

The background is a dark blue gradient. On the left side, there is a large, semi-circular scale with tick marks and numbers ranging from 140 to 260. Several concentric circles and arcs are scattered across the left half of the image, some with arrows indicating a clockwise direction. On the right side, there are more concentric circles and arcs, some with arrows indicating a counter-clockwise direction. The overall design is abstract and geometric.

TEKST JAKO LISTA LITER

ZNAJDYWANIE FRAGMENTÓW W TEKSTACH

```
a='Beata lubi wiosnę i lato'
```

```
print('lubi' in a)
```

```
print('Lubi' in a)
```

```
print(a.find('lubi'))
```

```
print(a.find('Lubi'))
```

```
print(a.find('Beatka'))
```

Jaki efekt uzyskamy na monitorze?

True *(można w instrukcji warunkowej zamiennie użyć 1 np.*

```
if ('lubi' in a)==1:
```

```
print('tak')
```

False *(można w instrukcji warunkowej zamiennie użyć 0)*

6 *(czyli nr pozycji w tekście od którego zaczyna się szukany fragment)*

-1 *(jeżeli w łańcuchu znaków nie występuje szukany fragment uzyskujemy -1)*

-1 *(jeżeli w łańcuchu znaków nie występuje szukany fragment uzyskujemy -1)*

ZNAJDYWANIE FRAGMENTÓW W TEKSTACH

```
a='Beata lubi wiosnę i lato'  
print(a.startswith('Beata'))  
print(a.endswith('lato'))  
print(a.startswith('beata'))  
print(a.startswith('Beatka'))
```

Jaki efekt uzyskamy na monitorze?

True *(można w instrukcji warunkowej zamiennie użyć 1)*

True *(można w instrukcji warunkowej zamiennie użyć 1)*

False *(można w instrukcji warunkowej zamiennie użyć 0)*

False *(można w instrukcji warunkowej zamiennie użyć 0)*

INNE OPERACJE NA TEKSTACH

`a='Beata lubi wiosnę i lato'`

`print(a.replace('a','$'))` *(uzyskamy Be\$t\$t lubi wiosnę i l\$to)*

`print('*'.join(a))` *(uzyskamy B*e*a*t*a* *l*u*b*i* *w*i*o*s*n*ę* *i* *l*a*t*o)*

`print(a.isalpha())` *(uzyskamy False ponieważ są tam spacje nie będące znakami alfabetu)*

`print(a.replace(' ','').isalpha())` *(True ponieważ usunęliśmy spacje , a reszta znaków należy do alfabetu)*

`print(ord('A'))` *(uzyskamy 65- czyli odpowiednik w kodach ASCII znaku „A”)*

`print(chr(65))` *(uzyskamy A- czyli odpowiednik w ASCII kodu 65)*

Półśłówka

Kajtek jadąc pociągiem ze swoim braciszkiem lubi z nim grać w półśłówka. Polega to na tym, że zapisuje dwa słowa, najpierw długie a potem krótsze... I jeszcze liczbę naturalną N, mniejszą od długości pierwszego słowa. Jego braciszek musi szybko odpowiedzieć, czy drugie słowo jest fragmentem pierwszego, ale skróconego o N pierwszych liter.

Nie rozumiesz? Gdy Kajtek mówi 'kosodrzewina', 'win' i '9', to z pierwszego słowa zabiera początkowych 9 liter, uzyskuje wówczas słówko 'ina'. I oczywiście słówko 'win' nie jest jego częścią. Gdyby podał liczbę 7 otrzymałby słówko 'ewina' i 'win' jak najbardziej zawierałoby się w nim.

Bracia nie rozróżniają jeszcze za dobrze wielkich i małych liter alfabetu. Ale to im nie przeszkadza dobrze się bawić.

Przykład: Wejście: kosodrzewina win 9 Wyjście: nie	Przykład: Wejście: kosodrzewina win 7 Wyjście: tak	Przykład: Wejście: abrakadabra RaK 1 Wyjście: tak
--	--	---

1. Wprowadzamy zmienne:
a- 1 długie słowo
b- 2 krótkie słowo
N- liczba znaków o które skrócimy 1 słowo

2. Skracamy a- o N znaków
`a[N:]`

3. Sprawdzamy czy b znajduje się w `a[N:]`
`if (b in a[N:])==1:`

4. Zmieniamy teksty na duże litery
`if (b.upper() in a[N:].upper())==1:`

```
a=input()
b=input()
N=int(input())
if (b.upper() in a[N:].upper())==1:
    print('tak')
else:
    print('nie')
```

Słówko z psikusem



Maciuś chce zrobić kajtkowi dowcip. Pozamieniał mu w jego słowie wszystkie literki „a” na „o” oraz „A” na „O”. Jeśli nie znalazł żadnej z tych literek to w złości pisał „A gdzie A?” Jakie słowo zostało po tym psikusie?

Przykład:	Przykład:	Przykład:
Wejście:	Wejście:	Wejście:
Abakadabra	pszczoła	krosno
Wyjście:	Wyjście:	Wyjście:
Obrokodobre	pszczolo	A gdzie A?

1. Wprowadzamy zmienną: a- słowo
`a=input()`

2. Sprawdzamy czy w a znajduje się znak a lub A
`if a.find('a' or "A")>=0:`

4. Jeżeli tak to zamieniamy małe a na małe o,
a potem duże A na duże O
`b=a.replace('a', 'o')`
`print(b.replace('A','O'))`

```
a=input()
if a.find('a' or "A")>=0:
    b=a.replace('a', 'o')
    print(b.replace('A','O'))
else:
    print('A gdzie A?')
```

Następna literka

Kajtek i Maciuś znudzili się poprzednimi grami. Znowu coś nowego wymyślili. Ci chłopcy to studnia pomysłów!

Maciuś podaje trzyliterkowe słowo, a Kajtek wybiera z niego trzecią, ostatnią literkę. Sprawdza jej pozycję w alfabecie i podaje jaką literka w alfabecie jest następna. Jeśli trafi na ostatnią w alfabecie, to podaje pierwszą.

Tym razem chłopcy używają tylko małych literek alfabetu pozbawionego „polskich ogonków”.

Przykład: Wejście: kos Wyjście: t	Przykład: Wejście: woz Wyjście: a	Przykład: Wejście: kra Wyjście: b
---	---	---

1. Wprowadzamy zmienną:
a- słowo

2. Sprawdzamy, czy słowo kończy się na 'z', bo wówczas kolejna literka jest a: ,
if a[len(a)-1]=='z':
(len(a)-1 ponieważ 1 znak jest na pozycji 0 czyli np. kowal : długość =5 ale ostatni znak jest na pozycji 4)

3. W innym wypadku będzie to kolejna litera z alfabetu
ord(a[len(a)-1])-uzyskujemy nr kodu Ascii ostatniej litery ze słowa
chr(ord(a[len(a)-1])+1)- zamieniamy kod ASCII kolejnej litery- na literę

```
a=input()
```

```
if a[len(a)-1]=='z':  
    print('a')  
else:  
    print(chr(ord(a[len(a)-1])+1))
```