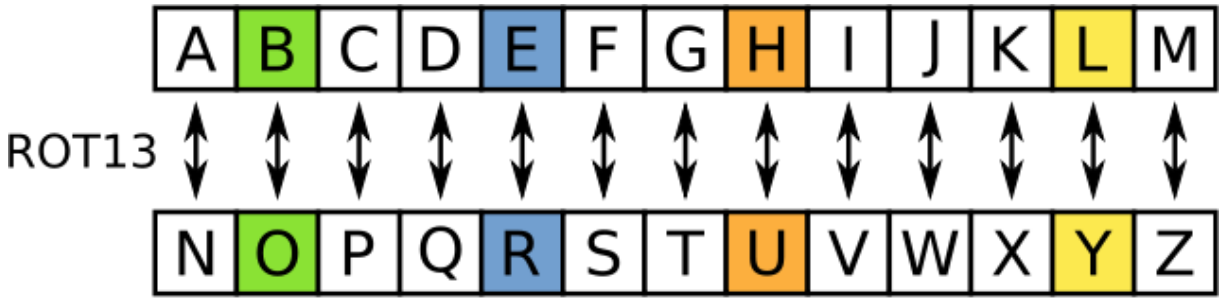
Algorytm szyfru rot-13

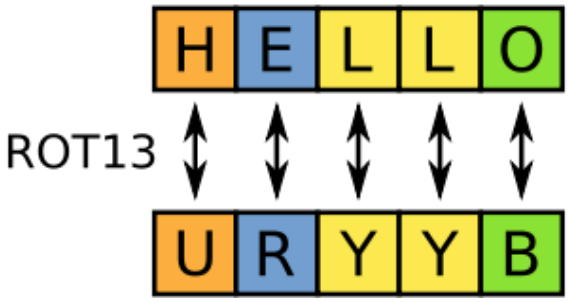
Szyfr rot-13 to szyfr przesuwny, w którym klucz wynosi 13. Jest podobny do szyfru Cezara (tam klucz wynosi 3).

Początkowo jego głównym celem było omijanie filtrów blokujących niecenzuralne zwroty. W ten sposób można było pisać wypowiedzi, które odczytywały tylko te osoby, które tego sobie życzyły. Oprócz tego tak szyfruje się puenty żartów, rozwiązania zagadek.

Szyfr rot-13

W *rot-13* każdą literę alfabetu zamieniamy na literę znajdującą się 13 pozycji dalej.





Zamiana wygląda następująco: A → N, B → O, C → P i jednocześnie N → A, O → B, P → C.

Właściwości szyfru rot-13

Jest to rodzaj szyfru podstawieniowego, w którym każda litera tekstu jawnego (niezaszyfrowanego) zastępowana jest inną, oddaloną o stałą liczbę pozycji w alfabecie, literą (szyfr monoalfabetyczny), przy czym kierunek zamiany musi być zachowany. Nie rozróżnia się przy tym liter dużych i małych.

W standardowym alfabecie łacińskim (26 znaków) ROT13 jest swoją własną funkcją odwrotną. Ten sam algorytm wykorzystywany jest do szyfrowania i deszyfrowania wiadomości. Złożenie dwóch operacji szyfrowania algorytmem rot-13 da z powrotem tekst jawny.

Kodowanie w językach programowania

Szyfr rot-13 jest prosty i łatwym do implementacji algorytmem. Można skorzystać z kilku sposobów implementacji:

Użycie tablicy

Postać algorytmu:

Algorytm korzysta z tablicy 1-wymiarowej mającej liczbę kolumn zależną od długości alfabetu danego języka (26 dla alfabetu łacińskiego).

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Przy szyfrowaniu kolejnych znaków tekstu jawnego, należy przesunąć indeks tablicy o 13, i zawinąć go dzieleniem modulo.

```
i = (i + 13) mod 26
```

Zaletą jest prostota i elegancja algorytmu. Łatwo zmodyfikować ustawienia znaków w tablicy. Można też dodać inne znaki do podstawowego alfabetu.

Kodowanie ASCII

W tym algorytmie wykorzystuje się sposób kodowania liter, cyfr i znaków interpunkcyjnych za pomocą liczb. Podstawowy sposób kodowania nazywa się ASCII i obejmuje zakres od 0 do 127. W tej części są zawarty duże (od 65 do 90) i małe litery (od 97 do 122).

Działanie algorytmu:

Przy szyfrowaniu kolejnych znaków należy obliczyć ich kod ASCII. `ord (x)`

Następnie dana liczba jest zwiększana o 13. `ord (x) + 13`

Potem wystarczy zamienić to na znak i litera jest zaszyfrowana. `char (ord (x) + 13)`

Zapętlenie algorytmu:

Niestety pojawia się problem z ostatnimi literami. Po dodaniu do nich 13, nie pokazują początku alfabetu, ale znaki znajdujące się dalej. Konieczne jest zapętlenie i zmuszenie algorytmu, by po dojściu do ostatniej litery, wracał do początku alfabetu.

Do tego celu służy dzielenie modulo o długości równej ilości znaków alfabetu.

```
char ((ord (x) + 13) mod 26)
```

Konieczne jest również przesunięcie całego zbioru liter w dolną część tablicy ASCII, a potem powrót do poprzedniego zakresu. Dla małych liter (z zakresu 97- 122) będzie to wyglądać następująco:

```
char (((ord (x)- 97 + 13) mod 26)+ 97)
```

Algorytm deszyfrujący:

Rot-13 jest również swoją własną funkcją odwrotną. Algorytm szyfrujący jest jednocześnie algorytmem deszyfrującym.

```
char (((ord (x)- 97 + 13) mod 26)+ 97)
```

Ciekawostki szyfru rot-13:

Dla niektórych wyrażeń ROT13 nie spełnia swojego zadania, ponieważ zakodowane słowa przybierają formę innych słów w tekście oryginalnym (niezakodowanym). W skrajnym przypadku może dojść jedynie do zamiany słów miejscami. Przykładowo polski tekst "`ten hejnal urwany gra`" po zakodowaniu ma postać "`gra urwany hejnal ten`".

Ćwiczenia

1. Używając szyfru rot-13 napisz program:
    a. szyfrujący i deszyfrujący wiadomość
2. Sprawdź czy przesunięcie klucza w lewo (zamiast w prawo) da takie same efekty.
3. Zaproponuj wersję szyfru dla alfabetu polskiego (wersja z 32 literami – bez Q, X, V)
4. Wyszukaj jakie tekst po zaszyfrowaniu nie zostaną dobrze zakodowane.