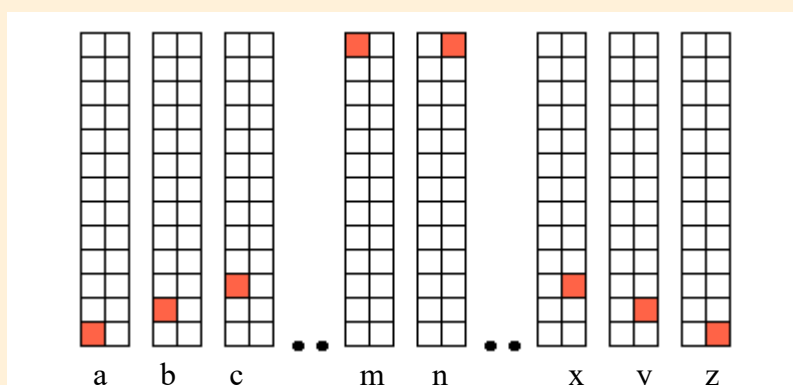


Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Kwadratowy szyfr – LOGIA 20 (2019/20), etap 2

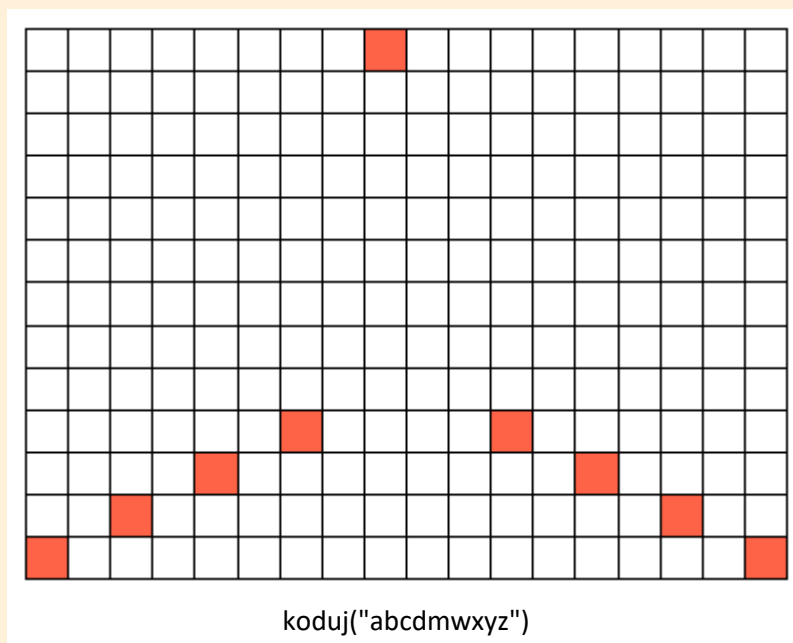
Treść zadania

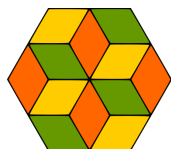
Kolejnym literom alfabetu łacińskiego (jest ich 26) odpowiadają rysunki przedstawione poniżej:



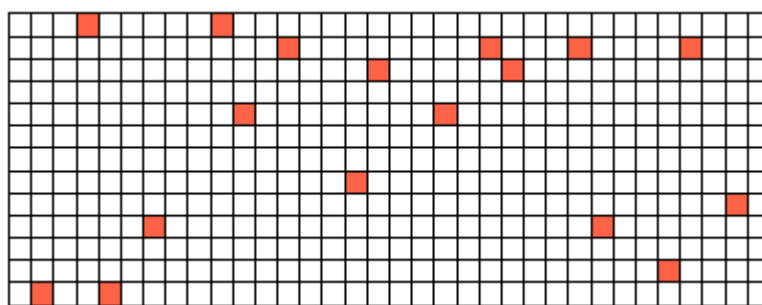
Napisz funkcję **koduj(napis)**, której parametrem jest napis składający się z małych liter alfabetu łacińskiego o długości od **9** do **40** liter. Po jej wywołaniu na środku ekranu powstanie rysunek zakodowanego napisu. Szerokość rysunku wynosi **760**.

Przykłady:





Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty



koduj("znadnilukrokodyle")

Omówienie rozwiązania

Każda litera jest zaszyfrowana 26 kwadratami, z których jeden jest zamalowany. Do narysowania takich kwadratów potrzebna jest znajomość alfabetu łacińskiego, który składa się z 26 liter. Warto go wygenerować z pomocą kodów ASCII, żeby nie pominąć żadnej litery oraz nie pomylić kolejności liter.

Można to zrobić wstawiając do tablicy kolejne litery począwszy od kodu ASCII 97, który odpowiada literze "a".

```
1. tab = []
2. for i in range(26):
3.     tab += chr(i + 97)
```

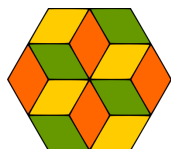
Rysowanie zakodowanej litery, to narysowanie dwóch kolumn kwadratów, po 13 kwadratów w każdej z nich. Jeden z kwadratów jest zamalowany kolorem czerwonym. Dla litery "a" jest to pierwszy dolny kwadrat lewej kolumny, dla litery "b" drugi itd. Ostatnią literą z zamalowanym kwadratem w pierwszej kolumnie jest "m". Kolejne litery od "n" do "z" zamalowane są w drugiej kolumnie kwadratów.

```
1. def rysuj(litera, bok):
2.     tab = alfabet()
3.     for i in range(26):
4.         if litera != tab[i]:
5.             kwadrat(bok)
6.         else:
7.             kwadrat_cz(bok)
8.             fd(bok)
9.             if i == 12:
10.                 rt(90); fd(2 * bok); rt(90)
```

Funkcja rysująca zakodowaną literę była najtrudniejszą częścią zadania. Samo kodowanie danego napisu jest proste, wystarczy przeglądać napis dany jako parametr znak po znaku i rysować kwadraty odpowiadające odpowiedniej literze.

Przy tworzeniu rysunku należy uwzględnić skalowanie. Ponieważ szerokość rysunku wynosi 760, można obliczyć długość boku kwadratu:

```
1. bok = 760 / (2 * len(napis))
```



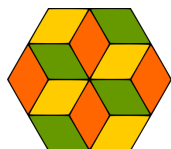
Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rysunek musi być także na środku ekranu, dlatego na początku funkcji głównej należy żółwia przenieść w odpowiednie miejsce. Zakodowany wyraz jest prostokątem o szerokości 760 i wysokości równej trzynastu długościom boku pojedynczego kwadratu.

```
1. def koduj(napis):
2.     bok = 760 / (2 * len(napis))
3.     pu(); bk(760 / 2); lt(90); bk(13 * bok / 2); pd()
4.     for zn in napis:
5.         rysuj(zn, bok)
6.         lt(180)
```

Rozwiązanie w języku Python

```
1. from turtle import *
2.
3. def alfabet():
4.     tab = []
5.     for i in range(26):
6.         tab += chr(i + 97)
7.     return tab
8.
9. def kwadrat(bok):
10.    for i in range(4):
11.        fd(bok); rt(90)
12.
13. def kwadrat_cz(bok):
14.    fillcolor("red")
15.    begin_fill()
16.    kwadrat(bok)
17.    end_fill()
18.
19. def rysuj(litera, bok):
20.    tab = alfabet()
21.    for i in range(26):
22.        if litera != tab[i]:
23.            kwadrat(bok)
24.        else:
25.            kwadrat_cz(bok)
26.        fd(bok)
27.        if i == 12:
28.            rt(90); fd(2 * bok); rt(90)
29.
30. def koduj(napis):
31.     bok = 760 / (2 * len(napis))
32.     pu(); bk(760 / 2); lt(90); bk(13 * bok / 2); pd()
33.     for zn in napis:
34.         rysuj(zn, bok)
35.         lt(180)
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Testy

Podczas samodzielnego rozwiązywania zadania należy sprawdzać program nie tylko dla przykładów podanych w treści zadania, ale testować rozwiązania dokładnie.

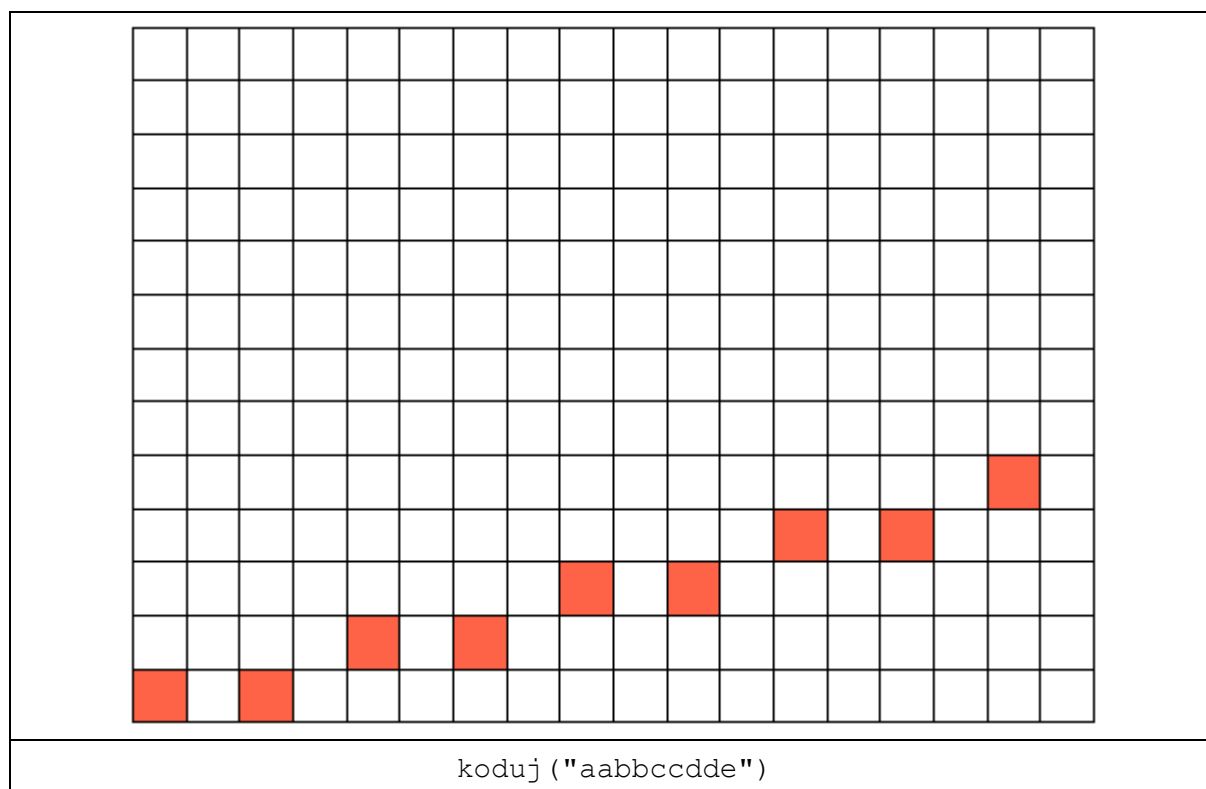
Warto sprawdzić poprawność rysowania wszystkich 26 liter alfabetu. W tym celu można uruchomić dwa niezależne testy dla pierwszych 13-tu liter alfabetu i 13-tu ostatnich.

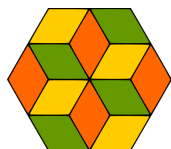
1. `koduj("abcdefghijklm")`
2. `koduj("nopqrstuvwxyz")`

Następnie sprawdzamy wyśrodkowanie rysunku.

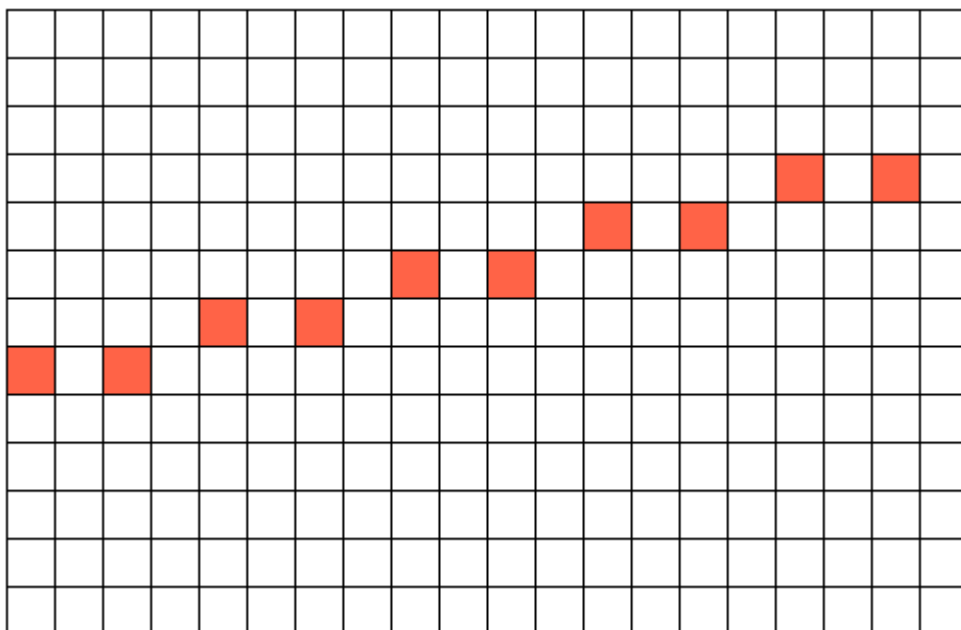
1. `tracer(0)`
2. `koduj("abcdefghj")`
3. `pu(); home();pd()`
4. `update()`

Po zakończeniu rysowania żółtów znajdzie się w środku ekranu, który powinien pokrywać się ze środkiem rysunku. Takie testowanie warto przeprowadzić dla kilku wartości parametru, na przykład dla słowa długości 9 (minimalna długość według treści zadania), dla słowa o parzystej i nieparzystej długości oraz dla słowa o długości 40 (maksymalna długość).

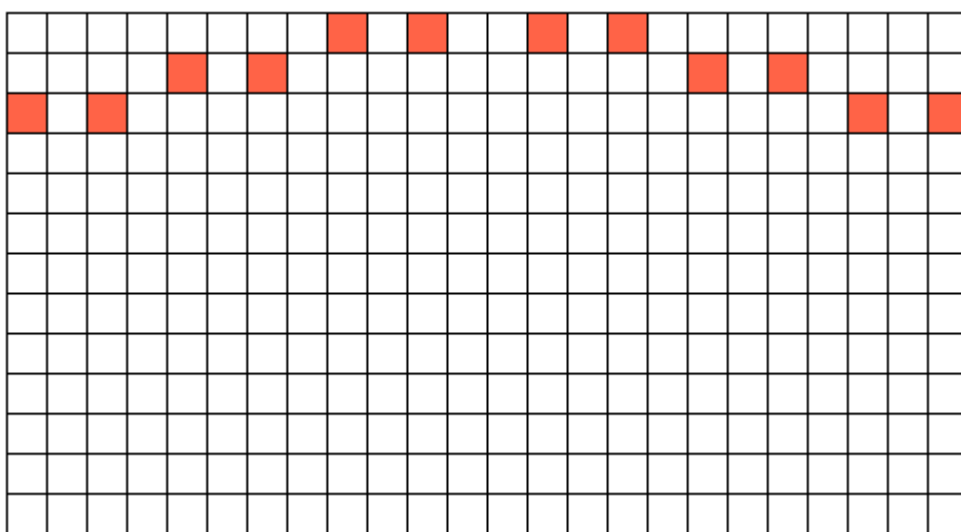




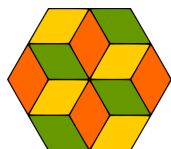
Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty



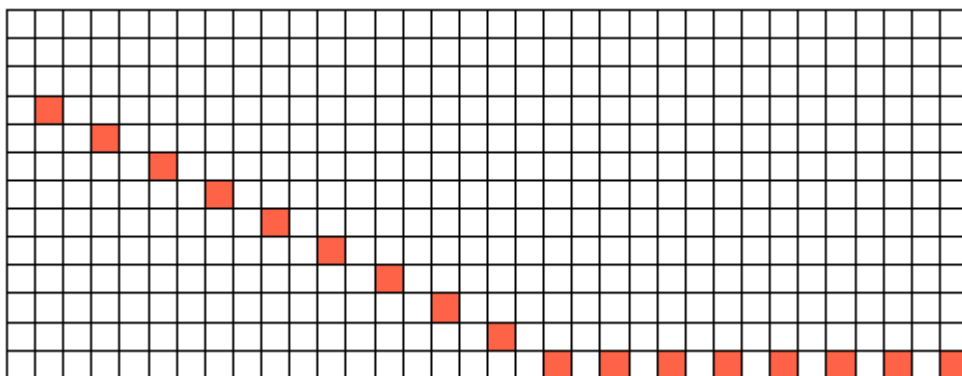
`kodej ("ffgghhiijj")`



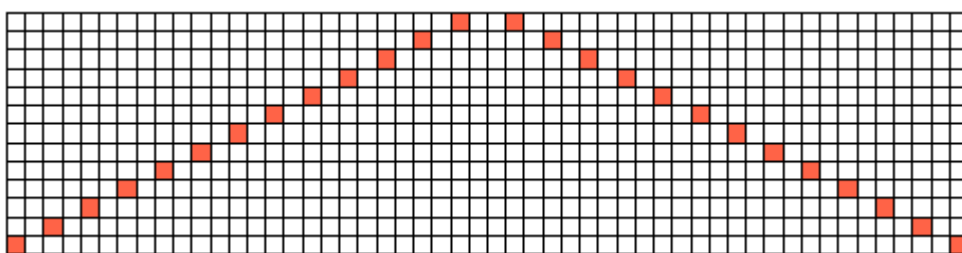
`kodej ("kkllmmnnopp")`



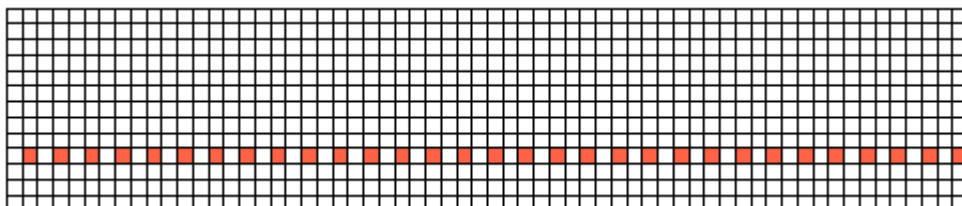
Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty



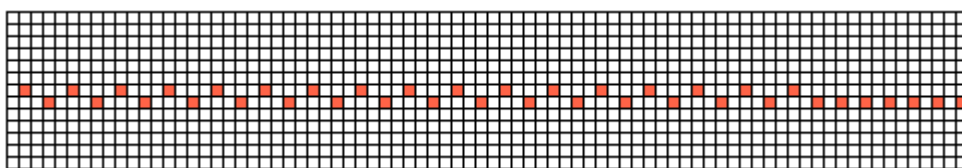
kodej ("qrstuvwxyzzzzzzzzz")



kodej ("abcdefghijklmnopqrstuvwxy")



kodej ("wwwwwwwwwwwwwwwwwwwwwwwwww")



kodej ("tutututututututututututututututuuuuuu")