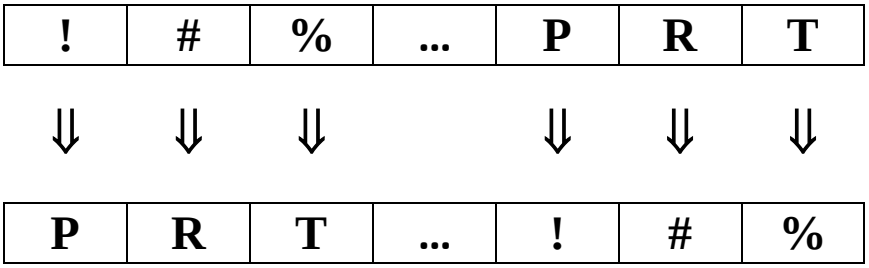
Algorytm szyfru rot-47

Szyfr rot-47 to szyfr przesuwny, w którym klucz wynosi 47. Jest podobny do szyfru Cezara (tam klucz wynosi 3).

ROT47 zamienia każdy widoczny znak ASCII z przedziału 33-126 na znak znajdujący się 47 pozycji dalej, ale nie dalej niż do 126 pozycji.

Szyfr rot-47

W *rot-47* każdy znak ASCII zamieniamy na znak znajdujący się 47 pozycji dalej.



Zamiana wygląda następująco: ! → P, # → R, % → T i jednocześnie P → !, R → #, T → %.

Właściwości szyfru rot-47

Jest to rodzaj szyfru podstawieniowego, w którym każdy znak kodu ASCII tekstu jawnego (niezaszyfrowanego) zastępowany jest innym, oddalonym o stałą liczbę pozycji w tabeli, przy czym kierunek zamiany musi być zachowany.

ROT47 jest swoją własną funkcją odwrotną. Ten sam algorytm wykorzystywany jest do szyfrowania i deszyfrowania wiadomości. Złożenie dwóch operacji szyfrowania algorytmem rot-47 da z powrotem tekst jawny.

Kodowanie w językach programowania

Szyfr rot-47 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 równą ilości widocznych znaków ASCII (od znaku 33 do 126).

!	"	#	$	%	&	'	(	)	....	x	y	z	{	|	}	~

Przy szyfrowaniu kolejnych znaków tekstu jawnego, należy przesunąć indeks tablicy o 47, i zawinąć go dzieleniem modulo.

```
i = (i + 47) mod 94
```

Kodowanie ASCII

W kodowaniu ASCII pierwsze 31 znaków to znaki sterujące. Są przeznaczone do sterowania urządzeniem odbierającym dane. Na przykład, znak 10 (LF) oznaczający przejście do nowej linii,

powoduje przesunięcie papieru w drukarce, a znak 8, czyli Backspace powodował cofnięcie karetki o jedno pole. Znak 32 to spacja (blank). Znak 127 to delete.

Znaki drukowalne mają numery od 33 do 126. Jest ich więc 94.

Działanie algorytmu:

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

Następnie dana liczba jest zwiększana o 13. `ord (x) + 47`

Potem wystarczy zamienić to na znak i jest zaszyfrowany. `char (ord (x) + 47)`

Konieczne jest zapętlenie i zmuszenie algorytmu, by po dojściu do ostatniego znaku, wracał do znaku z początku zakresu. `char ((ord (x) + 47) mod 94)`

Żeby uniknąć drukowania znaków sterujących należy zakres przesunąć w dół, a po podzieleniu modulo wrócić z powrotem `char ((ord (x)-33 + 47) mod 94)+33`

Algorytm deszyfrujący:

Rot-47 jest również swoją własną funkcją odwrotną. Algorytm szyfrujący jest jednocześnie algorytmem deszyfrującym.

Ćwiczenia

1. Używając szyfru rot-47 napisz program:
    a. szyfrujący i deszyfrujący wiadomość
2. Sprawdź czy przesunięcie klucza w lewo (zamiast w prawo) da takie same efekty.