

PYTHON – FOR - NAPISY – ĆWICZENIA (21)

Napis (łańcuch tekstowy), to ciąg znaków ujęty w cudzysłowy lub apostrofy: "Wacek" lub 'Wacek Libront'.

Napisem jest też tzw. łańcuch pusty: ''.

Znaki w napisach są ponumerowane. Pierwszy znak w napisie ma numer zero.

Dostęp do pojedynczych znaków realizujemy za pomocą indeksowania []

Pętla FOR wykonuje instrukcje wiele razy.

Funkcja RANGE zawiera listę kolejnych liczb: 0, 1, 2,

UWAGA

- obok instrukcji przykładowe wyniki działania programu
- wcięcia są bardzo ważne (ale nawet jedna spacja wystarczy)

Łańcuchy (3)

- Wpisz program do edytora
- Zamień nazwisko i imię na swoje i uruchom program

```
NA='Libront'
IM='Wacław'
OS=NA+' '+IM
print(OS)
print(len(OS))
print(OS[0],OS[1],OS[8],OS[-2],OS[-1])
print(NA[0]+IM[0])
```

Libront Wacław
14
L i W a w
LW

Przypisz do zmiennej o nazwie **NA** łańcuch tekstowy ze swoim nazwiskiem

Przypisz do zmiennej o nazwie **IM** łańcuch tekstowy ze swoim imieniem

Połącz oba napisy, rozdziel je spacją i przypisz do zmiennej **OS**

Wypisz zawartość tego łańcucha na ekranie

Wypisz na ekranie długość tego łańcucha (ile ma znaków) – funkcja **len**

Wypisz na ekranie: pierwszy, drugi, dziewiąty, przedostatni i ostatni znak łańcucha (nawiasy klamrowe)

Wypisz na ekranie inicjały (pierwsze znaki nazwiska i imienia)

- Wklej zrzut programu i wyników działania programu do ramki

Pętle (3)

- Zmienną OS zachowaj z poprzedniego zadania
- Wpisz program do edytora i uruchom

```
for i in range(10):
    print(i+1)
```

1
2
3
4
5
6
7
8
9
10

Pętla FOR wypisuje na ekranie liczby od 1 do 10

Zmienna sterująca pętlą „i” przyjmuje wartości od 0 do 9, dlatego powiększamy ją o 1

```
OS='Libront Wacław'
dlug=len(OS)
for i in range(dlug):
    print(OS[i])
for i in range(dlug):
    print(OS[dlug-i-1])
```

L
i
b
r
o
n
t
W
a
c
ł
a
w
t
n
o
r
b
i
l

Funkcja **LEN** wylicza długość napisu – liczbę znaków

Pętla FOR wypisuje na ekranie kolejne znaki ze zmiennej **OS**

Kolejna pętla FOR wypisuje na ekranie znaki ze zmiennej **OS**, ale w odwrotnej kolejności

- Wklej zrzut programu i wyników działania programu do ramki

Zliczanie (3)

DNA to kwas deoksyrybonykleinowy, którego najważniejszą częścią są cztery zasady azotowe: A,G,C,T, występujące w łańcuchach, np. "TCTAACAGCCCCATATCGGG"

Napisz funkcję, która policzy znaki w napisie

- Wpisz program do edytora i uruchom

```
def PoliczZnaki(napis, znak):
    licznik=0
    for i in range(len(napis)):
        if napis[i]==znak:
            licznik=licznik+1
    return licznik
```

```
DNA='TCTAACAGCCCCATATCGGG'
print(DNA)
print('A', PoliczZnaki(DNA, 'A'))
print('C', PoliczZnaki(DNA, 'C'))
print(OS)
```

```
TCTAACAGCCCCATATCGGG
A 5
C 7
Libront Waclaw
a 2
```

Funkcja **PoliczZnaki** ma dwa parametry:

napis – tekst, w którym szukamy znaków

znak – litera, której szukamy w napisie

Licznik znaków ustawiamy na zero

Pętla FOR numeruje od zero do długość napisu (mniej jeden)

Jeżeli znak w napisie jest taki sam jak poszukiwany,
to powiększamy licznik o jeden

Funkcja zwraca liczbę znalezionych znaków

- Sprawdź ile znaków „a” znajduje się w zmiennej OS (Twoim nazwisku i imieniu)
- Wklej zrzut programu i wyników działania programu do ramki

UWAGA – Python posiada gotową funkcję zliczającą znaki: `print(DNA.count('A'))`

Pary znaków (3)

Napisz funkcję, która będzie szukać, liczyć i wypisywać pary takich samych znaków w tekście.

- Wpisz program do edytora i uruchom

```
def ParyZnakow(napis):
    licznik=0
    for i in range(len(napis)-1):
        if napis[i]==napis[i+1]:
            licznik=licznik+1
            print(napis[i]*2)
    return licznik
print(DNA)
print(ParyZnakow(DNA))
zdanie='Ranny i bezbronny męzczennik'
print(zdanie)
```

```
TCTAACAGCCCCATATCGGG
```

```
AA
CC
CC
CC
GG
GG
6
Ranny i bezbronny męzczennik
nn
nn
nn
nn
3
```

Funkcja **ParyZnakow** ma jeden parametr:

napis – tekst, w którym szukamy takich samych znaków znajdujących obok siebie

Licznik ustawiamy na zero

Pętla FOR numeruje od zero do długość napisu

Odejmujemy jeden, bo sprawdzamy zawsze ten i następny znak, a ostatni obrót pętli nie miałby co sprawdzać (błąd)

Jeżeli znak w napisie jest taki sam jak kolejny znak, to
powiększamy licznik o jeden

wypisujemy na ekranie dwa takie same znaki (mnożenie napisów je powieli)

Funkcja zwraca liczbę par znaków

- Sprawdź ile par znajduje się w tekście **Ranny i bezbronny męzczennik**
- Wklej zrzut programu i wyników działania programu do ramki

Odwracanie (3)

Napisz funkcję, która będzie odwracać kolejność znaków w tekście.

- Zmienną DNA zachowaj z poprzedniego zadania
- Wpisz program do edytora i uruchom

```
def WSPAK(napis):
    n=''
    for i in range(len(napis)):
        n=napis[i]+n
    return n

d=WSPAK(DNA)
print(DNA)
print(d)
print(OS)
```

TCTAACAGCCCCATATCGGG
GGGCTATACCCCGACAATCT
Libront Waclaw
wałcaW tnorbiL

Funkcja WSPAK ma jeden parametr;

napis – tekst, który będziemy odwracać

Zmienna „n” zawiera napis pusty – do „n” będziemy doklejać kolejne litery

Pętla FOR numeruje; od zero do długość napisu

do zmiennej „n” doklejamy na początek kolejne znaki z napisu

w ten sposób ostatni znak znajdzie się na początku

Funkcja zwraca napis w zmiennej „n”

- Odwróć swoje imię i nazwisko
- Wklej zrzut programu i wyników działania programu do ramki

UWAGA – Nie usuwaj funkcji WSPAK

Zamiana (3)

Napisz funkcję, która będzie zamieniać znaki w łańcuchu

- Wpisz program do edytora i uruchom

```
def ZAMIANA(napis, z1, z2):
    n=''
    for i in range(len(napis)):
        z=napis[i]
        if z==z1: z=z2
        elif z==z2: z=z1
        n=n+z
    return n

print(DNA)
print(ZAMIANA(DNA, 'T', 'A'))
```

TCTAACAGCCCCATATCGGG
ACATTCTGCCCTATACGGG

Funkcja ZAMIANA ma trzy parametry:

napis – tekst, w którym szukamy znaków

z1 i z2 – znaki, które zamieniamy między sobą

Zmienna n zawiera napis pusty – tutaj będziemy doklejać kolejne znaki

Pętla FOR numeruje od zero do długość napisu

do zmiennej „z” pobieramy kolejny znak z napisu

Jeżeli badany znak jest taki sam jak pierwszy parametr – z1, to zamieniamy go na drugi – z2

Jeżeli badany znak jest taki sam jak drugi parametr – z2, to zamieniamy go na pierwszy – z1

Do zmiennej „n” doklejamy kolejny znak z napisu – zamieniony w instrukcji jeżeli

Funkcja zwraca napis w zmiennej „n”

- Zastąp w swoim nazwisku i imieniu wszystkie a na e, wszystkie u na o, wszystkie i na y

```
print(OS)
o=ZAMIANA(OS, 'a', 'e')
o=ZAMIANA(o, 'u', 'o')
```

Libront Waclaw
Lybrunt Weclaw

- Wklej zrzut programu i wyników działania programu do ramki

UWAGA – Nie usuwaj funkcji ZAMIANA

Replikacja (3)

Replikacja DNA polega na odwracaniu kolejności fragmentów DNA i zamianie par zasad A-T i G-C
Napisz funkcję, która będzie zamieniać znaki w łańcuchu DNA, a następnie odwracać ich kolejność.

- Wpisz program do edytora i uruchom

```
def REPLIKACJA(napis):  
    napis=ZAMIANA(napis, 'A', 'T')  
    napis=ZAMIANA(napis, 'C', 'G')  
    napis=WSPAK(napis)  
    return napis
```

```
DNA='TAGC'  
print(DNA)  
print(REPLIKACJA(DNA))
```

```
TAGC  
GCTA  
Libront Wacław  
wełceW tnurbyL
```

Funkcja RELIKACJA ma jeden parametr:

napis – w którym będziemy zamieniać pary liter

Za pomocą funkcji ZAMIANA zamieniamy litery A-T - wynik zapisujemy znów w zmiennej napis

Za pomocą funkcji ZAMIANA zamieniamy litery C-G - wynik zapisujemy znów w zmiennej napis

Za pomocą funkcji WSPAK odwracamy znaki w napisie - wynik zapisujemy ponownie w zmiennej napis

Funkcja zwraca napis po zamianach o odwróceniu znaków

- Popraw funkcję REPLIKACJA by zamieniała pary liter a-e, o-u, i-y
- Zreplikuj zmienną OS (swoje nazwisko i imię)
- Wklej zrzut programu i wyników działania programu do ramki