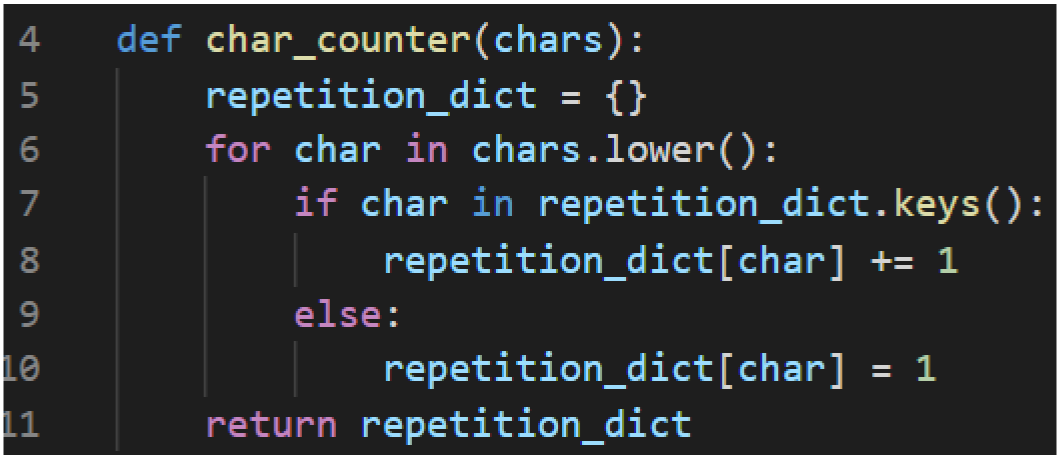
Systemy wbudowane

Laboratorium 2

Łukasz Chwistek 243662

1. Zadanie 1

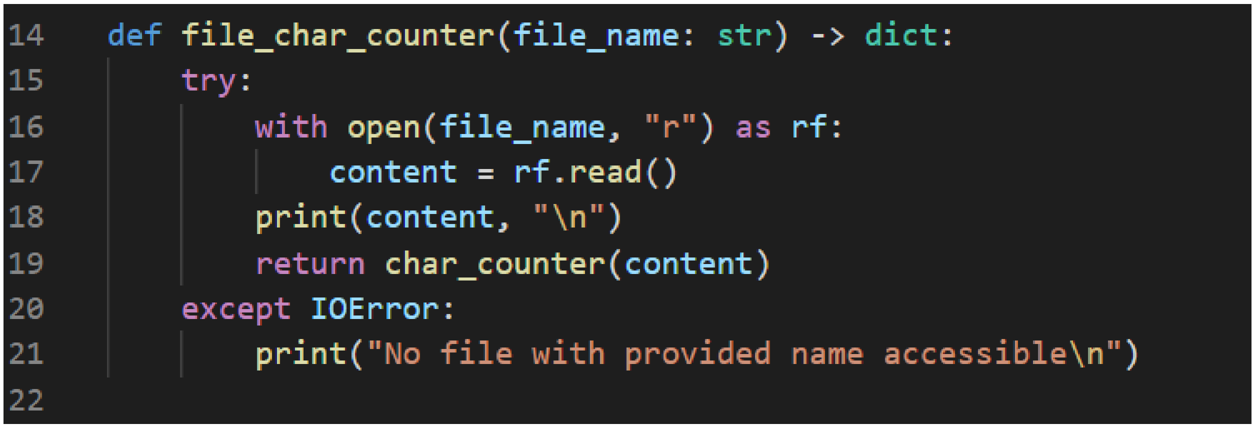
We wczytanym ciągu znaków program liczy indywidualne wystąpienia znaków (ile jest liter a,b,c itd.). Funkcja zwraca dane w postać dictionary gdzie keysstanowią występujący litery, a values ilość ich wystąpień.



```python
def char_counter(chars):
    repetition_dict = {}
    for char in chars.lower():
        if char in repetition_dict.keys():
            repetition_dict[char] += 1
        else:
            repetition_dict[char] = 1
    return repetition_dict
```

2. Zadanie 2

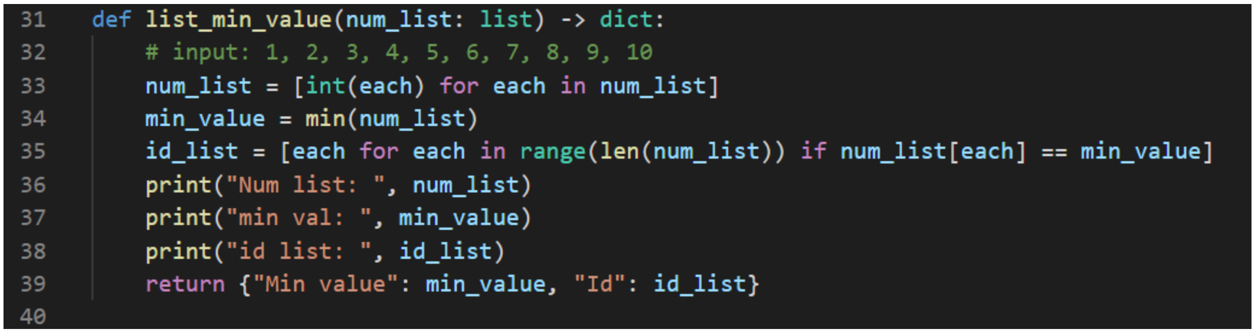
Wyświetlanie zawartości dowolnego, wieloliniowego pliku tekstowego z dysku i liczenie znaków jak zad. 1



```python
def file_char_counter(file_name: str) -> dict:
    try:
        with open(file_name, "r") as rf:
            content = rf.read()
        print(content, "\n")
        return char_counter(content)
    except IOError:
        print("No file with provided name accessible\n")

```

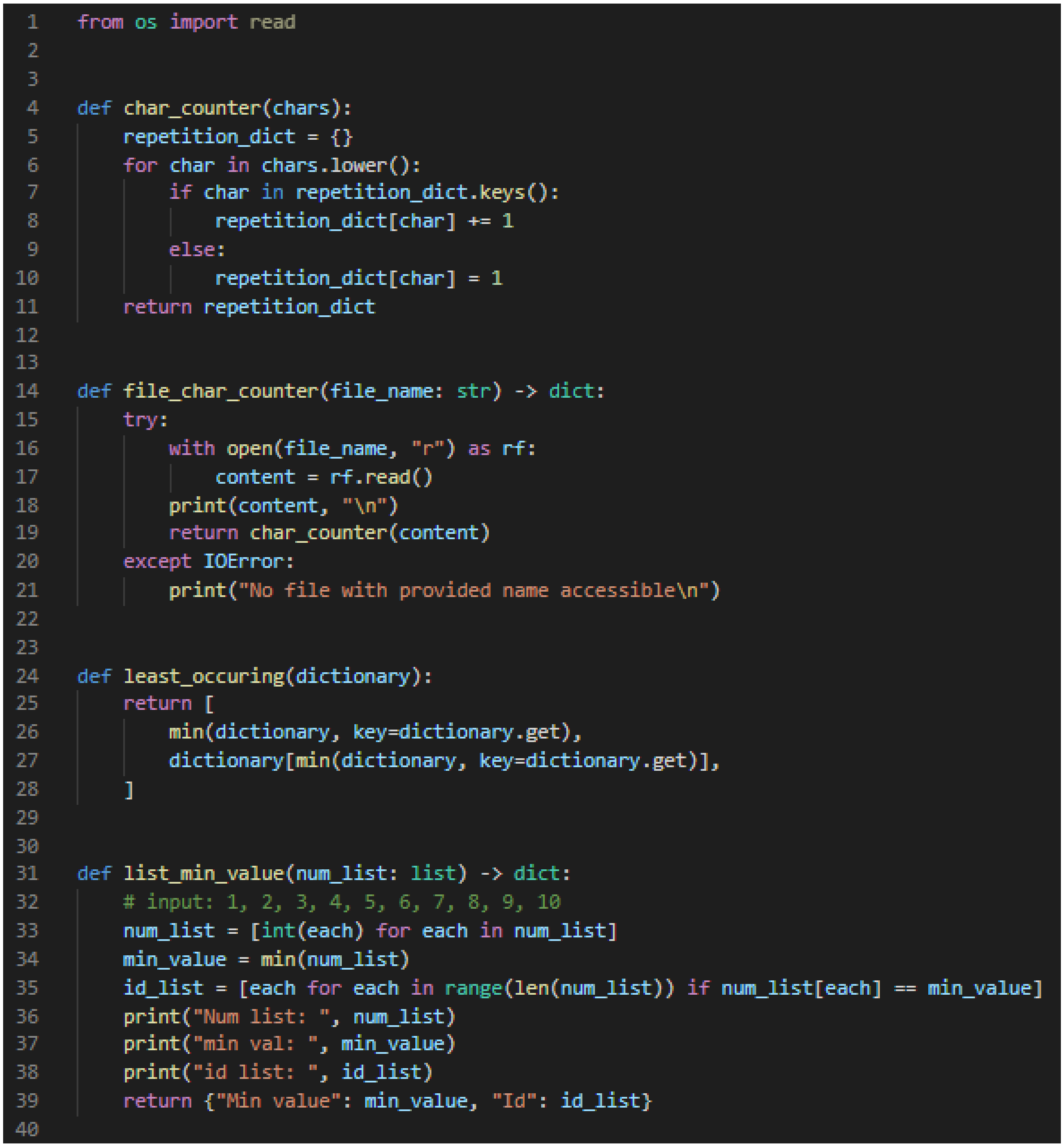
3. Zadanie 3

Ostatnie zadanie polegało na napisaniu funkcji, która wypisuje numery indeksów najmniejszej wartości w liście.

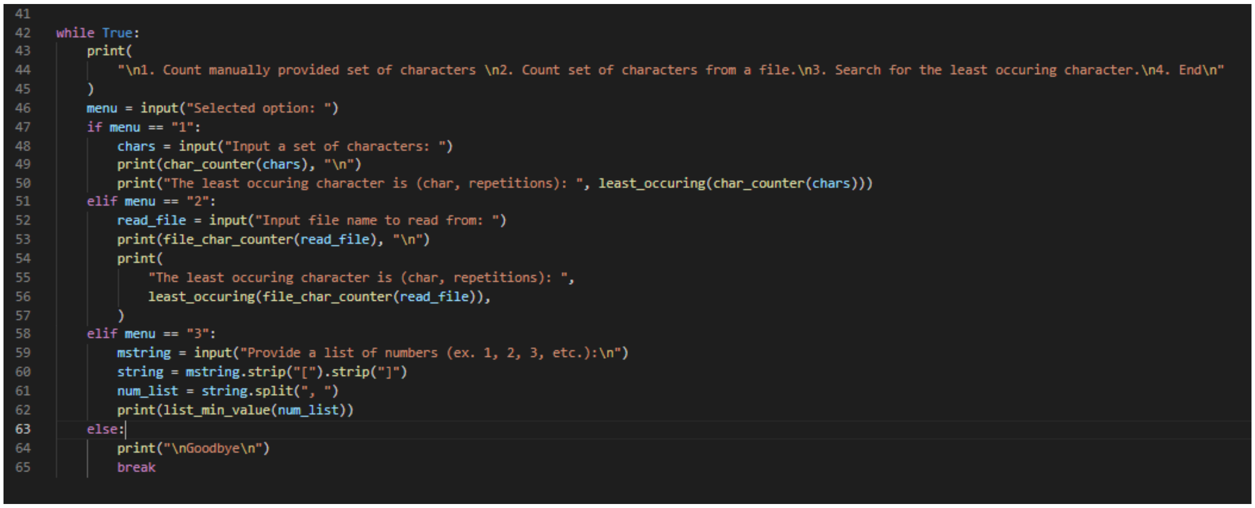


```python
def list_min_value(num_list: list) -> dict:
    # input: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
    num_list = [int(each) for each in num_list]
    min_value = min(num_list)
    id_list = [each for each in range(len(num_list)) if num_list[each] == min_value]
    print("Num list: ", num_list)
    print("min val: ", min_value)
    print("id list: ", id_list)
    return {"Min value": min_value, "Id": id_list}

```

4. Kod programu

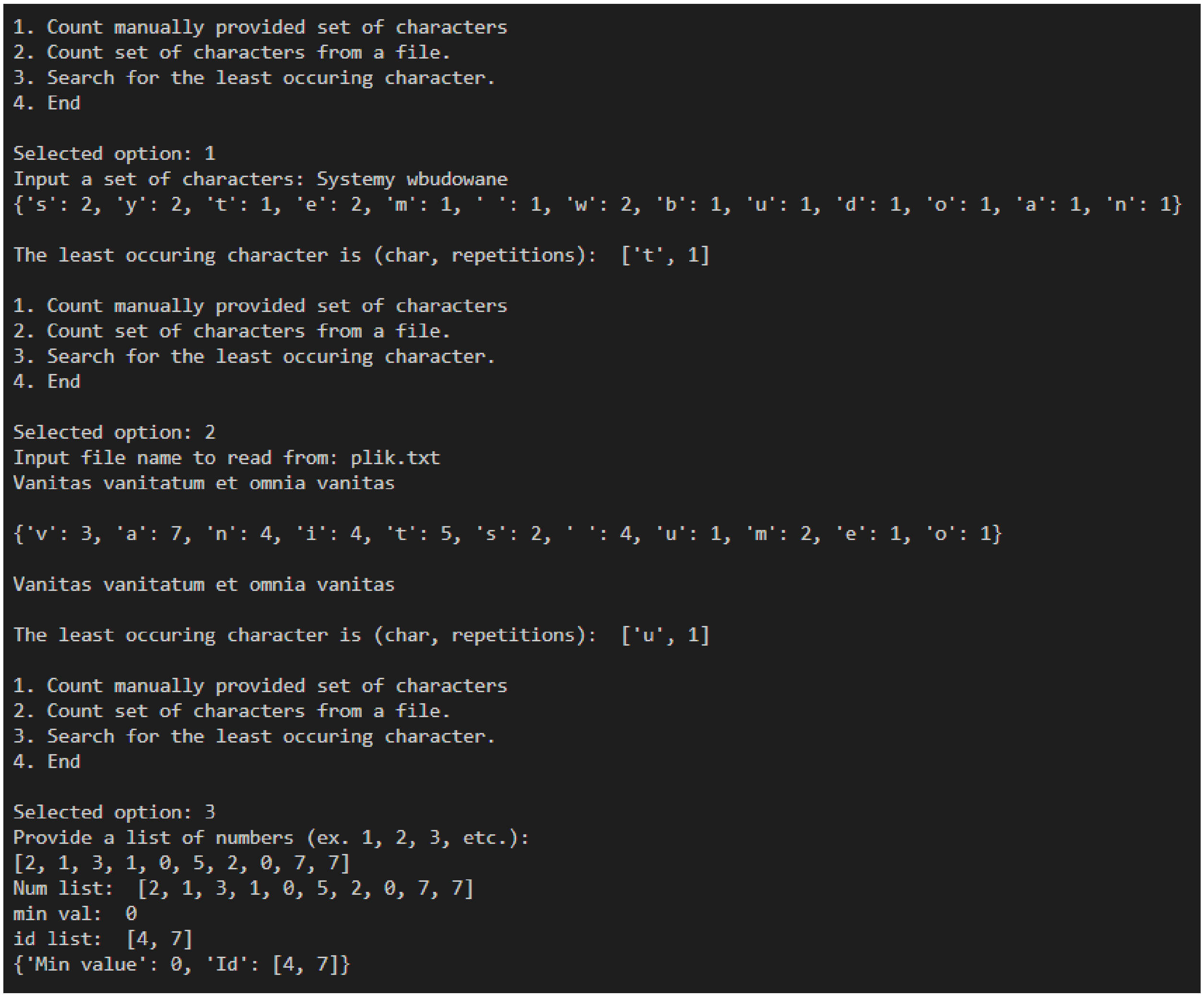


```python
from os import read


def char_counter(chars):
    repetition_dict = {}
    for char in chars.lower():
        if char in repetition_dict.keys():
            repetition_dict[char] += 1
        else:
            repetition_dict[char] = 1
    return repetition_dict


def file_char_counter(file_name: str) -> dict:
    try:
        with open(file_name, "r") as rf:
            content = rf.read()
        print(content, "\n")
        return char_counter(content)
    except IOError:
        print("No file with provided name accessible\n")


def least_occuring(dictionary):
    return [
        min(dictionary, key=dictionary.get),
        dictionary[min(dictionary, key=dictionary.get)],
    ]


def list_min_value(num_list: list) -> dict:
    # input: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
    num_list = [int(each) for each in num_list]
    min_value = min(num_list)
    id_list = [each for each in range(len(num_list)) if num_list[each] == min_value]
    print("Num list: ", num_list)
    print("min val: ", min_value)
    print("id list: ", id_list)
    return {"Min value": min_value, "Id": id_list}

```



```python

while True:
    print(
        "\n1. Count manually provided set of characters \n2. Count set of characters from a file.\n3. Search for the least occuring character.\n4. End\n"
    )
    menu = input("Selected option: ")
    if menu == "1":
        chars = input("Input a set of characters: ")
        print(char_counter(chars), "\n")
        print("The least occuring character is (char, repetitions): ", least_occuring(char_counter(chars)))
    elif menu == "2":
        read_file = input("Input file name to read from: ")
        print(file_char_counter(read_file), "\n")
        print(
            "The least occuring character is (char, repetitions): ",
            least_occuring(file_char_counter(read_file)),
        )
    elif menu == "3":
        mstring = input("Provide a list of numbers (ex. 1, 2, 3, etc.):\n")
        string = mstring.strip("[").strip("]")
        num_list = string.split(", ")
        print(list_min_value(num_list))
    else:
        print("\nGoodbye\n")
        break
```

5. Wyniki



```
1. Count manually provided set of characters
2. Count set of characters from a file.
3. Search for the least occuring character.
4. End

Selected option: 1
Input a set of characters: Systemy wbudowane
{'s': 2, 'y': 2, 't': 1, 'e': 2, 'm': 1, ' ': 1, 'w': 2, 'b': 1, 'u': 1, 'd': 1, 'o': 1, 'a': 1, 'n': 1}

The least occuring character is (char, repetitions):  ['t', 1]

1. Count manually provided set of characters
2. Count set of characters from a file.
3. Search for the least occuring character.
4. End

Selected option: 2
Input file name to read from: plik.txt
Vanitas vanitatum et omnia vanitas

{'v': 3, 'a': 7, 'n': 4, 'i': 4, 't': 5, 's': 2, ' ': 4, 'u': 1, 'm': 2, 'e': 1, 'o': 1}

Vanitas vanitatum et omnia vanitas

The least occuring character is (char, repetitions):  ['u', 1]

1. Count manually provided set of characters
2. Count set of characters from a file.
3. Search for the least occuring character.
4. End

Selected option: 3
Provide a list of numbers (ex. 1, 2, 3, etc.):
[2, 1, 3, 1, 0, 5, 2, 0, 7, 7]
Num list:  [2, 1, 3, 1, 0, 5, 2, 0, 7, 7]
min val:  0
id list:  [4, 7]
{'Min value': 0, 'Id': [4, 7]}
```