



Kierunek: Informatyka

Rok akademicki: 2021/2022

Przedmiot: Python

Semestr: IV

Grupa dziekańska: WZ_ININ4_IS

Sprawozdanie z: Laboratorium numer 1

data: 30 marca 2022

Grupa robocza: GrupaRobocza05

Skład grupy:

Osoba 1: Filip Dworak

Osoba 2: Dariusz Wroniak

Osoba 3 : Grzegorz Żuromski

Osoba 4:

Aktywności realizowane przez członków grupy roboczej, czyli kto i za co był odpowiedzialny podczas laboratorium/zadania domowego oraz utworzenia sprawozdania (min 3 na osobę):

Osoba 1: Filip Dworak

Rozwiązanie zadań : C, 2, 3, 7, 8
Opracowanie sprawozdania

Osoba 2: Dariusz Wroniak

Rozwiązanie zadań : B, 1, 4, 5, 9, 10, 14

Osoba 3: Grzegorz Żuromski

Rozwiązanie zadań : A, D, 6, 6a, 11, 12 ,13

Osoba 4:

--

Sposoby komunikacji:

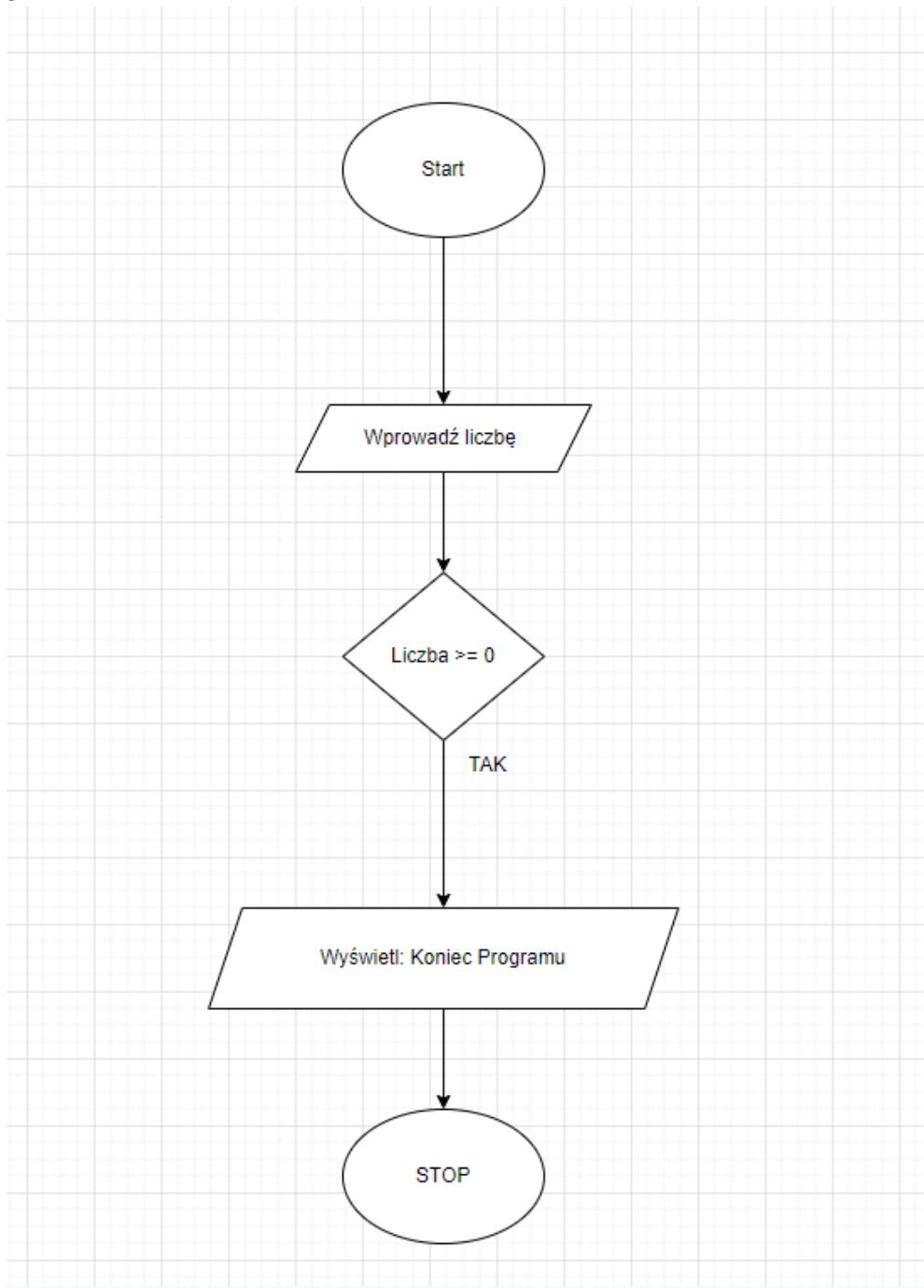
Komunikacja poprzez program Discord oraz Github

Zadanie A

Treść zadania

Utworzyć schemat blokowy algorytmu sprawdzający czy użytkownik podał liczbę większą od zera. W odpowiedzi program ma wyświetlić słowo "TAK" jeśli liczba jest większa od zera. Ostatnia instrukcja programu to wyświetlenie informacji "Koniec programu." Kolejno wprowadzić poprawkę umożliwiającą sprawdzenie czy podana wartość jest większa lub równa zero. Która instrukcja wykona się warunkowo a która zawsze? Z czego to wynika?

Rozwiązanie



Zadanie B

Treść zadania :

Utworzyć algorytm sprawdzający ile lat użytkownikowi zostało do stu lat. Program ma także podać informację czy użytkownik jest pełnoletni czy nie (wyświetlenie słowa "PEŁNOLETNI" "MŁODOCIANY")

Rozwiązanie :

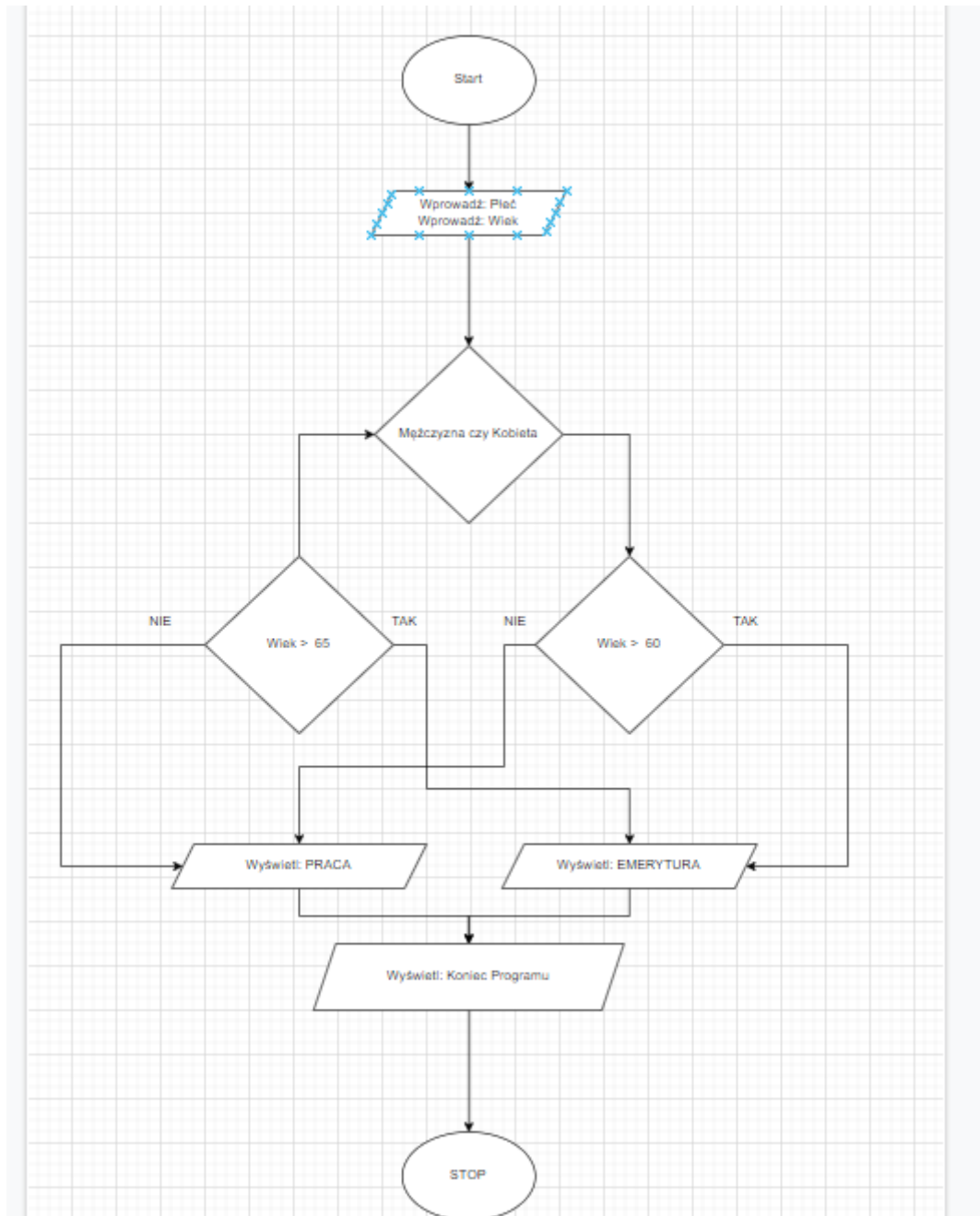


Zadanie C

Treść zadania :

Utworzyć schemat algorytmu sprawdzającego czy dana osoba może już przejść na emeryturę w zależności od podanego wieku i płci. Przyjmujemy wiek 60 lat dla kobiet i 65 dla mężczyzn. W efekcie działania algorytm wyświetla tylko słowo PRACA lub EMERYTURA.

Rozwiązanie :

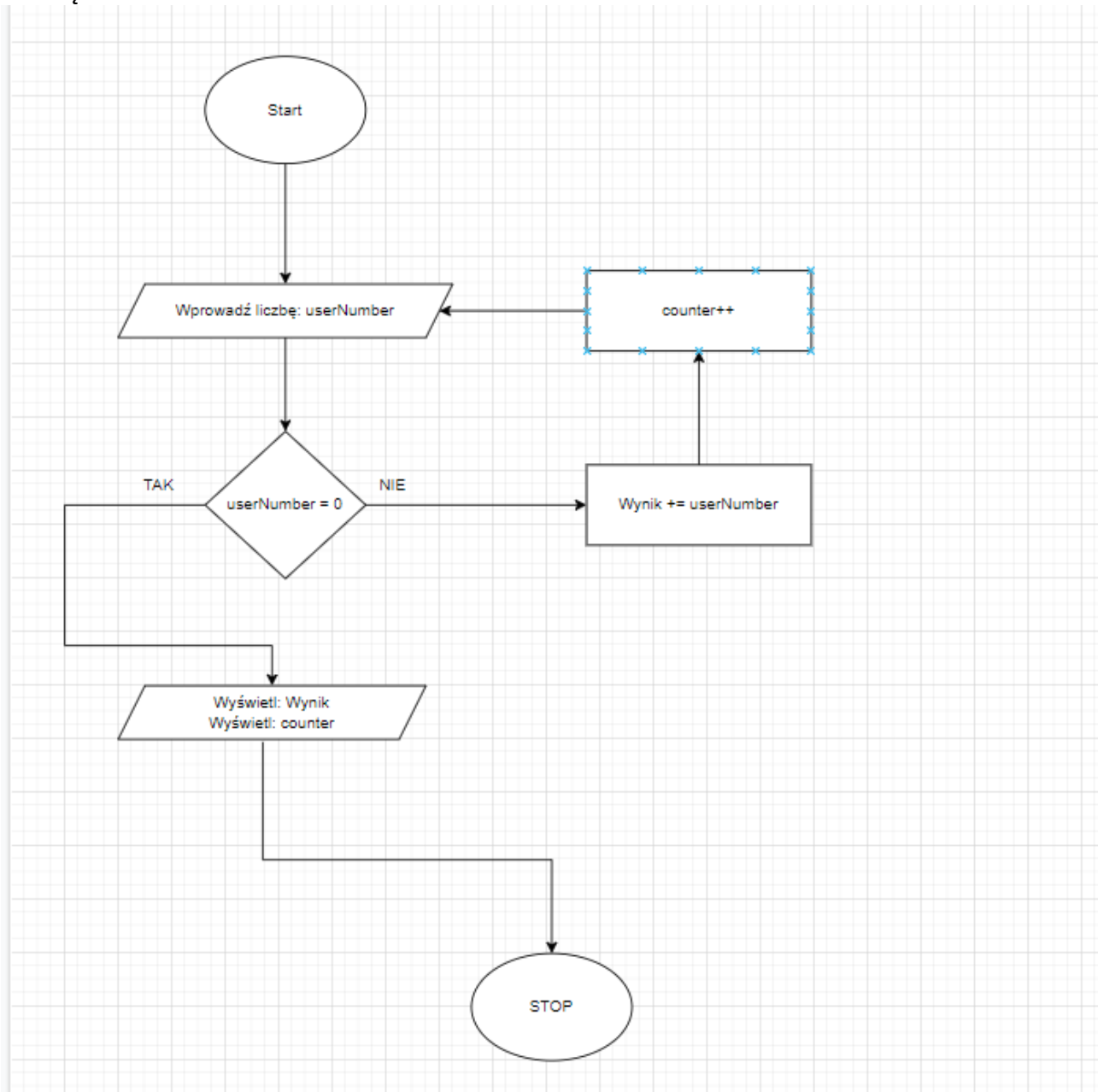


Zadanie D

Treść zadania :

Utworzyć schemat blokowy algorytmu wyznaczającego sumę oraz ilość liczb podawanych przez użytkownika tak długo, aż wprowadzi on liczbę zero. W tym momencie algorytm kończy wczytywanie liczb i wyświetla dane (suma, ilość). Liczba zero jest ogranicznikiem ciągu i nie wchodzi w jego skład. Utworzyć także pseudokod.

Rozwiązanie :



Zadanie 0

Treść zadania :

Prześledzić sposób i kolejność wykonywania następujących operacji:

$$3+2*5$$

$$10/2+7$$

$$10+1*2-3$$

$$6-3**2+7//2$$

$$6-3*2+8/4 \text{ a } (6-3)*(2+8)/4 \text{ a } (6-3)*2+(8/4)$$

Rozwiązanie :

```
# =====  
# M1Z0  
  
x = 3 + 2 * 5  
y = 10 / 2 + 7  
z = 10 + 1 * 2 - 3  
a = 6 - 3 ** 2 + 7 // 2  
abc1 = 6-3*2+8/4  
abc2 = (6-3)*(2+8)/4  
abc3 = (6-3)*2+(8/4)  
  
print('x = ' + str(x))  
print('x =', x, sep=" #", end=""" + " | ")  
print('y = ' + str(y))  
print('z = ' + str(z))  
print('a = ' + str(a))  
print('abc1 6-3*2+8/4 = ' + str(abc1))  
print('abc2 (6-3)*(2+8)/4 = ' + str(abc2))  
print('abc3 (6-3)*2+(8/4) = ' + str(abc3))  
  
# =====
```

Zadanie 1

Treść zadania :

Korzystając z interpretera online np onlinegdb.com wyświetlić na ekranie napis "Programowanie w Python". Kolejno, za pomocą jednej instrukcji wyświetlić ten napis w dwóch wierszach. Użyj dwóch konstrukcji: potrójnego apostrofu i kolejno znaku formatującego przejścia do nowego wiersza.

- Czy mamy tu dane wejściowe?
- Czy jest jakieś przetwarzanie danych?
- Czy mamy dane wyjściowe?
- W jaki sposób uzyskano dwa wiersze jedną instrukcją?

Rozwiązanie :

```
# =====  
# M1Z1  
  
print("Programowanie\nw Python")  
print("""Programowanie  
    w  
    Python""")  
  
# Brak danych wejściowych  
# jest przetwarzanie danych - wyświetlanie napisów  
# Dane wyjściowe: napis  
# poprzez \n lub """""" czyli jeden do jednego.  
# =====
```


Zadanie 2

Treść zadania :

Korzystając z interpretera dostarczonego z pakietem Python napisać program przechowujący dwie zmienne o wartościach np 20 i 30. Program ma wykonać na zmiennych operacje dodawania, odejmowania, mnożenia i dzielenia. Wyniki powinny być w czytelny sposób zaprezentowane użytkownikowi. Jak wygląda kwestia dzielenia? Jak należy dzielić aby uzyskać wynik rzeczywisty a jak całkowity z pominięciem części rzeczywistej?

Rozwiązanie :

```
# =====
# M1Z2
# zmienne
liczba1 = 20
liczba2 = 30

#Działania
print("Działania")
suma = liczba1 + liczba2
iloczyn = liczba1 * liczba2
odejmowanie1 = liczba1 - liczba2
odejmowanie2 = liczba2 - liczba1
dzielenie1 = liczba1 / liczba2
dzielenie2 = liczba2 / liczba1
dzielenieCalkowite1 = liczba1 // liczba2
dzielenieCalkowite2 = liczba2 // liczba1

#Wyświetlenie czytelne dla użytkownika
print(str(liczba1) + " + " + str(liczba2) + " = " + str(suma))
print(str(liczba1) + " * " + str(liczba2) + " = " + str(iloczyn))
print(str(liczba1) + " - " + str(liczba2) + " = " + str(odejmowanie1))
print(str(liczba2) + " - " + str(liczba1) + " = " + str(odejmowanie2))
print("Dzielenie rzeczywiste")
print(str(liczba1) + " : " + str(liczba2) + " = " + str(dzielenie1))
print(str(liczba2) + " : " + str(liczba1) + " = " + str(dzielenie2))
print("Dzielenie całkowite")
print(str(liczba1) + " : " + str(liczba2) + " = " + str(dzielenieCalkowite1))
print(str(liczba2) + " : " + str(liczba1) + " = " + str(dzielenieCalkowite2))

# Dzielenie rzeczywiste robimy jednym znakiem "/"
# Dzielenie całkowite wykonujemy dwoma znakami "/"

# =====
```

Zadanie 3

Treść zadania :

Za pomocą wbudowanego w pakiet Python programu IDLE napisać program prezentujący sposoby zapisu wartości liczbowych w systemach o podstawie 2, 8 i 16.

Rozwiązanie :

```
# =====  
# M1Z3  
print("Liczba 21")  
print('System binarny 10101 (a = 0b10101): ', end='')  
a = 0b10101  
print(a)  
  
print('System octal a = "0o25": ', end='')  
a = 0o25  
print(a)  
  
print('System hex a = "0x15": ', end='')  
a = 0x15  
print(a)  
# =====
```

Zadanie 4

Treść zadania :

Za pomocą środowiska VSC (o ile masz go zainstalowane) sprawdź, który z podanych sposobów zapisu liczby 7 jest poprawny. Co oznaczają wykorzystane operatory i sposoby zapisu? Utwórz zmienną o dowolnej nazwie, przypisz jej odpowiednią wartość i wyświetl.

```
7
7L
.7
7.
7,
7.0
7.e-1
7e+01
0.7e-1
0.7e+1
[3.0, 5.12, 70e-1][-1]
int('732')
int('7')
{3:[2,5], 'a':[5,12], 0:[1,7,6]}[0][1:2:1]
(2+7j).imag
07
0xA
1<<2+3
(1<<2)+3
(1<<2)+(1<<1)+1
15 & 7
22 // 3
21 % 14
```

Rozwiązanie :

```
# =====  
# M1Z4  
a = 7  
print("a=7 | ", a)  
# aa = 7L  
# aaa = . 7  
# b = 7 .  
bb = 7 ,  
print("bb= 7 | ", bb)  
# bbb = 7. 0  
# c = 7 . e-1  
cc = 7e+01  
print("cc=7e+01 | ", cc)  
# ccc = 0. 7 e-1  
# d = 0. 7 e+1  
# dd = [ 3 . 0 , 5 . 1 2 , 70e-1][-1]  
# ddd = i n t ( ' 7 3 2 ' )  
# e = i n t ( ' 7 ' )  
# ee = { 3 : [ 2 , 5 ] , ' a ' : [ 5 , 1 2 ] , 0 : [ 1 , 7 , 6 ] } [ 0 ] [ 1 : 2 : 1 ]  
# eee = (2+7 j ).imag  
# f = 07  
ff = 0xA  
print("0xA | ", ff)  
fff = 1<<2+3  
print("1<<2+3 | ", fff)  
g = (1<<2)+3  
print("(1<<2)+3 | ", g)  
gg = (1<<2)+(1<<1)+1  
print("(1<<2)+(1<<1)+1 | ", gg)  
ggg = 15 & 7  
print("15 & 7 | ", ggg)  
h = 22 // 3  
print("22 // 3 | ", h)  
hh = 21 % 14  
print("21 % 14 | ", hh)  
# =====
```

Zadanie 5

Treść zadania :

Korzystając z dowolnego interpretera (od tego zadania sam go wybierz) sprawdź, który ze sposobów zapisu liczby $1/2$ jest poprawny.

Utwórz zmienną o dowolnej nazwie, przypisz jej odpowiednią wartość i wyświetl.

```
5E-1
(3==3)/2
(3==3)/2.0
1/2
float(1)/2
5.
0.5L
5*((2>3)<2)
1.0>>2
1 / ( ( ' a'=='A' |2 <3)+1)
```

Rozwiązanie :

```
# =====
# M1Z5

zmienna01 = 5E-1
zmienna02 = (3 == 3)/2
zmienna03 = (3 == 3)/2.0
zmienna04 = 1/2
zmienna05 = float(1)/2
zmienna06 = 5.
#zmienna07 = 0.5L
zmienna08 = 5*((2>3)<2)
#zmienna09 = 1.0>>2
#zmienna10 = 1 / ( ( ' a'=='A' |2 <3)+1))
print("liczba z działania 5E-1 to: ", zmienna01)
print("liczba z działania (3==3)/2 to: ", zmienna02)
print("liczba z działania (3==3)/2.0 to: ", zmienna03)
print("liczba z działania 1/2to: ", zmienna04)
print("liczba z działania 1/2to: ", zmienna04)
print("liczba z działania float(1)/2 to: ", zmienna05)
print("liczba z działania 5. to: ", zmienna06)
print("liczba z działania 0.5L to: error")
print("liczba z działania 5*((2>3)<2)to: ", zmienna08)
print("liczba z działania 1.0>>2 to: error")
print("liczba z działania 1 / ( ( ' a'=='A' |2 <3)+1) to: error")

# =====
```

Zadanie 6

Treść zadania :

Określanie i porównywanie typów zmiennych.

Jakie typy danych wspiera Python i w jaki sposób? Jak język ten rozpoznaje i kategoryzuje wprowadzone dane?

Czym będzie się różnił zapis $x=3$ od $x=3.0$.

Czy poniższy zapis

```
x=3
```

```
x=3.5
```

```
x="ała ma kota"
```

```
x=x+1
```

jest poprawny? Z jakiego powodu?

Jaki jest wynik operacji:

```
x = 5 * 2.0
```

Jak można dokonać konwersji typu danych? Jak konwertować powyższy wynik do int i jaką przyjmie wartość?

Rozwiązanie :

```
# =====  
# M1Z6  
# Typy danych:  
# - numeryczne (liczbowe): int, float, complex  
# - tekstowe: str  
# - sekwencji: list, tuple  
# - odwzorowania (mapping type): dict  
# - zestawów (set types): set, frozenset  
# - logiczne: bool  
# - binarne: bytes, bytearray  
#  
# Typ danych ustawiamy przypisując daną wartość do zmiennej. Nie deklarujemy jakiego typu ma być zmienna.  
# Przyjmuje ona taki typ, jaki jest w przypisanej do niej wartości.  
  
a = 3  
print('-a jest to zmienna "a = 3": ', type(a))  
b = 3.5  
print('-b jest to zmienna "b = 3.5": ', type(b))  
c = "ała ma kota"  
print('-c jest to zmienna "c = ała ma kota": ', type(c))  
# d = d + 1 #niepoprawne  
  
e = 5*2.0  
print('-wynik działania: "e = 5*2.0" to: ', e)  
print('-e jest to zmienna typu: ', type(e))  
print('-konwertujemy zmienną typu float to int: "int(round(e))\n'  
      'round przy okazji zaokrągla ', int(round(e)))  
# =====
```

Zadanie 6 A

Treść zadania :

Napisać program wczytujący od użytkownika jego imię i wiek. Kolejno program wita się z użytkownikiem wyświetlając jego imię oraz informacje ile lat zostało mu do setki. Użyć stałej o nazwie STO.

Rozwiązanie :

```
# =====  
# M1Z6a  
  
STO = 100  
imie = input("Wprowadź imię: ")  
wiek = int(input("Wprowadź wiek: "))  
  
print("Twoje imię to: " + imie + ", a wiek: " + str(wiek))  
ile_do_setki = STO - wiek  
print("Do setki pozostało: " + str(ile_do_setki))  
  
# =====
```

Zadanie 7

Treść zadania :

Napisać program pozwalający przeliczać km -> mile (lądowe). Wartość km podaje użytkownik.

Rozwiązanie :

```
# =====  
# M1Z7  
  
PRZELICZNIK = 0.62137  
print("Podaj km do przeliczenia na mile: ", end="")  
km = int(input())  
mile = km * PRZELICZNIK  
print("Wynik", km, "km to:", round(mile, 2), "mil")  
# =====
```


Zadanie 8

Treść zadania :

Napisać program pozwalający na przeliczający stopni C na F. Wartość stopni C podaje użytkownik.

Rozwiązanie :

```
# =====  
# M1Z8  
  
tCelsjusz = input("Podaj wartość stopni Celsjusza aby przeliczyć na Fahrenheita: ")  
tCelsjusz = float(tCelsjusz)  
tCels = 9/5.0 * tCelsjusz + 32  
print("Temperatury w stopniach Fahrenheita to: ", round(tCels, 2))  
# =====
```

Zadanie 9

Treść zadania :

Napisać program pozwalający na obliczanie średniego zużycia paliwa na 100km. Wartość przebytej odległości oraz zużytego paliwa podaje użytkownik.

Rozwiązanie :

```
# =====  
# M1Z9  
  
przejechaneKm = float(input("Podaj ilość przejechanych km: "))  
iloscPaliwa = float(input("Podaj ilość spalonego paliwa: "))  
sredniaSpalania = iloscPaliwa / przejechaneKm * 100  
print("Średnia spalania to: ", round(sredniaSpalania, 2))  
# =====
```

Zadanie 10

Treść zadania :

Napisz program wczytujący od użytkownika rok jego urodzenia i podający ile ma dziś lat.

Rozwiązanie :

```
# =====  
# M1Z10  
from datetime import datetime  
  
teraz = datetime.now()  
  
print("Aktualnie mamy: ", teraz)  
  
rok = int(input("Podaj rok urodzenia: "))  
rok_aktualny = int(teraz.strftime("%Y"))  
wynik = rok_aktualny - rok  
print("Wiek: ", wynik)  
  
# =====
```

Zadanie 11

Treść zadania :

Napisać program wyświetlający użytkownikowi aktualną datę i pozwalającą obliczyć jaka data będzie za x dni, gdzie x podaje użytkownik.

Rozwiązanie :

```
# =====  
# M1Z11  
from datetime import datetime, timedelta  
  
today = datetime.now()  
print("Aktualna data to:", today.year)  
daysInput = int(input("Podaj ilość dni: "))  
today = today + timedelta(days=daysInput)  
print("Za", daysInput, "dni będzie rok", today.year)  
# =====
```

Zadanie 12

Treść zadania :

Napisz program wczytujący dwie daty i obliczający ich różnicę w dniach.

Rozwiązanie :

```
# =====  
# M1Z12  
  
import datetime  
  
today = datetime.date.today()  
endWorldWarII = datetime.date(1945, 5, 8)  
  
print("Dzisiejsza data to:", today)  
print("Data zakończenia drugiej wojny światowej:", endWorldWarII)  
difference = today - endWorldWarII  
  
print("Od drugiej wojny światowej minęło:", difference.days)  
  
# =====
```

Zadanie 13

Treść zadania :

Zadanie 13

Operacje matematyczne.

Napisać program przedstawiający sposoby

- zaokrąglania liczby rzeczywistej

- wyświetlania takiej wartości w środku ciągu znakowego na 10 polach z dokładnością do trzech miejsc po przecinku, dosunięte do lewej i kolejno do prawej a także z zerami wiodącymi.

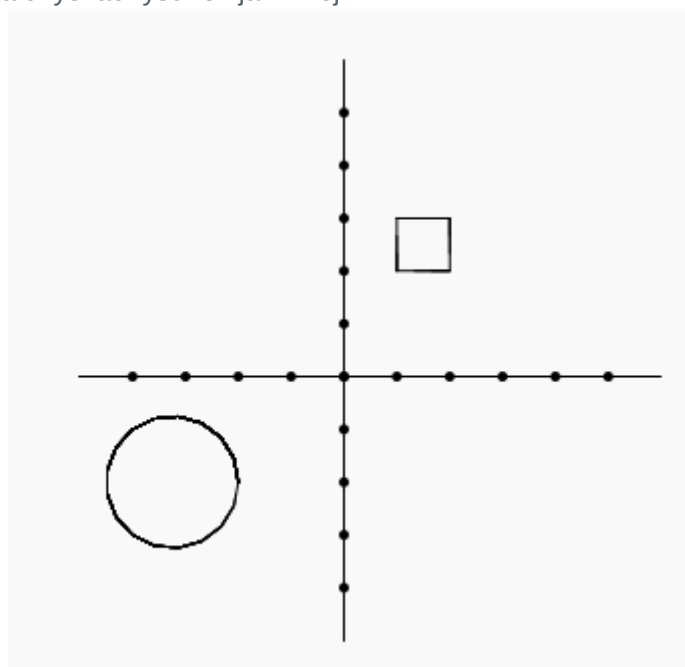
Rozwiązanie :

```
# =====  
# M1Z13  
import math  
  
numer = 45.12345676  
print(math.trunc(numer)) # ucięcie  
print(math.ceil(numer)) # do sufitu  
print(math.floor(numer)) # do podłogi  
print(round(numer, 3)) # zaokrąglenie  
print("Liczba: {0:.3f}".format(numer))  
# =====
```

Zadanie 14

Treść zadania :

Za pomocą grafiki żółwia uzyskać rysunek jak niżej:



Rozwiązanie :

```
# =====  
# M1Z14  
import turtle  
  
t = turtle.Turtle()  
  
t.penup()  
t.goto(-300, 0)  
t.pendown()  
for i in range(0, 11):  
    t.forward(50)  
    t.dot(5)  
t.forward(50)  
t.penup()  
t.goto(0, 300)  
t.right(90)  
t.pendown()  
for i in range(0, 11):  
    t.forward(50)  
    t.dot(5)  
t.forward(50)  
t.penup()  
t.goto(50, 150)  
t.pendown()  
for i in range(0, 4):  
    t.forward(50)  
    t.left(90)  
t.penup()  
t.left(90)  
t.goto(-200, -50)  
t.pendown()  
for i in range(0, 360):  
    t.forward(1)  
    t.right(1)  
t.penup()  
t.goto(0, 0)  
  
# =====  
# =====
```