

Imie	Szymon
Nazwisko	Madeja
Kierunek	Cyberbezpieczeństwo
Grupa	L6
Semestr	Pierwszy
Nr albumu	74178
Nr laboratorium	LAB2
Tematyka	Instrukcja warunkowa if elif else, praca z plikami

## SPIS TREŚCI

Zadanie 1 - Program obliczający wyniki egzaminu dla studentów

Zadanie 2 - Program sortowania trzech liczb

Zadanie 3 - Sprawdzanie rozszerzenia pliku

Zadanie 4a - Obliczanie wyniku drużyny piłkarskiej

Zadanie 4b - Obliczanie wyniku drużyny piłkarskiej

Zadanie 5a - Odczyt danych z pliku tekstowego

Zadanie 5b - Dopisywanie danych do pliku tekstowego

Zadanie 6 - Sprawdzanie wielkości litery

Zadanie 7 - Sprawdzanie poprawności hasła

Zadanie 8 - Wycinanie znaków z tekstu

Zadanie 9 - Zamiana wielkości liter w tekście

# ZADANIE 1

## CEL ZADANIA:

Stworzenie programu sprawdzającego, czy wynik egzaminu pozwala na zaliczenie w terminie, wymaga poprawki czy jest niewystarczający.

## KOD PROGRAMU:

```
1 wynik = int(input("Podaj wynik egzaminu (ile procent): "))
2
3 if wynik >= 80:
4     print("Zdano egzamin w terminie 0")
5 elif wynik >= 50 and wynik <=80:
6     print("Możesz poprawić wynik egzaminu")
7 else:
8     print("Musisz poprawić wynik egzaminu")
```

## WYNIK PROGRAMU:

```
Podaj wynik egzaminu (ile procent): 10
Musisz poprawić wynik egzaminu
Podaj wynik egzaminu (ile procent): 51
Możesz poprawić wynik egzaminu
Podaj wynik egzaminu (ile procent): 100
Zdano egzamin w terminie 0
```

## ANALIZA KODU:

Do zmiennej „**wynik**” przypisujemy input, który może być tylko liczbą całkowitą oraz w którym użytkownik musi podać liczbę procent uzyskanych z egzaminu.

Później w instrukcji warunkowej **if** porównujemy wpisaną wartość do podanych przedziałów w zadaniu oraz program wypisuje odpowiednią wiadomość.

## WNIOSKI:

Program poprawnie pobrał i porównał wpisany przez użytkownika wynik egzaminu oraz wypisał odpowiednią wiadomość.

# ZADANIE 2

## CEL ZADANIA:

Stworzenie programu, który sortuje 3 podane liczby przez użytkownika od najmniejszej do największej.

## KOD PROGRAMU:

```
1  x = float(input("Podaj liczbę x: "))
2  y = float(input("Podaj liczbę y: "))
3  z = float(input("Podaj liczbę z: "))
4
5  lista = []
6
7  if x <= y <= z:
8      lista.append(x)
9      lista.append(y)
10     lista.append(z)
11     print(lista)
12 elif x <= z <= y:
13     lista.append(x)
14     lista.append(z)
15     lista.append(y)
16     print(lista)
17 elif y <= x <= z:
18     lista.append(y)
19     lista.append(x)
20     lista.append(z)
21     print(lista)
22 elif y <= z <= x:
23     lista.append(y)
24     lista.append(z)
25     lista.append(x)
26     print(lista)
27 elif z <= x <= y:
28     lista.append(z)
29     lista.append(x)
30     lista.append(y)
31     print(lista)
32 else:
33     lista.append(y)
34     lista.append(x)
35     lista.append(z)
36     print(lista)
```

## WYNIK PROGRAMU:

```
Podaj liczbę x: 1
Podaj liczbę y: 565
Podaj liczbę z: 21
[1.0, 21.0, 565.0]
```

## ANALIZA KODU:

Do zmiennych „x”, „y”, „z” użytkownik przypisuje 3 liczby typu float(), które będą zapisane w postaci liczby zmiennoprzecinkowej.

W instrukcji warunkowej **if** sprawdzamy podane warunki i w zależności od wyniku program wypisuje odpowiednio utworzoną **listę** liczb, które są posortowane od najmniejszej do największej.

W instrukcji warunkowej wykorzystaliśmy ostatnio poznany operator logiczny:

`<=` - mniejsze lub równe,

oraz poznaliśmy metodę „**append(argument)**”, która może zostać użyta na liście i odpowiada za dodanie elementu (argument) na końcu podanej listy.

## WNIOSKI:

Program poprawnie pobrał, odczytał i porównał wszystkie podane liczby przez użytkownika oraz odpowiednio je posortował.

# ZADANIE 3

## CEL ZADANIA:

Stworzenie programu sprawdzającego, czy końcówka tekstu w zmiennej to „.xlsx”

### KOD PROGRAMU:

```
1 nazwa_pliku = 'Raport_maj.xlsx'
2
3 rozszerzenie_pliku = nazwa_pliku.endswith(".xlsx")
4
5 if rozszerzenie_pliku == True:
6     print("Tak")
7 else:
8     print("Nie")
```

### WYNIK PROGRAMU:

```
PS C:\Users\fil53\Desktop
  atoria/Laboratoria 2 - 25.
  a/zadanie_3.py"
Tak
```

### ANALIZA KODU:

Zmienna „nazwa\_pliku” przechowuje tekst podany w treści zadania.

Następnie w instrukcji warunkowej if sprawdzamy, czy końcówka tego tekstu to „.xlsx”, jeśli tak program wypisuje „tak”, a jeżeli nie to program wypisuje „nie”.

### WNIOSKI:

Program poprawnie pobrał i sprawdził, czy tekst posiada podana końcówkę oraz wypisał na tej podstawie odpowiednią odpowiedź.

## ZADANIE 4a

### CEL ZADANIA:

Stworzenie programu obliczającego ilość punktów strzelonych przez drużynę z uwzględnieniem bonusów.

### KOD PROGRAMU:

```
1 gol = int(input("Liczba goli: "))
2 bonus = int()
3
4 punkty_za_gole = gol * 10
5
6 if gol > 10:
7     bonus = bonus + 10
8     wynik = punkty_za_gole + bonus
9     print(f'Wynik drużyny wynosi: {wynik}')
10 elif gol > 5:
11     bonus = bonus + 5
12     wynik = punkty_za_gole + bonus
13     print(f'Wynik drużyny wynosi: {wynik}')
14 else:
15     wynik = punkty_za_gole
16     print(f'Wynik drużyny wynosi: {wynik}')
```

### WYNIK PROGRAMU:

Liczba goli: 1	Liczba goli: 6	Liczba goli: 11
Wynik drużyny wynosi: 10	Wynik drużyny wynosi: 65	Wynik drużyny wynosi: 120

### ANALIZA KODU:

Do zmiennej "gol" przypisujemy input, który może być tylko liczbą całkowitą oraz w którym użytkownik musi podać liczbę goli strzelonych przez drużynę.

Później w instrukcji warunkowej **if** porównujemy wpisaną wartość do podanych przedziałów za które można otrzymać bonus w zadaniu oraz program wypisuje odpowiednią wiadomość.

### WNIOSKI:

Program poprawnie pobrał i porównał wpisaną przez użytkownika liczbę goli, obliczył oraz wypisał odpowiednią wiadomość.

## ZADANIE 4b

### CEL ZADANIA:

Stworzenie programu obliczającego ilość punktów strzelonych przez drużynę z uwzględnieniem bonusów.

### KOD PROGRAMU:

```
1 gol = int(input("Liczba goli: "))
2 bonus = int()
3
4 punkty_za_gole = gol * 10
5
6 if gol > 10:
7     bonus = bonus + 10
8     wynik = punkty_za_gole + bonus
9     print(f'Wynik drużyny wynosi: {wynik}')
10 elif gol > 5:
11     bonus = bonus + 5
12     wynik = punkty_za_gole + bonus
13     print(f'Wynik drużyny wynosi: {wynik}')
14 else:
15     wynik = punkty_za_gole
16     print(f'Wynik drużyny wynosi: {wynik}')
```

### WYNIK PROGRAMU:

Liczba goli: 1	Liczba goli: 6	Liczba goli: 12
Wynik drużyny wynosi: 10	Wynik drużyny wynosi: 65	Wynik drużyny wynosi: 130

### ANALIZA KODU:

Do zmiennej "gol" przypisujemy input, który może być tylko liczbą całkowitą oraz w którym użytkownik musi podać liczbę goli strzelonych przez drużynę.

Później w instrukcji warunkowej if porównujemy wpisaną wartość do podanych przedziałów za które można otrzymać bonus w zadaniu oraz program wypisuje odpowiednią wiadomość.

### WNIOSKI:

Program poprawnie pobrał i porównał wpisaną przez użytkownika liczbę goli, obliczył oraz wypisał odpowiednią wiadomość.

## ZADANIE 5a

### CEL ZADANIA:

Stworzenie programu odczytującego plik

### KOD PROGRAMU:

```
1 file = open("notowania_gieldowe.txt","r")
2
3 with file as plik:
4     print(plik.read())
```

### WYNIK PROGRAMU:

```
PS C:\Users\fil53\Desktop\Laborat
danie_5a.py
KGHM, 123
Tauron, 150
Orange, 45
PGE, 24
PKN Orlen, 70
PKO BP, 56
```

### ANALIZA KODU:

Program otwiera plik tekstowy o nazwie „notowania\_gieldowe.txt” w trybie odczytu

Zmienna „file” przechowuje obiekt pliku, który następnie zostaje użyty w instrukcji „with”.

Dzięki temu po zakończeniu odczytu plik zostaje automatycznie zamknięty.

Instrukcja „plik.read()” wczytuje całą zawartość pliku i wypisuje ją na ekranie za pomocą funkcji „print()”.

### WNIOSKI:

Program poprawnie odczytuje zawartość pliku tekstowego i wyświetla dane.

Zastosowanie instrukcji „with” jest zalecanym sposobem pracy z plikami, ponieważ automatycznie zamyka plik po zakończeniu na nim pracy.

## ZADANIE 5b

### CEL ZADANIA:

Stworzenie programu sprawdzającego, czy wynik egzaminu pozwala na zaliczenie w terminie, wymaga poprawki czy jest niewystarczający.



## KOD PROGRAMU:

```
1 with open("notowania_gieldowe.txt", "r") as plik:
2     print("PRZED ZMIANĄ:")
3     print(plik.read())
4
5 with open("notowania_gieldowe.txt", "a") as plik:
6     plik.write("ALR, 113")
7
8 with open("notowania_gieldowe.txt", "r") as plik:
9     print("PO ZMIANIE")
10    print(plik.read())
```

## WYNIK PROGRAMU:

```
PRZED ZMIANĄ:
KGHM, 123
Tauron, 150
Orange, 45
PGE, 24
PKN Orlen, 70
PKO BP, 56

PO ZMIANIE
KGHM, 123
Tauron, 150
Orange, 45
PGE, 24
PKN Orlen, 70
PKO BP, 56
ALR, 113
```

## ANALIZA KODU:

Kod trzykrotnie otwiera plik „notowania\_gieldowe.txt”: najpierw odczytuje i wyświetla jego pierwotną zawartość (tryb „r” – tylko do odczytu), a następnie w trybie dopisywania (tryb „a”) dodaje nowy wiersz „ALR, 113”. Końcowo, plik jest otwierany ponownie, aby wyświetlić nową zawartość, która posiada już nowo dodany tekst.

## WNIOSKI:

Wynik potwierdza, że tryb „a” modyfikuje plik poprzez dodanie danych na końcu, co jest widoczne w ostatecznej wersji pliku. Konstrukcja „with open()” jest zalecanym sposobem pracy z plikami, ponieważ automatycznie zamyka plik po zakończeniu na nim pracy.

# ZADANIE 6

## CEL ZADANIA:

Stworzenie programu sprawdzającego, czy pierwsza litera podana przez użytkownika jest mała lub duża.

## KOD PROGRAMU:

```

1  litera = input("Wprowadź literę: ")
2
3  wielkosc = litera[0].isupper()
4  ✓ if wielkosc == True:
5      print("Pierwsza jest duża")
6  ✓ else:
7      print("Pierwsza litera jest mała")

```

### WYNIK PROGRAMU:

```

Wprowadź literę: A      Wprowadź literę: a
Pierwsza litera jest duża  Pierwsza litera jest mała

```

### ANALIZA KODU:

Do zmiennej "litera" przypisujemy input, który może być tylko tekstem oraz w którym użytkownik musi podać literę/słowo/zdanie.

Następnie w instrukcji warunkowej if program sprawdza, czy we wpisanej literze/słowie/zdaniu przez użytkownika pierwsza litera jest mała lub duża oraz wypisuje odpowiednią odpowiedź.

### WNIOSKI:

Program poprawnie pobrał i sprawdził czy w podanej literze/słowie/zdaniu pierwsza litera jest mała lub duża oraz wypisał odpowiednią odpowiedź.

## ZADANIE 7

### CEL ZADANIA:

Stworzenie programu sprawdzającego, czy podane hasło jest prawidłowe.

### KOD PROGRAMU:

```

1  haslo = 'pk47!jy0893'
2
3  dlugosc = len(haslo)
4
5
6  ✓ if '!' in haslo and dlugosc >= 11:
7      |     print(f"Podane hasło: {haslo}\nHasło jest poprawne")
8  ✓ else:
9      |     print(f"Podane hasło: {haslo}\nHasło jest niepoprawne")

```

### WYNIK PROGRAMU:

```

Podane hasło: pk47!jy0893  Podane hasło: pk47sjy0893
Hasło jest poprawne      Hasło jest niepoprawne

```

### ANALIZA KODU:

Do zmiennej „haslo” przypisujemy podane hasło w zadaniu.

Do zmiennej „dlugosc” przypisujemy dlugosc hasla uzywajac metody len()

Później w instrukcji warunkowej **if** sprawdzamy czy haslo posiada znak specjalny „!” używając do tego „in” oraz sprawdzamy czy hasło ma wymagane 11 znakow

### WNIOSKI:

Program poprawnie przypisał hasło do zmiennej oraz sprawdził czy jest ono odpowiednie zgodnie z podanymi instrukcjami, oraz wypisał odpowiednia odpowiedź.

## ZADANIE 8

### CEL ZADANIA:

Stworzenie programu sprawdzającego, który wytnie z podanego tekstu pierwsze i ostatnie litery

### KOD PROGRAMU:

```
1 text = 'Studiuje-Informatykę'
2 pierwsze = text[0] + text[1] + text[2]
3 ostatnie = text[-1] + text[-2]
4
5 print(f"Pierwsze trzy znaki: {pierwsze}")
6 print(f'Ostatnie dwa znaki: {ostatnie}")
```

### WYNIK PROGRAMU:

```
Szymon_Hudec_74178_CS_pra
Pierwsze trzy znaki: Stu
Ostatnie dwa znaki: ęk
```

### ANALIZA KODU:

Do zmiennej „**tekst**” przypisujemy tekst podany w zadaniu, natomiast później do zmiennych „**pierwsze**” i „**ostatnie**” przypisujemy litery.

Do zapisania tych liter w tym zadaniu zostało użyte „**zmienna[index]**”. Z podanej zmiennej pobieramy litery z odpowiednim index’em, indexy liczymy od 0 a jeżeli chcemy liczyć od końca to zapisujemy „-” przed liczbą

Później program wypisał tekst z użyciem zmiennych w których znajdowały się pobrane litery.

### WNIOSKI:

Program poprawnie pobrał, połączył i wypisał pobrane litery z tekstu.

## ZADANIE 9

### CEL ZADANIA:

Stworzenie programu zamieniającego wielkość liter.

### KOD PROGRAMU:

```

1  male = "TeN TeKsT jeST złożONy z MAŁYCH liter"
2  duze = "TeN TeKsT jeST złożONy z DUŻYCH liter"
3
4  print(f"PRZED ZMIANĄ: \nTekst 1: {male} \nTekst 2: {duze}")
5
6  male_new = male.swapcase()
7  duze_new = duze.swapcase()
8
9  print(f"PO ZMIANIE: \nTekst 1: {male_new} \nTekst 2: {duze_new}")

```

## WYNIK PROGRAMU:

```

PRZED ZMIANĄ:
Tekst 1: TeN TeKsT jeST złożONy z MAŁYCH liter
Tekst 2: TeN TeKsT jeST złożONy z DUŻYCH liter
PO ZMIANIE:
Tekst 1: tEn tEkSt jEst ZłożonY Z małych LITER
Tekst 2: tEn tEkSt jEst ZłożonY Z dużych LITER

```

## ANALIZA KODU:

Do zmiennej „**male**” i „**duze**” przypisujemy tekst, który później będziemy zmieniać.

Program wypisuje tekst przed zmianą a następnie używając metody „**swapcase()**” zmienia wielkość każdej litery z osobna w tekście. Gdy to zrobi wypisuje obydwa tekstu po zamianie.

## WNIOSKI:

Program poprawnie odczytał i zamienił wielkość liter w podanych tekstach oraz je wypisał.