

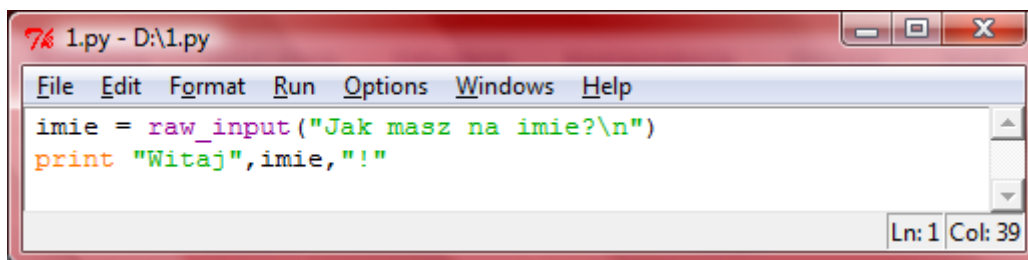

```

IDLE 2.6.5
>>> ===== RESTART =====
>>>
Jak masz na imie?Wojtek
Witaj Wojtek !
>>>

```

Nie ładnie to wygląda, że zaraz po pytaniu o imię...dopisujemy....swoj tekst...!

Jeśli chcemy aby imię było wpisywane w nowej linii wystarczy dopisać „\n” – co odpowiada dodaniu „nowej linii”



```

7% 1.py - D:\1.py
File Edit Format Run Options Windows Help
imie = raw_input("Jak masz na imie?\n")
print "Witaj", imie, "!"
Ln: 1 Col: 39

```

Teraz efekt będzie następujący:

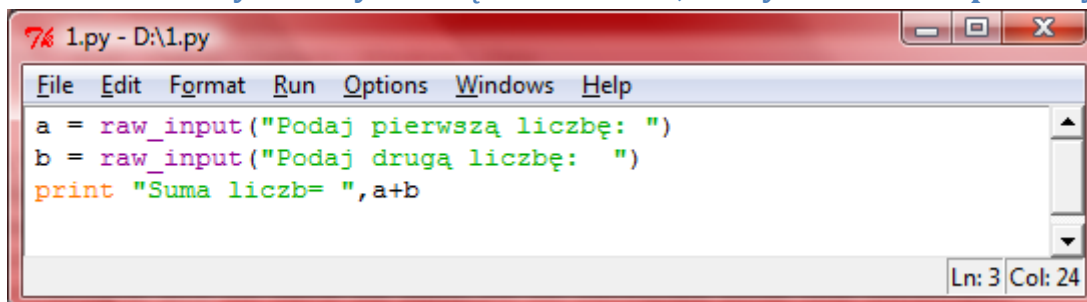
```

IDLE 2.6.5
>>> ===== RESTART =====
>>>
Jak masz na imie?Wojtek
Witaj Wojtek !
>>> ===== RESTART =====
>>>
Jak masz na imie?
Wojtek
Witaj Wojtek !
>>>

```

Lepiej, prawda ?

Krok 3. Chcemy obliczyć sumę dwóch liczb, których wartości podamy.



```

7% 1.py - D:\1.py
File Edit Format Run Options Windows Help
a = raw_input("Podaj pierwszą liczbę: ")
b = raw_input("Podaj drugą liczbę: ")
print "Suma liczb= ", a+b
Ln: 3 Col: 24

```

!!! tyle że to nie są liczby.....a teksty !!! W zmiennych [a] i [b] będą przechowane te wprowadzone wartości jako tekst a nie jako wartość liczbowa !!! Przez to efekt będzie następujący !!!

```

>>> ===== RESTART =====
>>>
Podaj pierwszą liczbę: 4
Podaj drugą liczbę: 3
Suma liczb= 43
>>> |

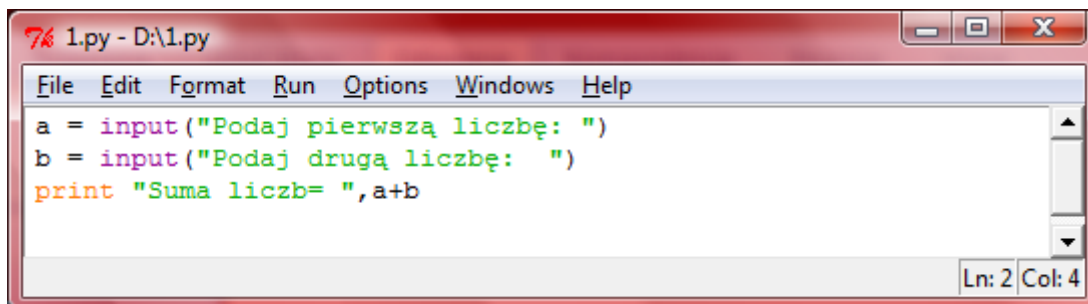
```

Nie tego chcieliśmy prawda ? Aby tekst traktować jak tekst, a liczby jak liczby używaj innych funkcji:

input() – dla liczb

raw_input() – dla tekstu

A więc:



```

7% 1.py - D:\1.py
File Edit Format Run Options Windows Help
a = input("Podaj pierwszą liczbę: ")
b = input("Podaj drugą liczbę: ")
print "Suma liczb= ",a+b
Ln: 2 Col: 4

```

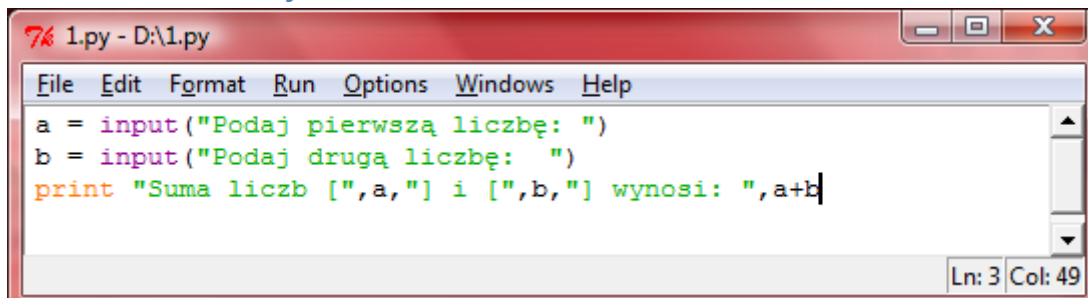
```

>>> ===== RESTART =====
>>>
Podaj pierwszą liczbę: 4
Podaj drugą liczbę: 3
Suma liczb= 7
>>> |

```

Teraz jest OK. !

Krok 3a. Chcemy by nam się ładnie wyświetlało dla jakich liczb była suma liczona i ile ona wynosi !

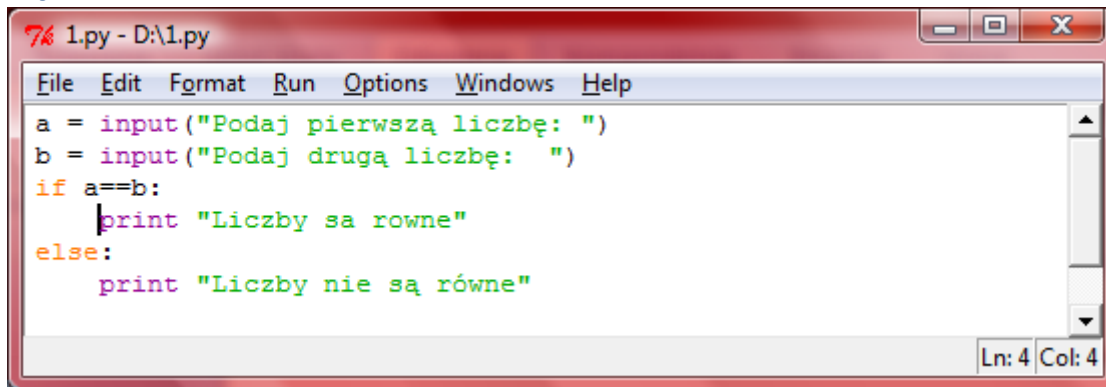


```

7% 1.py - D:\1.py
File Edit Format Run Options Windows Help
a = input("Podaj pierwszą liczbę: ")
b = input("Podaj drugą liczbę: ")
print "Suma liczb [" ,a, "] i [" ,b, "] wynosi: ",a+b|
Ln: 3 Col: 49

```

Krok 4. Chcemy sprawdzać, czy dwie wprowadzone liczby są równe, czy różne !

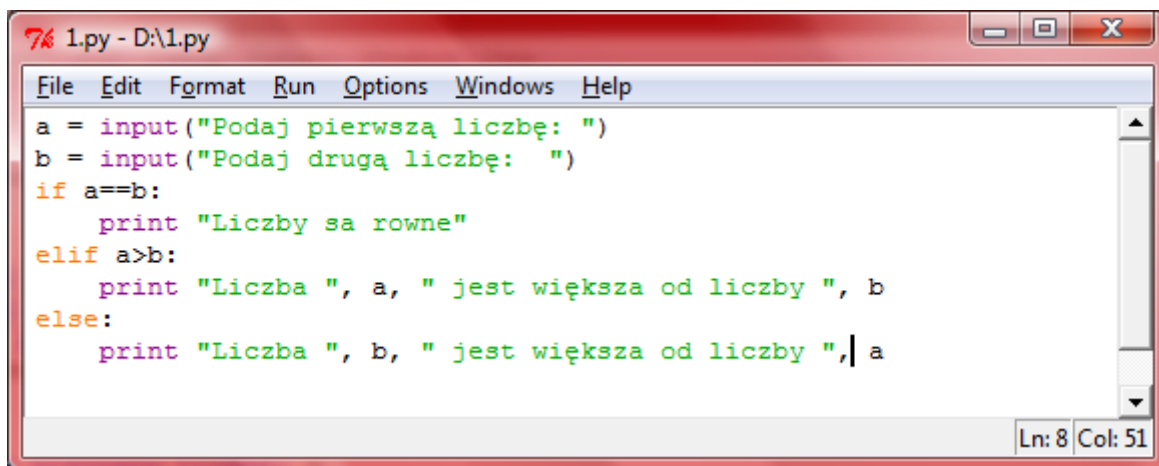


```
7% 1.py - D:\1.py
File Edit Format Run Options Windows Help
a = input("Podaj pierwszą liczbę: ")
b = input("Podaj drugą liczbę: ")
if a==b:
    print "Liczby sa rowne"
else:
    print "Liczby nie są równe"
Ln: 4 Col: 4
```

Czego efekt będzie następujący !

```
>>> =====
>>>
Podaj pierwszą liczbę: 4
Podaj drugą liczbę: 3
Liczby nie są równe
>>>
```

A co gdy jest więcej możliwości niż tylko „liczby są równe” albo „nie są równe”?



```
7% 1.py - D:\1.py
File Edit Format Run Options Windows Help
a = input("Podaj pierwszą liczbę: ")
b = input("Podaj drugą liczbę: ")
if a==b:
    print "Liczby sa rowne"
elif a>b:
    print "Liczba ", a, " jest większa od liczby ", b
else:
    print "Liczba ", b, " jest większa od liczby ", a
Ln: 8 Col: 51
```

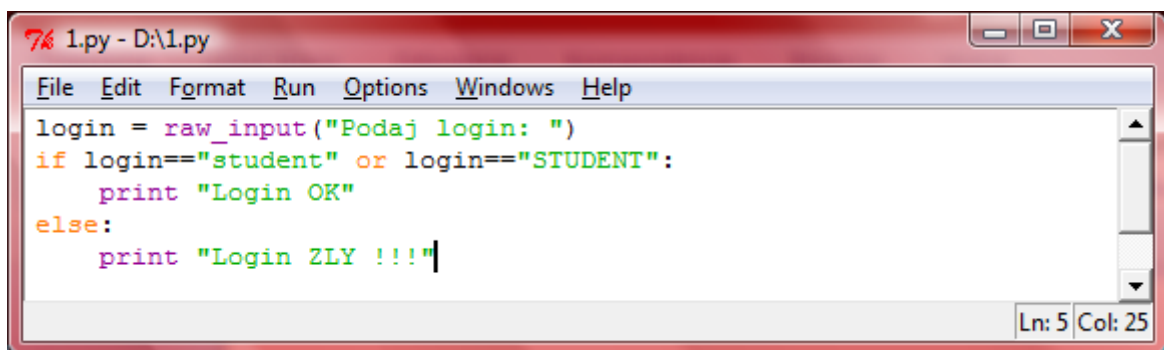
Co w efekcie da nam coś takiego:

```
>>> ===== RESTART :
==
>>>
Podaj pierwszą liczbę: 3
Podaj drugą liczbę: 4
Liczba 4 jest większa od liczby 3
>>> |
```

Ćwiczenia do wykonania

1. Napisz program, który pyta o wiek i w zależności od podanej wartości, mówi czy ktoś jest pełnoletni czy nie.
2. Rozbuduj program 1 tak, by podawał:
 - Dla osoby pełnoletniej od ilu lat jest pełnoletnia
 - Dla osoby niepełnoletniej – ile lat brakuje jej jeszcze do pełnoletności

Krok 5. Sprawdzanie logiczne. Np. chcemy sprawdzić, czy użytkownik podał właściwy login czy hasło.



```
7% 1.py - D:\1.py
File Edit Format Run Options Windows Help
login = raw_input("Podaj login: ")
if login=="student" or login=="STUDENT":
    print "Login OK"
else:
    print "Login ZLY !!!"
Ln: 5 Col: 25
```

Scenariusze programu mogą być różne:

```
==
>>>
Podaj login: aga
Login ZLY !!!
>>> ===== RESTART =====
==
>>>
Podaj login: student
Login OK
>>> ===== RESTART =====
==
>>>
Podaj login: STUDENT
Login OK
>>>
```

Ćwiczenia do wykonania:

1. Sprawdź czy podana liczba jest parzysta czy nie i wyświetl odpowiedni komunikat. Tak jak na rysunku poniżej:

```

>>> ===== RESTART =====
==
>>>
program pyta o liczbe i mowi czy byla parzysta czy nie
podaj liczbe: 5
nieparzysta
>>> ===== RESTART =====
==
>>>
program pyta o liczbe i mowi czy byla parzysta czy nie
podaj liczbe: 4
parzysta
>>>

```

2. Napisz program, który pyta o wartości [a] i [b] i przedstawia postać równania liniowego z tymi parametrami. Przykładowy scenariusz programu taki jak na rysunku poniżej:

```

>>> ===== RESTART =====
==
>>>
podaj [a]: 4
podaj [b]: 3
y= 4 x + 3
>>> ===== RESTART =====
==
>>>
podaj [a]: -4
podaj [b]: 0.5
y= -4 x + 0.5
>>> |

```

3. Rozbuduj program 2 tak by przedstawiał jaka będzie wartość zmiennej [y] w zależności od wartości [x] – którą podaje użytkownik. Przykładowy scenariusz programu przedstawia rysunek poniżej:

```

>>> ===== RESTART =====
==
>>>
podaj [a]: 1
podaj [b]: 2
y= 1 x + 2
dla jakiej wartości [x] obliczyć wartość funkcji [y] ?: 6
y= 8
>>> |

```

4. Napisz program, który pyta o 3 parametry i przedstawia postać równania kwadratowego z tymi parametrami. Przykładowy scenariusz:

```

>>> ===== RESTART =====
==
>>>
podaj [a]: 1
podaj [b]: 2
podaj [c]: 1
równanie ma postać: 1 x^2 + 2 x + 1
>>> |

```

5. Rozbuduj program 4 tak by obliczał DELTĘ. Scenariusz poniżej:

```
>>> ===== RESTART =====
==
>>>
podaj [a]: 1
podaj [b]: 2
podaj [c]: 1
równanie ma postać: 1 x^2 + 2 x + 1
Delta = 0
>>>
```

6. Rozbuduj program 5 tak by mówił ile rozwiązań ma takie równanie kwadratowe. Scenariusz poniżej:

```
>>> ===== RESTART =====
>>>
podaj [a]: 1
podaj [b]: 2
podaj [c]: 1
równanie ma postać: 1 x^2 + 2 x + 1
Delta = 0
Delta = 0, więc równanie ma 1 pierwiastek
>>> ===== RESTART =====
>>>
podaj [a]: 5
podaj [b]: 1
podaj [c]: 1
równanie ma postać: 5 x^2 + 1 x + 1
Delta = -19
Delta jest ujemna, a więc równanie nie ma pierwiastków !
>>> ===== RESTART =====
>>>
podaj [a]: 1
podaj [b]: 4
podaj [c]: 1
równanie ma postać: 1 x^2 + 4 x + 1
Delta = 12
Delta > 0, więc równanie ma 2 pierwiastki.
>>>
```

7. Rozbuduj program 6 tak, by obliczał pierwiastki równania kwadratowego. Scenariusz poniżej:

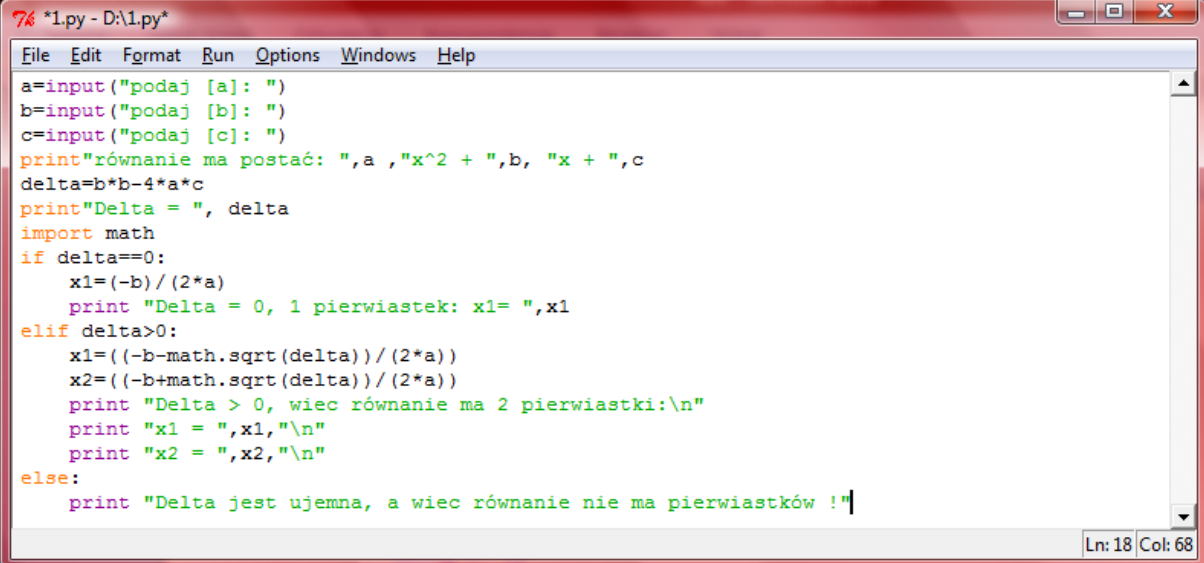
```

>>> ===== RESTART =====
>>>
podaj [a]: 1
podaj [b]: 2
podaj [c]: 1
równanie ma postać: 1 x^2 + 2 x + 1
Delta = 0
Delta = 0, 1 pierwiastek: x1= -1
>>> ===== RESTART =====
>>>
podaj [a]: 1
podaj [b]: 4
podaj [c]: 1
równanie ma postać: 1 x^2 + 4 x + 1
Delta = 12
Delta > 0, więc równanie ma 2 pierwiastki:

x1 = -3.73205080757
x2 = -0.267949192431

```

Podpowiedź:

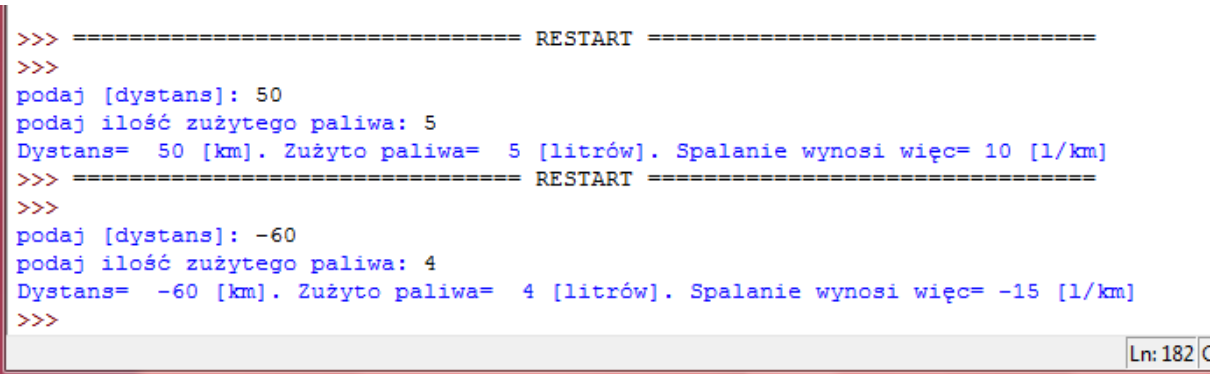


```

% *l.py - D:\l.py*
File Edit Format Run Options Windows Help
a=input("podaj [a]: ")
b=input("podaj [b]: ")
c=input("podaj [c]: ")
print"równanie ma postać: ",a ,"x^2 + ",b, "x + ",c
delta=b*b-4*a*c
print"Delta = ", delta
import math
if delta==0:
    x1=(-b)/(2*a)
    print "Delta = 0, 1 pierwiastek: x1= ",x1
elif delta>0:
    x1=(-b-math.sqrt(delta))/(2*a)
    x2=(-b+math.sqrt(delta))/(2*a)
    print "Delta > 0, więc równanie ma 2 pierwiastki:\n"
    print "x1 = ",x1,"\n"
    print "x2 = ",x2,"\n"
else:
    print "Delta jest ujemna, a więc równanie nie ma pierwiastków !"
Ln: 18 Col: 68

```

8. Napisz program, który oblicza średnie spalanie samochodu. Wiadomo, że trzeba podać dystans który się przejechało i ilość paliwa, którą się zużyło. Pamiętajmy, że nasz program powinien być odporny na sytuację taką jak przedstawia rysunek poniżej:



```

>>> ===== RESTART =====
>>>
podaj [dystans]: 50
podaj ilość zużytego paliwa: 5
Dystans= 50 [km]. Zużyto paliwa= 5 [litrów]. Spalanie wynosi więc= 10 [l/km]
>>> ===== RESTART =====
>>>
podaj [dystans]: -60
podaj ilość zużytego paliwa: 4
Dystans= -60 [km]. Zużyto paliwa= 4 [litrów]. Spalanie wynosi więc= -15 [l/km]
>>>
Ln: 182 C

```

Program ma więc wyświetlać użytkownikowi komunikat o błędnie wprowadzonych danych, gdy któraś z podanych wartości jest mniejsza bądź równa 0. Jak na rysunku poniżej:

```

>>>
podaj [dystans]: -5
Dystans powinien byc wartoscia większą od 0 !
>>> ===== RESTART =====
>>>
podaj [dystans]: 4
podaj ilość zużytego paliwa: -9
Nie zużyłeś wcale paliwa ? to chyba niemożliwe ?!
>>> ===== RESTART =====
>>>
podaj [dystans]: 6
podaj ilość zużytego paliwa: 2
Dystans= 6 [km]. Zużyto paliwa= 2 [litrów]. Spalanie wynosi więc= 3 [l/km]
>>> |

```

Programy dodatkowe:

1. Obliczanie podatku dochodowego
2. Konwersja jednostek miar, walut (np. kilometry na mile, złotówki na euro, itp)
3. Obliczanie BMI

Wiedząc, że wskaźnik BMI oblicza się następująco:

Wskaźnik Masy Ciała obliczamy
wg poniższego wzoru:

$$\text{BMI} = \frac{\text{masa ciała [kg]}}{\text{wzrost}^2 \text{ [m}^2\text{]}}$$

Oraz, że możemy na podstawie wyznaczonej wartości BMI określać przynależność do klasy: niedowagi, nadwagi czy nawet otyłości, jak poniżej:

Klasyfikacja zaburzeń masy ciała	BMI (kg/m ²)	Ryzyko chorób towarzyszących otyłości
Niedowaga	<18,5	Niskie (ale zwiększone ryzyko innych problemów zdrowotnych)
Norma	18,5 - 24,9	Średnie
Nadwaga	>= 25	
Okres przed otyłością	25,0-29,9	Podwyższone
I° otyłości	30,0-34,9	Umiarkowanie podwyższone
II° otyłości	35,0-39,9	Wysokie
III° otyłości	>= 40,0	Bardzo wysokie

Klasyfikacja otyłości w zależności od BMI wg WHO, Report of a WHO "Consultation on Obesity", Genewa 1997

Napisz program, który dla podanych wartości: [waga] oraz [wzrost] oblicza BMI i podaje klasyfikację zaburzeń masy ciała.

4. Określanie poziomu ciśnienia – na podstawie informacji podawanych przez tabelę poniżej:

Klasyfikacja nadciśnienia tętniczego wg ESH/ESC^[1]

Kategoria	Ciśnienie skurczowe (mmHg)	Ciśnienie rozkurczowe (mmHg)
Ciśnienie optymalne	< 120	< 80
Ciśnienie prawidłowe	120–129	80–84
Ciśnienie wysokie prawidłowe	130–139	85–89
Nadciśnienie stopień 1 (łagodne)	140–159	90–99
Nadciśnienie stopień 2 (umiarkowane)	160–179	100–109
Nadciśnienie stopień 3 (ciężkie)	≥ 180	≥ 110

I znów: napisz program, który dla podanej wartości ciśnienia skurczowego i rozkurczowego podaje kategorię wg klasyfikacji podanej w tabeli.

I jak było ?