

Zadania (07.11.2025)

1 Zakres zmiennych

Co zwróci poniższy kod? Dlaczego? Co się zmieni, gdy zakomentujemy w funkcji linię `count =1` oraz zastąpimy linię `count += 1` przez `continue`?

```
count = 5

def modify_count():
    count = 1
    for i in range(3):
        count +=1
        print("Wewnątrz funkcji:", count*i)

modify_count()
print("Poza funkcją:", count)
```

2 Suma ważona

Napisz funkcję (wraz z wykorzystującym ją programem), która będzie przyjmowała następujące argumenty:

1. wartość początkową,
2. dowolnie wiele dodatkowych liczb.

Dodatkowe liczby mają zostać przemnożone przez ich pozycję na liście argumentów (pierwsza przez 1, druga przez 2 itd.) i dodane do wartości początkowej, po czym zwrócone przez funkcję.

3 Adresy IP

Adres IP(v4) zwyczajowo przedstawiany jest jako czwórka liczb całkowitych dodatnich z zakresu 8-bitowego, tzn. z przedziału $[0, 255]$, oddzielonych kropkami (np. 192.168.0.1). Szczególne znaczenie mają wymienione poniżej zakresy IP, gdyż rezerwuje się je do użytku w sieci lokalnej:

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

Napisz funkcję (wraz z wykorzystującym ją programem), która będzie przyjmowała jako argument listę lub krotkę z adresami IP (np. tę niżej), po czym zwróci jako wynik krotkę zawierającą jedynie adresy lokalne.

```
ips = [
    "192.168.0.7",
    "198.23.12.9",
    "13.8.32.1",
    "10.11.2.37",
    "11.11.3.123",
    "199.231.1.0",
    "61.221.34.59",
    "48.54.206.145",
    "21.181.83.183",
    "172.30.27.219",
    "123.170.108.177",
    "216.227.101.121",
    "51.126.7.165",
    "23.96.221.226",
    "208.212.144.77",
    "192.152.96.133",
    "37.13.232.193",
    "211.121.69.142",
    "16.122.66.255",
    "118.57.180.250",
    "228.65.59.248",
    "172.175.159.72",
    "10.195.34.158",
]
```

4 Trójkąt Pascala

Wygeneruj tzw. trójkąt Pascala, czyli taki, na którego bokach stoją jedynki, a każda inna liczba jest sumą dwóch leżących bezpośrednio ponad nią. Innymi słowy, liczby stojące w n -tym wierszu to kolejne współczynniki dwumianu Newtona: $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ (w szczególności $\binom{n}{0} = \binom{n}{n} = 1$).

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
```

5 Częstość występowania znaków

Napisz program, który będzie przyjmował jako argument wywołania nazwę pliku tekstowego, po czym policzy częstość znaków, które w nim występują. Niech zwraca wynik w postaci histogramu podobnego do tego niżej:

Rozkład dla pliku: nevergonnagiveyouup.txt:

:	#####	288
e :	#####	181
n :	#####	153
o :	#####	150
a :	#####	105
r :	#####	87
u :	#####	76
t :	####	69
g :	####	66
y :	####	64
\n:	###	54
d :	###	49
v :	###	45
l :	##	43
s :	##	36
N :	##	36
i :	##	31
h :	##	30
w :	#	26
m :	#	20
' :	#	20
k :	#	17
b :	#	15
c :		11
f :		9
I :		9
p :		8
W :		5
Y :		4
,	:	4
j :		2
G :		2
A :		2
D :		1

6 Zadanie domowe: Sito Eratostenesa

Sito Eratostenesa to dość sprytny sposób na znalezienie wszystkich liczb pierwszych –podzielnych tylko przez siebie i 1 (choć 1, o ironio, nie jest liczbą pierwszą), mniejszych od zadanego n . Polega ono na:

1. utworzeniu tablicy liczb naturalnych mniejszych niż n
2. przejściu przez kolejne dotychczas nie wykreślone liczby (nazwijmy je i)
3. skreślaniu (możemy to zrealizować przez przypisanie wartości **False**) wszystkich liczb będących wielokrotnościami i

Te liczby, które przetrwają powyższą procedurę, niechybnie muszą być pierwsze.

Wskazówki (nie są niezbędne, ale nieco przyspieszają działanie programu):

1. Nie musimy sprawdzać wszystkich liczb aż do n , bo jeśli jakaś liczba się wyraża przez $l = a \cdot b$, to jedno z $\{a, b\}$ zawsze będzie mniejsze niż $\sqrt{n}$. Wystarczy więc iterować i tylko do $\sqrt{n}$.

2. Wykreślamy wielokrotności kolejnych liczb. Należałoby więc teoretycznie zacząć od $2 \cdot i$. Jednakże jeżeli dla wcześniej sprawdzanych liczb mniejszych niż $i \cdot i$ istniała jakaś liczba złożona (nie-pierwsza), to musiała już mieć czynniki mniejsze niż $\sqrt{i^2} = i$, więc została już wykreślona. Np. dla $i = 3$ nie musimy sprawdzać $6 = 2 \cdot 3$, bo zostało już wykreślone przy $i = 2$: $6 = 3 \cdot 2$. Sprawdzanie możemy zatem zaczynać od i^2 .