

C++

szyfrowanie i deszyfrowanie (manipulatory)

studia dzienne

Zaimplementuj manipulator `encrypt(k)`, który zapewnia, że wyjście na strumień `ostream` będzie kodowane za pomocą klucza `k`. Podobnie zaimplementuj manipulator `decrypt(k)` dla strumienia `istream`, który będzie odkodowywał tekst zakodowany wcześniej kluczem `k`. Zdefiniuj również manipulatory bezparametrowe `noencrypt` i `nodecrypt`, wyłączające kodowanie/dekodowanie strumienia, tak by dalsze wyjście/wejście dotyczyło czystego tekstu.

Kodowanie i dekodowanie ma dotyczyć strumieni znakowych. Kodowane/dekodowane powinny być tylko znaki o kodach z zakresu `32...126`. Parametr `texttk` powinien być liczbą naturalną nie większą niż 95. Jako funkcję kodującą znak o kodzie `z` przyjmij $c(z)$:

$$c(z) = \begin{cases} ((z - 32 + k) \bmod 95) + 32 & \text{dla } z \in \{32 \dots 126\} \\ z & \text{dla } z \notin \{32 \dots 126\} \end{cases}$$

Funkcja dekodująca $d(z)$ ma przywracać pierwotną postać znku.

Następnie napisz program, który czyta dane linia po linii ze standardowego wejścia, ewentualnie koduje tekst i posyła go pliku `szyfr.txt`. Sygnałem do rozpoczęcia/zakończenia kodowania jest linia zawierająca pojedynczy znak kropki. Klucz `k` przekaz do programu poprzez parametry wywołania. Sprawdź poprawność szyfru analogicznym programem dekodującym.