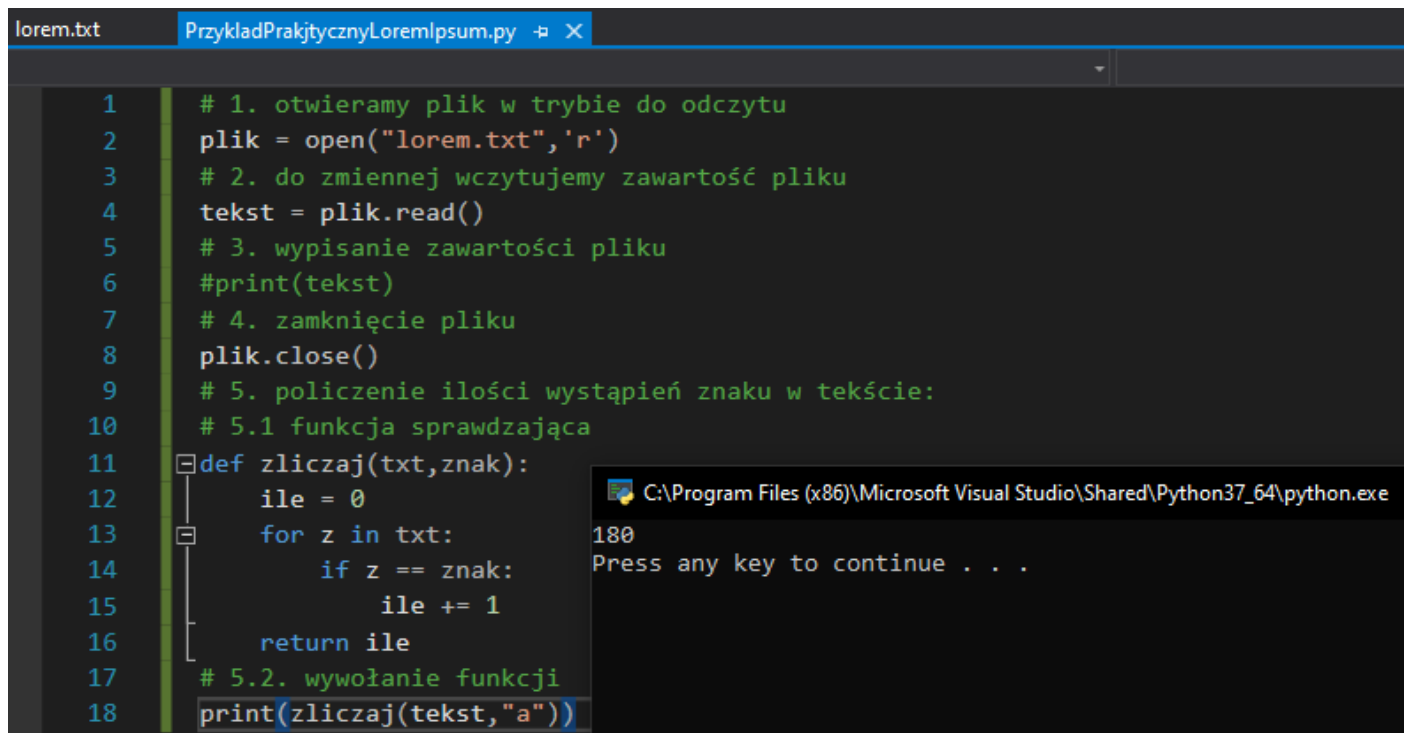


## Operacje na stringach – przykład praktyczny

Do tego ćwiczenia przyda się plik zawierający 5 akapitów z tekstem Lorem Ipsum. Plik nazwałem lorem.txt .

W projekcie dodajemy plik tekstowy.

Opis rozwiązania zagadnienia:

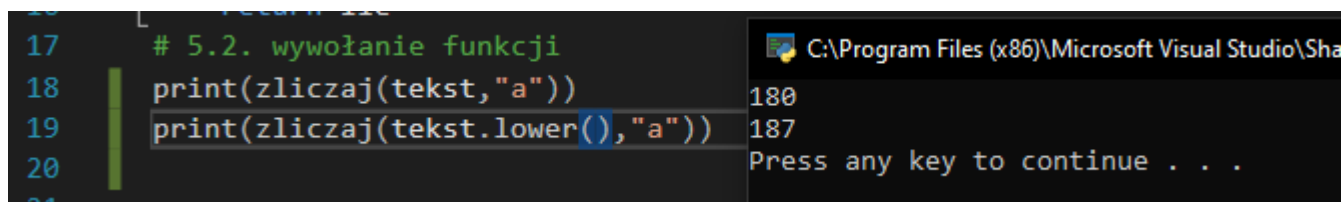


The screenshot shows a Visual Studio editor with two tabs: 'lorem.txt' and 'PrzykładPraktycznyLoremIpsum.py'. The Python script in the second tab contains the following code:

```
1 # 1. otwieramy plik w trybie do odczytu
2 plik = open("lorem.txt",'r')
3 # 2. do zmiennej wczytujemy zawartość pliku
4 tekst = plik.read()
5 # 3. wypisanie zawartości pliku
6 #print(tekst)
7 # 4. zamknięcie pliku
8 plik.close()
9 # 5. policzenie ilości wystąpień znaku w tekście:
10 # 5.1 funkcja sprawdzająca
11 def zliczaj(txt,znak):
12     ile = 0
13     for z in txt:
14         if z == znak:
15             ile += 1
16     return ile
17 # 5.2. wywołanie funkcji
18 print(zliczaj(tekst,"a"))
```

The output window on the right shows the command prompt running 'C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37\_64\python.exe' and displaying the result '180' followed by 'Press any key to continue . . . '.

Powyższy sposób nie uwzględnia dużych i małych liter. Można łatwo zliczyć wszystkie litery zamieniając najpierw tekst na małe litery:

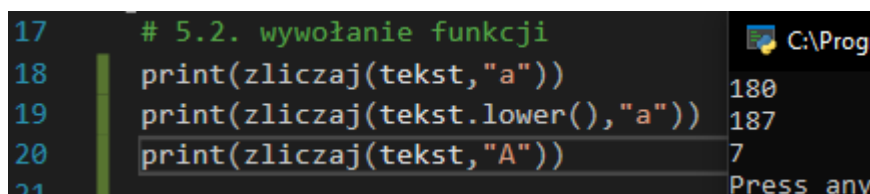


This screenshot shows the same Python script as before, but with an additional line of code added at line 19:

```
17 # 5.2. wywołanie funkcji
18 print(zliczaj(tekst,"a"))
19 print(zliczaj(tekst.lower(),"a"))
20
```

The output window now shows two results: '180' for the first call and '187' for the second call, followed by 'Press any key to continue . . . '.

Chcąc wyliczyć tylko duże litery musimy zamias „a” wpisać „A”:



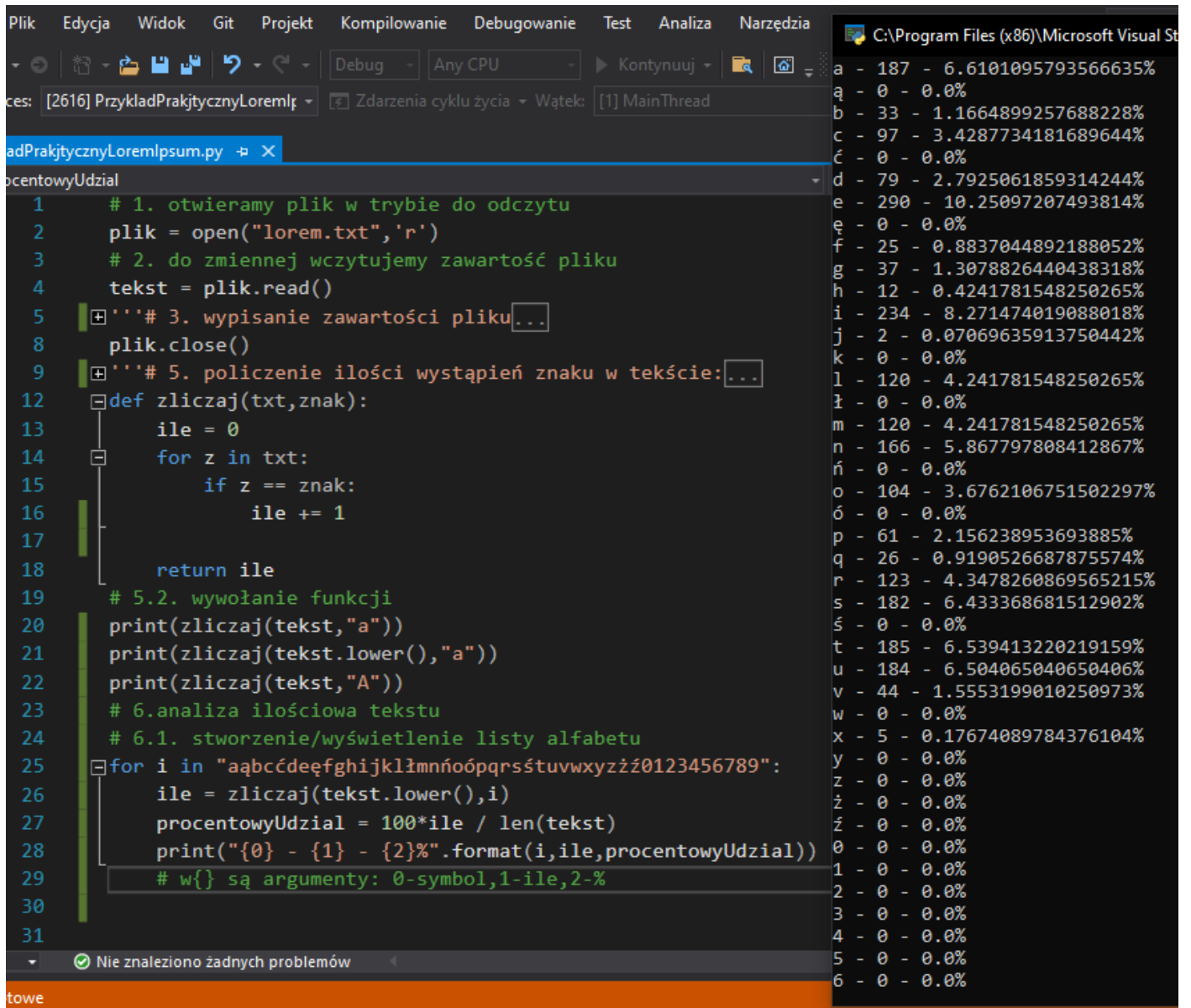
This screenshot shows the script with a third line of code added at line 20:

```
17 # 5.2. wywołanie funkcji
18 print(zliczaj(tekst,"a"))
19 print(zliczaj(tekst.lower(),"a"))
20 print(zliczaj(tekst,"A"))
21
```

The output window shows three results: '180' for lowercase 'a', '187' for lowercase 'a' (from the lowercased text), and '7' for uppercase 'A', followed by 'Press any key to continue . . . '.

Rachunek się zgadza.

Analiza tekstu pod kątem ilościowym:

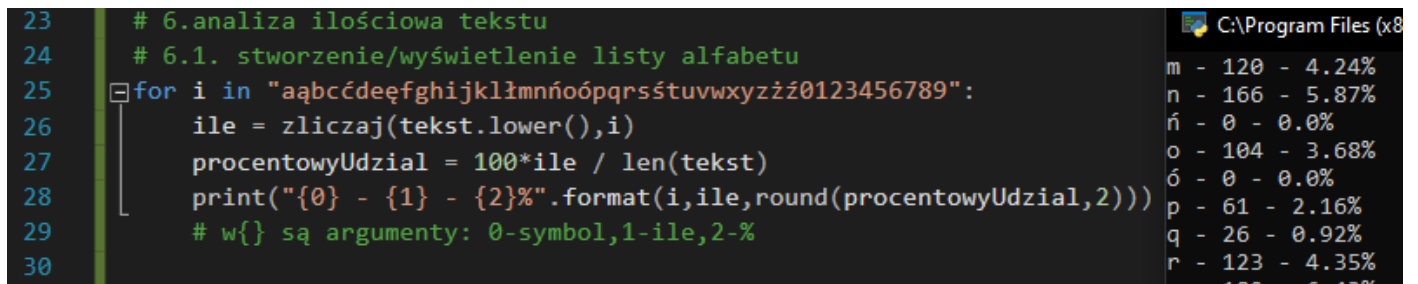


```
Plik  Edycja  Widok  Git  Projekt  Kompilowanie  Debugowanie  Test  Analiza  Narzędzia
[2616] PrzykładPraktycznyLoremIpsum.py  Debug  Any CPU  Kontynuuj
Zdarzenia cyklu życia  Wątek: [1] MainThread

PrzykładPraktycznyLoremIpsum.py  X
procentowyUdzial
1  # 1. otwieramy plik w trybie do odczytu
2  plik = open("lorem.txt",'r')
3  # 2. do zmiennej wczytujemy zawartość pliku
4  tekst = plik.read()
5  '''# 3. wypisanie zawartości pliku...
8  plik.close()
9  '''# 5. policzenie ilości wystąpień znaku w tekście:...
12 def zliczaj(txt,znak):
13     ile = 0
14     for z in txt:
15         if z == znak:
16             ile += 1
17
18     return ile
19
20 # 5.2. wywołanie funkcji
21 print(zliczaj(tekst,"a"))
22 print(zliczaj(tekst.lower(),"a"))
23 print(zliczaj(tekst,"A"))
24 # 6.analiza ilościowa tekstu
25 # 6.1. stworzenie/wyświetlenie listy alfabetu
26 for i in "aąbcćdeęfghijklłmnńoópqrsśtuvwxyzż0123456789":
27     ile = zliczaj(tekst.lower(),i)
28     procentowyUdzial = 100*ile / len(tekst)
29     print("{0} - {1} - {2}%".format(i,ile,procentowyUdzial))
30     # w{} są argumenty: 0-symbol,1-ile,2-%
31
Nie znaleziono żadnych problemów
towe
```

Symbol	Ilość	Procentowy udział
a	187	6.6101095793566635%
ą	0	0.0%
b	33	1.1664899257688228%
c	97	3.4287734181689644%
ć	0	0.0%
d	79	2.7925061859314244%
e	290	10.25097207493814%
ę	0	0.0%
f	25	0.8837044892188052%
g	37	1.3078826440438318%
h	12	0.4241781548250265%
i	234	8.271474019088018%
j	2	0.07069635913750442%
k	0	0.0%
l	120	4.241781548250265%
ł	0	0.0%
m	120	4.241781548250265%
n	166	5.867797808412867%
ń	0	0.0%
o	104	3.6762106751502297%
ó	0	0.0%
p	61	2.156238953693885%
q	26	0.9190526687875574%
r	123	4.3478260869565215%
s	182	6.433368681512902%
ś	0	0.0%
t	185	6.539413220219159%
u	184	6.504065040650406%
v	44	1.5553199010250973%
w	0	0.0%
x	5	0.17674089784376104%
y	0	0.0%
z	0	0.0%
ż	0	0.0%
ź	0	0.0%
0	0	0.0%
1	0	0.0%
2	0	0.0%
3	0	0.0%
4	0	0.0%
5	0	0.0%
6	0	0.0%

Wprowadźmy jeszcze zaokrąglenie wartości procentowych do 2 miejsc po przecinku:



```
23  # 6.analiza ilościowa tekstu
24  # 6.1. stworzenie/wyświetlenie listy alfabetu
25  for i in "aąbcćdeęfghijklłmnńoópqrsśtuvwxyzż0123456789":
26      ile = zliczaj(tekst.lower(),i)
27      procentowyUdzial = 100*ile / len(tekst)
28      print("{0} - {1} - {2}%".format(i,ile,round(procentowyUdzial,2)))
29      # w{} są argumenty: 0-symbol,1-ile,2-%
30
```

Symbol	Ilość	Procentowy udział
m	120	4.24%
n	166	5.87%
ń	0	0.0%
o	104	3.68%
ó	0	0.0%
p	61	2.16%
q	26	0.92%
r	123	4.35%
s	182	6.43%