

Krzysztof Ryś	Nr albumu: 73765	14.11.2025 r
Cyberbezpieczeństwo	Wstęp do programowania	Semestr: pierwszy
Grupa: L6	Instrukcja warunkowa if elif else, praca z plikami	Laboratoria nr2

CEL ZADANÍ:

Celem zadań było przećwiczenie podstawowych konstrukcji języka Python, takich jak:

- instrukcje warunkowe if/elif/else,
- operacje na łańcuchach znaków,
- operacje wejścia–wyjścia (plik),
- proste algorytmy porządkowania danych,
- metody wbudowane dla typów danych,
- wykorzystanie operatorów wycinania „slicing”.

Zadanie 1

Program pobiera liczbę punktów od użytkownika i na podstawie warunków decyduje o wyniku egzaminu.

```

punkty = float(input("Podaj punkty: "))
if punkty > 80:
    print("Zaliczony w 0 terminie")
elif punkty <= 80 and punkty >= 50:
    print("Możesz poprawić jego wynik")
else:
    print("Musisz poprawić")

zadanie1 ×
:
C:\Users\KoxOper\AppData\Local\Programs\Python\Python313\python.exe
Podaj punkty: 4
Musisz poprawić

Process finished with exit code 0

```

```
C:\Users\KoxOper\AppData\Local\Programs\Python\Python313\python.exe
Podaj punkty: 54
Możesz poprawić jego wynik
```

```
Process finished with exit code 0
```

```
C:\Users\KoxOper\AppData\Local\Programs\Python\Python313\python.exe
Podaj punkty: 89
Zaliczony w 0 terminie
```

```
Process finished with exit code 0
```

Działanie programu:

- Jeśli punkty > 80 → egzamin zaliczony w 1 terminie.
- Jeśli punkty są z zakresu 50–80 → można poprawić wynik.
- Jeśli punkty < 50 → obowiązkowa poprawa.

Program korzysta z instrukcji warunkowych oraz porównywania wartości liczbowych.

Zadanie 2

Program porządkuje trzy liczby od najmniejszej do największej bez użycia funkcji wbudowanych.

```
x=float(input("podaj X:"))
y=float(input("podaj y:"))
z=float(input("podaj z:"))

if x<y<z:
    print(x,y,z)
elif x<z<y:
    print(x,z,y)
elif y<x<z:
    print(y,x,z)
elif y<z<x:
    print(y,z,x)
elif z<x<y:
    print(z,x,y)
elif z<y<x:
    print(z,y,x)

# Sortowanie bombelkowe:
# if x > y:
#     x, y = y, x
# if y > z:
#     y, z = z, y
# if x > y:
#     x, y = y, x
#
# print(x, y, z)
```

 zadanie2 ×

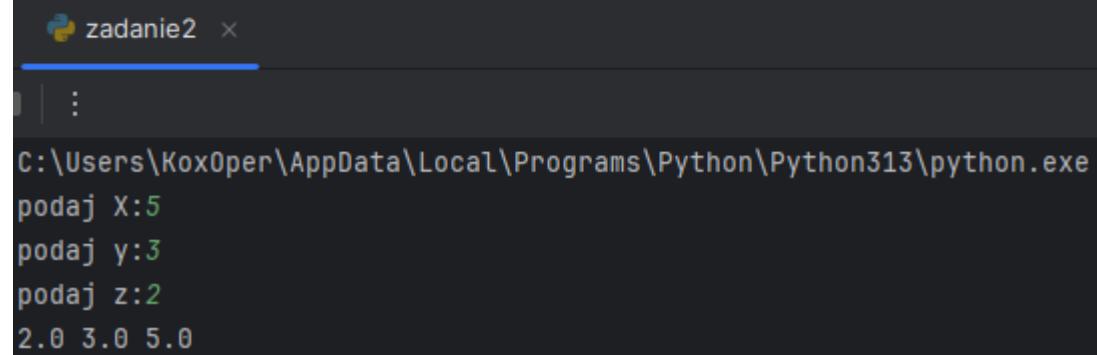
```
:
C:\Users\KoxOper\AppData\Local\Programs\Python\Python313\python.exe
podaj X:5
podaj y:3
podaj z:2
2.0 3.0 5.0
```

Zrobiłem to zadanie na dwa sposoby, pierwszy sposób opierał się na wykorzystaniu liczby kombinacji ustawień liczb od najmniejszej do największej, drugi ukazany poniżej przedstawia uproszczony algorytm sortowania bąbelkowego. Porównywane są pary liczb i zamieniane miejscami, jeżeli są w złej kolejności.

```
x=float(input("podaj x:"))
y=float(input("podaj y:"))
z=float(input("podaj z:"))

...
if x > y:
    x, y = y, x
if y > z:
    y, z = z, y
if x > y:
    x, y = y, x

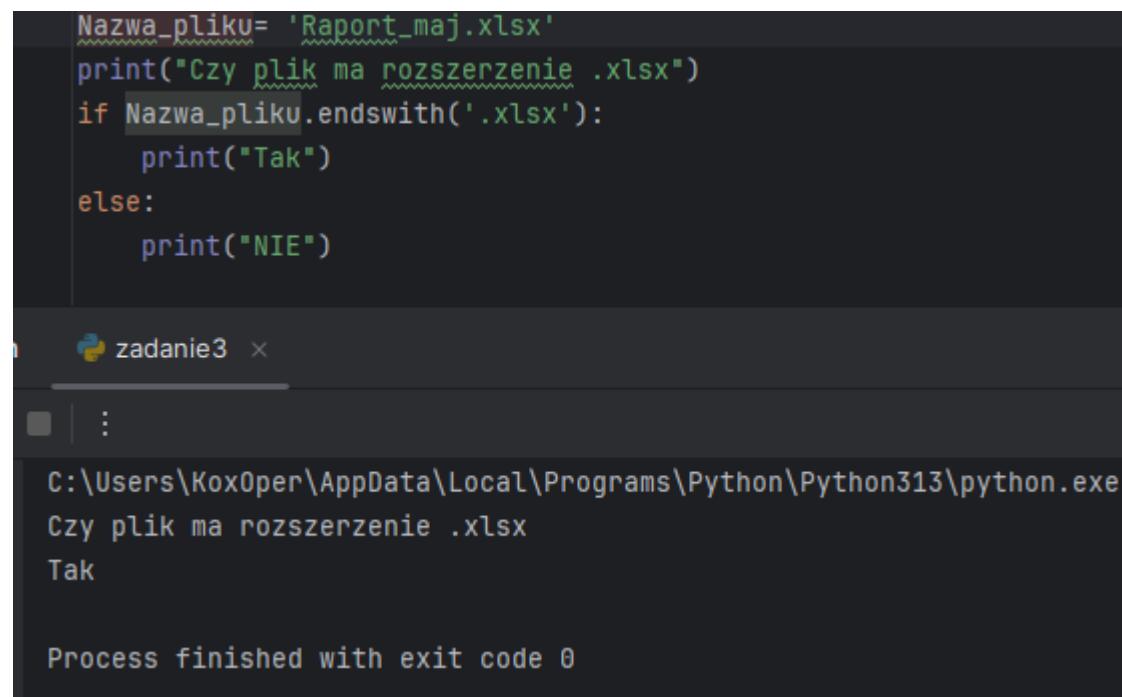
print(x, y, z)
```



The screenshot shows a code editor with a Python script named 'zadanie2.py'. The script contains a program to sort three numbers input by the user. Below the editor is a terminal window titled 'zadanie2' showing the execution of the script and its output. The terminal output shows the script being run from the command line, taking inputs '5', '3', and '2', and then printing the sorted values '2.0', '3.0', and '5.0'.

```
C:\Users\KoxOper\AppData\Local\Programs\Python\Python313\python.exe
zadanie2.py
podaj x:5
podaj y:3
podaj z:2
2.0 3.0 5.0
```

Zadanie 3 Program sprawdza, czy nazwa pliku kończy się rozszerzeniem .xlsx



```
Nazwa_pliku= 'Raport_maj.xlsx'
print("Czy plik ma rozszerzenie .xlsx")
if Nazwa_pliku.endswith('.xlsx'):
    print("Tak")
else:
    print("NIE")
```

zadanie3

```
C:\Users\KoxOper\AppData\Local\Programs\Python\Python313\python.exe
Czy plik ma rozszerzenie .xlsx
Tak

Process finished with exit code 0
```

Wykorzystana została metoda `endswith()`, która sprawdza zakończenie łańcucha znaków.

Zadanie 4

a)

```
1     gol = int(input("podaj liczbę goli:"))
2     bonus = 0
3     punkty = gol * 10
4     # A
5     if gol > 5 and gol < 10:
6         bonus += 5
7     elif gol > 10:
8         bonus += 10
9     punkty += bonus
10    print(f'Zdobyte punkty:{punkty} ')
11
12    # B
13    #
14    # if gol > 5:
15    #     bonus=bonus+5
16    # if gol > 10:
17    #     bonus=10+bonus
18    # punkty+=bonus
19    # print(f'Zdobyte punkty:{punkty} ')
```

Run zadanie4 ×

↻ | :

```
C:\Users\KoxOper\AppData\Local\Programs\Python\Python313\python.
podaj liczbę goli:4
Zdobyte punkty:40

Process finished with exit code 0
```

Opis działania (A):

- Każdy gol daje 10 punktów: punkty = gol * 10
- Jeśli drużyna zdobyła więcej niż 5 i mniej niż 10 goli, przyznawane jest 5 punktów bonusowych
- Jeśli zdobyła więcej niż 10 goli, dostaje tylko 10 punktów bonusowych
- Bonus nie jest sumowany — drużyna dostaje albo 5, albo 10, zależnie od poziomu.

Wynik:

- Program oblicza sumę punktów i wypisuje wynik w konsoli

b)

```
gol = int(input("podaj liczbe goli:"))
bonus = 0
punkty = gol * 10
# A
# if gol > 5 and gol < 10:
#     bouns +=5
# elif gol > 10:
#     bonus +=10
# punkty+=bonus
# print(f'Zdobyte punkty:{punkty}')

# B
#
if gol > 5:
    bonus=bonus+5
if gol > 10:
    bonus=10+bonus
punkty+=bonus
print(f'Zdobyte punkty:{punkty}')
```

```
n  zadanie4  ×
:
C:\Users\KoxOper\AppData\Local\Programs\Python\3.8\python.exe -u "C:/Users/KoxOper/Desktop/zadanie4.py"
podaj liczbe goli:7
Zdobyte punkty:75

Process finished with exit code 0
```

Opis działania (B):

- Każdy gol daje 10 punktów
- Po przekroczeniu 5 goli dodawane jest +5 bonusu.
- Po przekroczeniu 10 goli dodawane jest kolejne +10
- Bonusy sumują się, więc np. przy 12 golach → bonus = 5 + 10 = 15

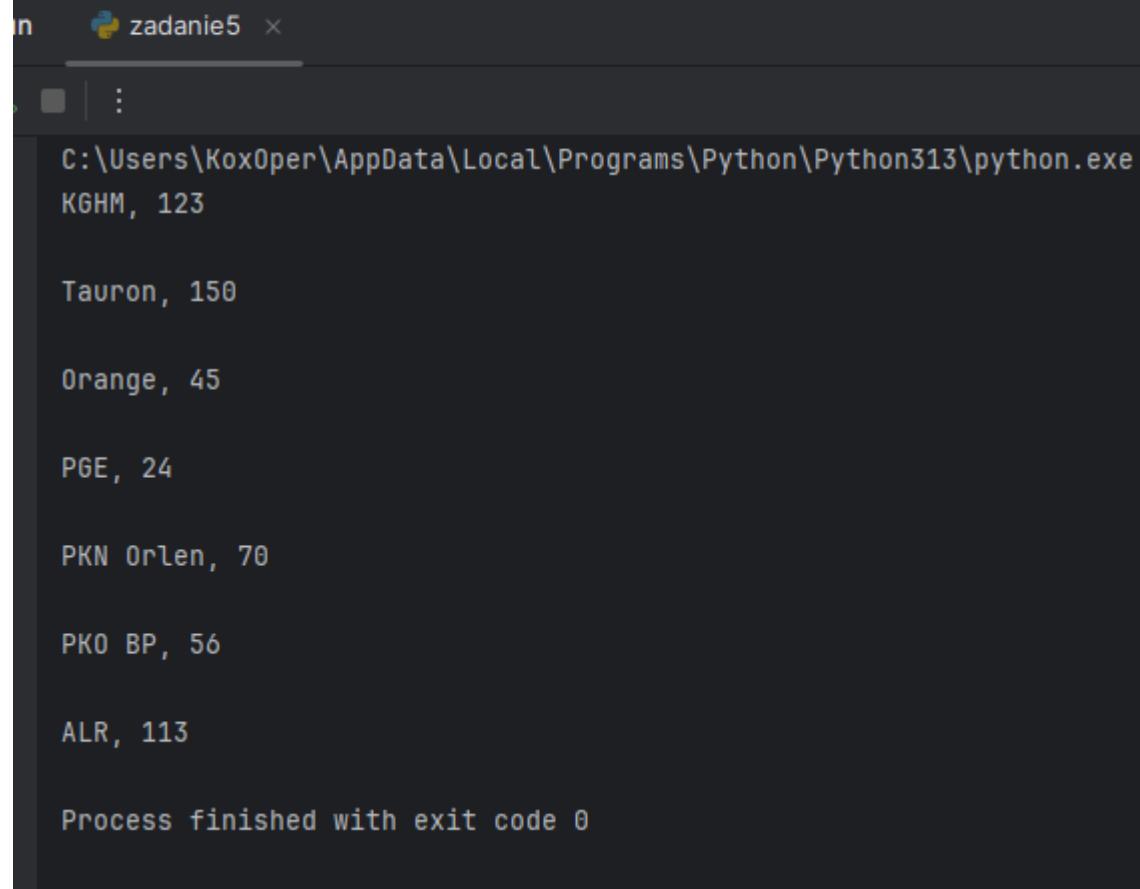
Wynik:

- Program wylicza całkowity wynik (punkty z goli + suma bonusów) i drukuje go do konsoli

Zadanie 5

Program dopisuje nową linię do pliku oraz ponownie wyświetla całą jego zawartość.

```
with open("notowania_gieldowe.txt", "a") as plik:  
    plik.write("\nALR, 113")  
  
with open("notowania_gieldowe.txt", "r") as plik:  
    for linia in plik:  
        print(linia)
```



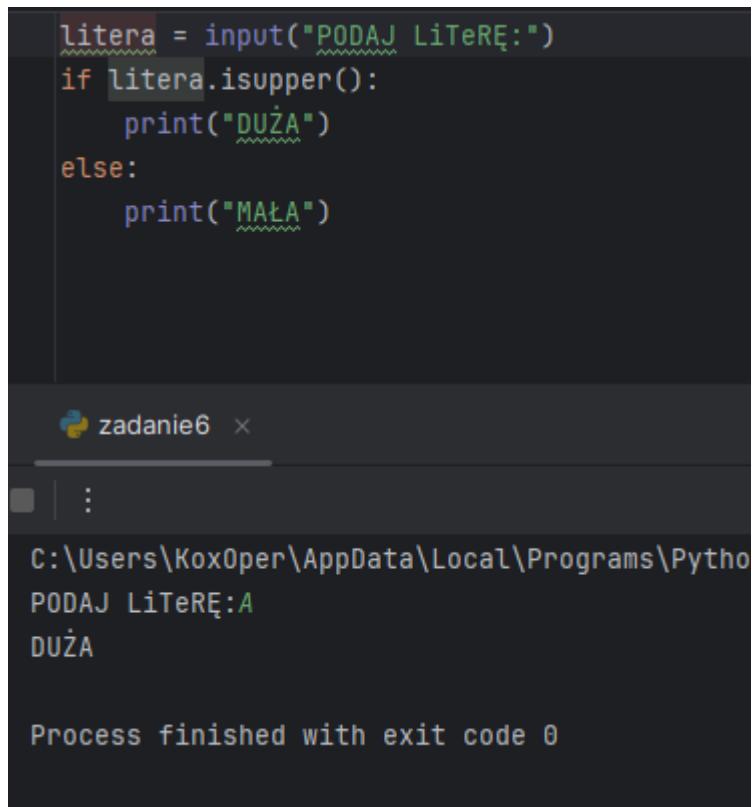
The screenshot shows a code editor window with a Python script named 'zadanie5.py'. The script contains two parts: writing a new line to a file and then reading it back. The output in the terminal shows the original content followed by the new line 'ALR, 113'.

```
zadanie5  
...  
C:\Users\KoxOper\AppData\Local\Programs\Python\Python313\python.exe  
KGHM, 123  
  
Tauron, 150  
  
Orange, 45  
  
PGE, 24  
  
PKN Orlen, 70  
  
PKO BP, 56  
  
ALR, 113  
  
Process finished with exit code 0
```

Użyłem tutaj dwóch atrybutów, a – do dopisywania do pliku oraz r – do odczytu. \n – oznacza kolejną linie.

Zadanie 6

Program sprawdza, czy podana litera jest duża czy mała:



```
litera = input("PODAJ LiTeRE:")  
if litera.isupper():  
    print("DUŻA")  
else:  
    print("MAŁA")
```

zadanie6

PODAJ LiTeRE:
DUŻA

Process finished with exit code 0

Metoda `isupper()` sprawdza, czy litera jest zapisana wielką literą.

Zadanie 7 Program sprawdza poprawność hasła:

```
Hasło = 'pk47!jy0893'

if len(Hasło)>=11 and Hasło.find('!'):
    print("Hasło jest poprawne")
else:
    print("Hasło jest nie poprawne")
```

zadanie7

C:\Users\KoxOper\AppData\Local\Programs\Python
Hasło jest poprawne

Process finished with exit code 0

Program sprawdza czy długość hasła wynosi co najmniej 11 znaków, oraz czy zawiera znak !.

Zadanie 8 Wyodrębnianie fragmentów tekstu poprzez slicing:

```
text = 'Studiuje-Informatykę'
print(text[0:3])
print(text[-2:])
```

zadanie8

C:\Users\KoxOper\AppData\Local\Programs\Python
Stu
kę

Program wypisuje pierwsze 3 znaki oraz ostatnie 2 znaki.

Zadanie 9

Zamiana małych liter na duże i odwrotnie:

The screenshot shows a code editor window with a dark theme. At the top, there is a code snippet:

```
tekst = input("Podaj tekst: ")
print(tekst.swapcase())
```

Below the code, the window title bar says "zadanie9". In the main pane, the output of the script is displayed:

```
C:\Users\KoxOper\AppData\Local\Programs\Python\Python38-32\zadanie9.py
Podaj tekst: OhnoMordini
oHNOmORDINI

Process finished with exit code 0
```

Wykorzystałem metodę swapcase(), która zamienia wielkość liter w całym ciągu.

Wnioski: Podczas pisania kodu utrwaliałem działania instrukcji warunkowych, obsługi plików w różnych trybach, wykorzystałem metody wbudowane dla napisów takie jak: endswith, isupper, swapcase, find. Przeanalizowałem jak tworzyć algorytmy sortowania. Poznałem sposób wyodrębniania fragmentów tekstu oraz łączenia warunków oraz stosowania operatorów logicznych.

Zadanie pozwoliły mi na zrozumienie podstaw języka Python.