;1 2 4;0;0
8	10000000006;Pāvels;Čičikovs;10A;10 10 10 10;9 8 10 10 7;4 6 7;4 4 8;5 6 4
9	10000000007;Boriss;Bojaršinovs;12A;10 10 10 10;10 10 10 10;10 10 10 10;10 10 10 10
10	10000000008;Antuāns;Rokatens;12A;10 9 8 9;4 4 4 4;9 9 9 10;6 6 7 8;4 4 5 6
11	10000000009;Stepans;Pluškins;6A;5 5 3 4;8 9 9 9;8 1 5;0;10 10 10

Liecība ID: 10000000000 (Klase: 10A):

Priekšmets: Matemātika, Vidējā atzīme: 1

Priekšmets: Dabaszinības, Vidējā atzīme: 7

Priekšmets: Vēsture, Vidējā atzīme: 10

Priekšmets: Ģeogrāfija, Vidējā atzīme: 6

Priekšmets: Datorika, Vidējā atzīme: 10

Liecība ID: 10000000001 (Klase: 11B):

Priekšmets: Matemātika, Vidējā atzīme: 5

Priekšmets: Dabaszinības, Vidējā atzīme: 6

Priekšmets: Vēsture, Vidējā atzīme: 9

Priekšmets: Ģeogrāfija, Vidējā atzīme: 8

Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000002 (Klase: 1A):

Priekšmets: Matemātika, Vidējā atzīme: 2

Priekšmets: Dabaszinības, Vidējā atzīme: 2

Priekšmets: Vēsture, Vidējā atzīme: 6

Liecība ID: 10000000003 (Klase: 6A):

Priekšmets: Matemātika, Vidējā atzīme: 4

Priekšmets: Dabaszinības, Vidējā atzīme: 5

Priekšmets: Vēsture, Vidējā atzīme: 6

Priekšmets: Datorika, Vidējā atzīme: 8

Liecība ID: 10000000004 (Klase: 6A):

Priekšmets: Matemātika, Vidējā atzīme: 4



Priekšmets: Dabaszinības, Vidējā atzīme: 7

Priekšmets: Vēsture, Vidējā atzīme: 7

Priekšmets: Datorika, Vidējā atzīme: 7

Liecība ID: 10000000005 (Klase: 1A):

Priekšmets: Matemātika, Vidējā atzīme: 1

Priekšmets: Dabaszinības, Vidējā atzīme: 2

Priekšmets: Vēsture, Vidējā atzīme: 2

Liecība ID: 10000000006 (Klase: 10A):

Priekšmets: Matemātika, Vidējā atzīme: 10

Priekšmets: Dabaszinības, Vidējā atzīme: 9

Priekšmets: Vēsture, Vidējā atzīme: 6

Priekšmets: Ģeogrāfija, Vidējā atzīme: 5

Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000007 (Klase: 12A):

Priekšmets: Matemātika, Vidējā atzīme: 10

Priekšmets: Dabaszinības, Vidējā atzīme: 10

Priekšmets: Vēsture, Vidējā atzīme: 10

Priekšmets: Ģeogrāfija, Vidējā atzīme: 10

Priekšmets: Datorika, Vidējā atzīme: 10

Liecība ID: 10000000008 (Klase: 12A):

Priekšmets: Matemātika, Vidējā atzīme: 9

Priekšmets: Dabaszinības, Vidējā atzīme: 4

Priekšmets: Vēsture, Vidējā atzīme: 9

Priekšmets: Ģeogrāfija, Vidējā atzīme: 7

Priekšmets: Datorika, Vidējā atzīme: 5

Liecība ID: 10000000009 (Klase: 6A):

Priekšmets: Matemātika, Vidējā atzīme: 4

Priekšmets: Dabaszinības, Vidējā atzīme: 9

Priekšmets: Vēsture, Vidējā atzīme: 5

Priekšmets: Datorika, Vidējā atzīme: 10

## 2. uzdevums

Sastādīt programmu, kas nodrošina informācijas ievadi teksta datnē un nolasa informāciju no teksta datnes un parāda to uz ekrāna, ievērojot teksta sadalījumu pa rindkopām.

### Kods:

```
# Programmas nosaukums: Datnes ievade un nolasīšana
```

```
# 2. uzdevums (1MPR14_Vladislavs_Babaņins)
```

```
# Uzdevuma formulējums: Sastādīt programmu, kas nodrošina informācijas ievadi teksta datnē un nolasa informāciju no teksta datnes
```

```
# un parāda to uz ekrāna, ievērojot teksta sadalījumu pa rindkopām.
```

```
# Programmas autors: Vladislavs Babaņins
```

```
# Versija 1.0
```

```
def input_lines_by_user():
```

```
    # Metode, kas paprasā lietotājam ievādīt jaunu tekstu pa rindiņam un atgriež katru rindu, kā saraksta elementu.
```

```
    # (Prasa ievadīt i + 1 rindu, lai 0.rindu būtu uzrakstīta kā 1.rinda).
```

```
    # Kad tiek uzrakstīts "...", tad apstāšana.
```

```
    # Var izmantot datnes nolasīšanai un apstrādei.
```

```
    # Atgriež sarakstu, kurā katrs elements ir rindas teksts (0.elements ir 0.rindas teksts, 1.elements ir 1.rindas teksts utt.)
```

```
    lines = []
```

```
    while True:
```

```
        ievade = input(f'Ievadiet {len(lines) + 1}. rindas tekstu. Ja vēlies pabeigt, ievadi \"...\\": ')
        lines.append(ievade)
```

```
if ievade == "...":  
    break  
    lines.append(ievade)  
  
return lines
```

```
def write_to_new_file_list(datne, lines):  
    # DZĒŠ VISU SATURU NO FAILĀ un ierāksta failā "datne" (piemēram, .txt failā) lines  
    saraksts, kur katrs saraksta elements ir rindas teksts.  
    # lines - saraksts, kur katrs saraksta elements ir rindas teksts.  
    # datne - datnes fails (.txt)  
    # Piemēram:  
    # datne = "C:\\Users\\User\\Desktop\\teksts.txt"  
  
    with open(datne, mode="w", encoding="utf-8") as datne:  
        for line in lines:  
            datne.write(line + "\\n")
```

```
def print_text_from_data_by_rows(datne):  
    # Uzrakstā termināla lietotājam visu tekstu no .txt failā pa rindām.  
    # datne - datnes fails (piemēram, .txt fails)  
    # Piemēram:  
    # datne = "C:\\Users\\User\\Desktop\\teksts.txt"  
  
    with open(datne, mode="r", encoding="utf-8") as datne:  
        for rinda in datne:  
            print(rinda, end="")
```

```
# -----  
# Galvenā programmas daļa  
# -----
```

```
lines = input_lines_by_user() # Izsaucam metodi, kas paprasa lietotājam ievādīt jaunu tekstu  
pa rindiņam.
```

```
datne = "C:\\Users\\User\\Desktop\\\\teksts.txt" # datne - ceļš līdz failām.
```

```
write_to_new_file_list(datne, lines) # NODZĒS VISU TEKSTU DATNĒ. Uzrakstīsim "datne"  
faila "lines" saraksta elementus, kur katru saraksta elementu izvadīsim kā atsevišķu rindiņu.
```

```
print("\\nIerakstītais teksts datnē:") # Šī rinda netiek ierakstīta failā, tas ir tikai informācija  
lietotājam.
```

```
print_text_from_data_by_rows(datne) # Izvadām lietotājam visu tekstu no .txt failā pa  
rindkopām.
```

## Testa piemēri:

1)

```
Ievadiet 1. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Labi  
Ievadiet 2. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Saule, saule  
Ievadiet 3. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Lietus, lietus  
Ievadiet 4. rindas tekstu. Ja vēlies pabeigt, ievadi "...": ...  
  
Ierakstītais teksts datnē:  
Labi  
Saule, saule  
Lietus, lietus
```

2)

```
Ievadiet 1. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Beidzam sadarbību!  
Ievadiet 2. rindas tekstu. Ja vēlies pabeigt, ievadi "...": ...  
  
Ierakstītais teksts datnē:  
Beidzam sadarbību!
```

3)

```
Ievadiet 1. rindas tekstu. Ja vēlies pabeigt, ievadi "...": labi
Ievadiet 2. rindas tekstu. Ja vēlies pabeigt, ievadi "...": labi
Ievadiet 3. rindas tekstu. Ja vēlies pabeigt, ievadi "...": labi
Ievadiet 4. rindas tekstu. Ja vēlies pabeigt, ievadi "...": ababa
Ievadiet 5. rindas tekstu. Ja vēlies pabeigt, ievadi "...": tururu
Ievadiet 6. rindas tekstu. Ja vēlies pabeigt, ievadi "...": lapala
Ievadiet 7. rindas tekstu. Ja vēlies pabeigt, ievadi "...": pampala
Ievadiet 8. rindas tekstu. Ja vēlies pabeigt, ievadi "...": 465545646
Ievadiet 9. rindas tekstu. Ja vēlies pabeigt, ievadi "...": ..
Ievadiet 10. rindas tekstu. Ja vēlies pabeigt, ievadi "...": darida
Ievadiet 11. rindas tekstu. Ja vēlies pabeigt, ievadi "...": jones
Ievadiet 12. rindas tekstu. Ja vēlies pabeigt, ievadi "...": ...
```

Ierakstītais teksts datnē:

```
labi
labi
labi
ababa
tururu
lapala
pampala
465545646
..
darida
jones
```

4)

```
Ievadiet 1. rindas tekstu. Ja vēlies pabeigt, ievadi "...": 123
Ievadiet 2. rindas tekstu. Ja vēlies pabeigt, ievadi "...": 12
Ievadiet 3. rindas tekstu. Ja vēlies pabeigt, ievadi "...": 333
Ievadiet 4. rindas tekstu. Ja vēlies pabeigt, ievadi "...": 5235
Ievadiet 5. rindas tekstu. Ja vēlies pabeigt, ievadi "...": sdg
Ievadiet 6. rindas tekstu. Ja vēlies pabeigt, ievadi "...": sdagsdg
Ievadiet 7. rindas tekstu. Ja vēlies pabeigt, ievadi "...": sdg
Ievadiet 8. rindas tekstu. Ja vēlies pabeigt, ievadi "...": sdasdf
Ievadiet 9. rindas tekstu. Ja vēlies pabeigt, ievadi "...": d
Ievadiet 10. rindas tekstu. Ja vēlies pabeigt, ievadi "...": sfge
Ievadiet 11. rindas tekstu. Ja vēlies pabeigt, ievadi "...": weg
Ievadiet 12. rindas tekstu. Ja vēlies pabeigt, ievadi "...": wgwegw
Ievadiet 13. rindas tekstu. Ja vēlies pabeigt, ievadi "...": gw
Ievadiet 14. rindas tekstu. Ja vēlies pabeigt, ievadi "...":
Ievadiet 15. rindas tekstu. Ja vēlies pabeigt, ievadi "...":
Ievadiet 16. rindas tekstu. Ja vēlies pabeigt, ievadi "...":
Ievadiet 17. rindas tekstu. Ja vēlies pabeigt, ievadi "...":
Ievadiet 18. rindas tekstu. Ja vēlies pabeigt, ievadi "...":
Ievadiet 19. rindas tekstu. Ja vēlies pabeigt, ievadi "...":
Ievadiet 20. rindas tekstu. Ja vēlies pabeigt, ievadi "...": labi
Ievadiet 21. rindas tekstu. Ja vēlies pabeigt, ievadi "...": ...
```

Ierakstītais teksts datnē:

```
123
12
333
5235
sdg
sdagsdg
sdg
sdasdf
d
sfge
weg
wgwegw
gw
```

labi

### 3. uzdevums

Sastādīt programmu, kas nodrošina informācijas papildināšanu teksta datnē un parāda jauno datnes saturu uz ekrāna, ievērojot teksta sadalījumu pa rindkopām.

#### Kods:

```
# Programmas nosaukums: Datnes papildināšana un jaunas informācijas parādīšana
```

```
# 3. uzdevums (1MPR14_Vladislavs_Babaņins)
```

```
# Uzdevuma formulējums: Sastādīt programmu, kas nodrošina informācijas papildināšanu  
teksta datnē un parāda jauno datnes saturu uz ekrāna,
```

```
# ievērojot teksta sadalījumu pa rindkopām.
```

```
# Programmas autors: Vladislavs Babaņins
```

```
# Versija 1.0
```

```
def input_lines_by_user():
```

```
    # Metode, kas paprasā lietotājam ievādīt jaunu tekstu pa rindiņam un atgriež katru rindu,  
    kā saraksta elementu.
```

```
    # (Prasa ievadīt i + 1 rindu, lai 0.rindu būtu uzrakstīta kā 1.rinda).
```

```
    # Kad tiek uzrakstīts "...", tad apstāšana.
```

```
    # Var izmantot datnes nolasīšanai un apstrādei.
```

```
    # Atgriež sarakstu, kurā katrs elements ir rindas teksts (0.elements ir 0.rindas teksts,  
    1.elements ir 1.rindas teksts utt.)
```

```
    lines = []
```

```
    while True:
```

```
        ievade = input(f"Ievadi {len(lines) + 1}. rindas tekstu. Ja vēlies pabeigt, ievadi \"...\": ")
```

```
        if ievade == "...":
```

```
            break
```

```
        lines.append(ievade)
```

```
    return lines
```

```

def write_to_file_list(lines, datne):

    # Ierāksta datne failā (piemēram, .txt failā) lines saraksts, kur katrs saraksta elements ir
    rindas teksts.

    # lines - saraksts, kur katrs saraksta elements ir rindas teksts.

    # datne - datnes fails (.txt)

    # Piemēram:

    # datne = "C:\\Users\\User\\Desktop\\teksts.txt"


    with open(datne, mode="a", encoding="utf-8") as datne:

        for line in lines:

            datne.write(line + "\n")


def print_text_from_data_by_rows(datne):

    # Uzrakstā termināla lietotājam visu tekstu no .txt failā pa rindām.

    # datne - datnes fails (piemēram, .txt fails)

    # Piemēram:

    # datne = "C:\\Users\\User\\Desktop\\teksts.txt"


    with open(datne, mode="r", encoding="utf-8") as datne:

        for rinda in datne:

            print(rinda, end="")


# -----
# Galvenā programmas daļa
# -----


datne = "C:\\Users\\User\\Desktop\\teksts.txt" # Ceļš līdz failām.

```



```
print("Ierakstītais teksts datnē:") # Informācija lietotājam.

print_text_from_data_by_rows(datne) # Parādam lietotājam, kas tagad ir ierakstīts failā.

print() # Lai būtu atstārpe. (Glīti).


lines = input_lines_by_user() # Izsaucam metodi, kas paprasa lietotājam ievādīt jaunu tekstu
pa rindiņam.

write_to_file_list(lines, datne) # Papildināsim "datne" failu ar "lines" sarakstu, kur katru
saraksta elementu izvadīsim kā atsevišķu rindiņu.


print("\nPapildināts teksts datnē:") # Informācija lietotājam.

print_text_from_data_by_rows(datne) # Parādam lietotājam, kas tagad ir ierakstīts failā pēc
tas papildināšanas.
```

## Testa piemēri:

1)

```
Ierakstītais teksts datnē:
Labdien
Esmu lācis

Ievadi 1. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Es mīlu savu sievu
Ievadi 2. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Gyadruabergu
Ievadi 3. rindas tekstu. Ja vēlies pabeigt, ievadi "...": viņa ir liela un skaista
Ievadi 4. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Atā!
Ievadi 5. rindas tekstu. Ja vēlies pabeigt, ievadi "...": ...

Papildināts teksts datnē:
Labdien
Esmu lācis
Es mīlu savu sievu
Gyadruabergu
viņa ir liela un skaista
Atā!
```

2)

```
Ierakstītais teksts datnē:
Labdien
Esmu lācis
Es mīlu savu sievu
Gyadruabergu
viņa ir liela un skaista
Atā!

Ievadi 1. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Labdien atkā!
Ievadi 2. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Šodien mēs gulēsīm!
Ievadi 3. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Jo ir ziema
Ievadi 4. rindas tekstu. Ja vēlies pabeigt, ievadi "...": un mēs esam lāči
Ievadi 5. rindas tekstu. Ja vēlies pabeigt, ievadi "...": ...

Papildināts teksts datnē:
Labdien
Esmu lācis
Es mīlu savu sievu
Gyadruabergu
viņa ir liela un skaista
Atā!
Labdien atkā!
Šodien mēs gulēsīm!
Jo ir ziema
un mēs esam lāči
```

3)

```
Ierakstītais teksts datnē:
Labdien
Esmu lācis
Es mīlu savu sievu
Gyadruabergu
viņa ir liela un skaista
Atā!
Labdien atkā!
Šodien mēs gulēsīm!
Jo ir ziema
un mēs esam lāči

Ievadi 1. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Labvakar
Ievadi 2. rindas tekstu. Ja vēlies pabeigt, ievadi "...": es gribu pildīt mat.analizes mājasdarbu
Ievadi 3. rindas tekstu. Ja vēlies pabeigt, ievadi "...": labi
Ievadi 4. rindas tekstu. Ja vēlies pabeigt, ievadi "...": es arī pildīšu
Ievadi 5. rindas tekstu. Ja vēlies pabeigt, ievadi "...": Atā!
Ievadi 6. rindas tekstu. Ja vēlies pabeigt, ievadi "...": ...

Papildināts teksts datnē:
Labdien
Esmu lācis
Es mīlu savu sievu
Gyadruabergu
viņa ir liela un skaista
Atā!
Labdien atkā!
Šodien mēs gulēsīm!
Jo ir ziema
un mēs esam lāči
Labvakar
es gribu pildīt mat.analizes mājasdarbu
labi
es arī pildīšu
Atā!
```

## 4. uzdevums

Sastādīt programmu, kas nodrošina informācijas pārrakstīšanu (pārkopēšanu) no vienas teksta datnes otrā jaunā teksta datnē.

### Kods:

```
# Programmas nosaukums: Vienas datnes informāciju pārkopēšanu otrā jaunā teksta datnē
```

```
# 4. uzdevums (1MPR14_Vladislavs_Babaņins)
```

```
# Uzdevuma formulējums: Sastādīt programmu, kas nodrošina informācijas pārrakstīšanu  
(pārkopēšanu) no vienas teksta datnes otrā jaunā teksta datnē.
```

```
# Programmas autors: Vladislavs Babaņins
```

```
# Versija 1.0
```

```
def tekstu_parkopesena_no_datne1_to_datne2(datne1, datne2):
```

```
    # Pārkope tekstu no datne1 -> datne2.
```

```
    # Atgriež None
```

```
    # datne1 - no kurā failā gribām paņemt informāciju.
```

```
    # datne2 - kurā failā gribām informāciju no datne1 pārkopēt.
```

```
    # Piemēram:
```

```
    # datne1 = "C:\\Users\\User\\Desktop\\teksts.txt"
```

```
    # datne2 = "C:\\Users\\User\\Desktop\\teksts2.txt"
```

```
    with open(datne1, mode="r", encoding="utf-8") as datne1:
```

```
        with open(datne2, mode="w", encoding="utf-8") as datne2:
```

```
            for rinda in datne1:
```

```
                datne2.write(rinda)
```

```
def print_text_from_data_by_rows(datne):
```

```
    # Uzrakstā termināla lietotājam visu tekstu no .txt failā pa rindam.
```

```
    # datne - datnes fails (piemēram, .txt fails)
```

```
    # Piemēram:
```

```
# datne = "C:\\Users\\User\\Desktop\\teksts.txt"
```

```
with open(datne, mode="r", encoding="utf-8") as datne:
```

```
    for rinda in datne:
```

```
        print(rinda, end="")
```

```
# -----
```

```
# Galvenā programmas daļa
```

```
# -----
```

```
# Ceļi līdz failiem
```

```
datne1 = "C:\\Users\\User\\Desktop\\teksts.txt"
```

```
datne2 = "C:\\Users\\User\\Desktop\\teksts2.txt"
```

```
print("Ierakstītais teksts 1.datnē:") # Informācija lietotājam.
```

```
print_text_from_data_by_rows(datne1) # Parādam lietotājam kāda informācija ir ierakstīta  
failā datne1 (no kurā kopēsim informāciju).
```

```
print("\n") # Lai būtu glīti lietotājam.
```

```
print("Ierakstītais teksts 2.datnē:") # Informācija lietotājam.
```

```
print_text_from_data_by_rows(datne2) # Parādam lietotājam kāda informācija ir ierakstīta  
failā datne2 (uz viņu mēs kopēsim informāciju).
```

```
print("\n") # Lai būtu glīti lietotājam.
```

```
tekstu_parkopesena_no_datne1_to_datne2(datne1, datne2) # Veicām pārkopēšanu.
```

```
print("Ierakstītais teksts 2.datnē pēc pārrakstīšanas:") # Informācija lietotājam.
```

```
print_text_from_data_by_rows(datne2) # Parādam lietotājam kāda informācija ir ierakstīta  
failā datne2 pēc pārkopēšanas (jābut informācijai no datne1).
```

## Testa piemēri:

1)

Ierakstītais teksts 1.datnē:

Labdien

Esmu lācis

Es mīlu savu sievu

Gyadruabergu

viņa ir liela un skaista

Atā!

Labdien atkāļ!

Šodien mēs gulēsim!

Jo ir ziema

un mēs esam lāči

Labvakar

es gribu pildīt mat.analīzes mājasdarbu

labi

es arī pildīšu

Atā!

Ierakstītais teksts 2.datnē:

ln

opn

fafa

affa

Ierakstītais teksts 2.datnē pēc pārrakstīšanas:

Labdien

Esmu lācis

Es mīlu savu sievu

Gyadruabergu

viņa ir liela un skaista

Atā!

Labdien atkāļ!

Šodien mēs gulēsim!

Jo ir ziema

un mēs esam lāči

Labvakar

es gribu pildīt mat.analīzes mājasdarbu

labi

es arī pildīšu

Atā!

2)

Ierakstītais teksts 1.datnē:

Māris

Ierakstītais teksts 2.datnē:

Gundega

Ierakstītais teksts 2.datnē pēc pārrakstīšanas:

Māris

3)

Ierakstītais teksts 1.datnē:

Labi

Labi

Mamma

Ierakstītais teksts 2.datnē:

Vanna

Ierakstītais teksts 2.datnē pēc pārrakstīšanas:

Labi

Labi

Mamma

4)

Ierakstītais teksts 1.datnē:

Kaut kas

Ierakstītais teksts 2.datnē:

Ierakstītais teksts 2.datnē pēc pārrakstīšanas:

Kaut kas

5)

```
Ierakstītais teksts 1.datnē:  
Kaut kas
```

```
Ierakstītais teksts 2.datnē:  
Nekas
```

```
Ierakstītais teksts 2.datnē pēc pārrakstīšanas:  
Kaut kas
```