

Akademia Nauk Stosowanych - Teoretyczne i technologiczne podstawy multimediiów - laboratorium			
Temat: Szyfr Cezara.			Symbol: TiTPM
Nazwisko i imię: Fyda Kamil		Ocena sprawozdania	Zaliczenie:
Data wykonania ćwiczenia: 4.10.2022r.	Grupa: L1		

Szyfr Cezara – jedna z najprostszych technik szyfrowania. Jest to rodzaj szyfru podstawieniowego, w którym każda litera tekstu jawnego zastępowana jest inną, oddaloną od niej o stałą liczbę pozycji w alfabecie, literą, przy czym kierunek zamiany musi być zachowany. Nie rozróżnia się przy tym liter dużych i małych. Algorytm ten polega na zaszyfrowaniu ciągu liter w taki sposób, że przesuwa każdą literę o zadana wartość czyli w naszym wypadku klucz szyfrujący obejmuje od -26 do 26.


Kod programu:

```

1  #include <iostream>
2  using namespace std;
3
4  string szyfrowanie(string tekst, int klucz)
5  {
6      string wynik = "";
7
8      for (int i = 0; i < tekst.length(); i++) {
9          if (isupper(tekst[i]))
10             {
11                 if (int(tekst[i] + klucz) < 65)
12                 {
13                     wynik = wynik + char(int(tekst[i] + klucz - 64) % 26 + 90);
14                 }
15                 else
16                 {
17                     wynik = wynik + char(int(tekst[i] + klucz - 65) % 26 + 65);
18                 }
19             }
20             else
21             {
22                 if (int(tekst[i] + klucz) < 97)
23                 {
24                     wynik = wynik + char(int(tekst[i] + klucz - 96) % 26 + 122);
25                 }
26                 else
27                 {
28                     wynik = wynik + char(int(tekst[i] + klucz - 97) % 26 + 97);
29                 }
30             }
31         }
32     }
33     return wynik;
34 }
35
36
37 int main()
38 {
39     string tekst;
40     cout << "Tutaj wpisz przykładowy tekst: " << endl;
41
42     cin >> tekst;
43
44     int klucz;
45     cout << "Tutaj wprowadz klucz do szyfrowania: " << endl;
46     cin >> klucz;
47
48     cout << "Tutaj dostajesz swój zaszyfrowany tekst: " << szyfrowanie(tekst, klucz) << endl;
49     return 0;
50 }


```

Wynik działania programu:

 C:\Users\Asus\Desktop\TiTPM_31003\TiTPM01.exe

```
Tutaj wpisz przykładowy tekst:
KamilFydatoPisze
Tutaj wprowadz klucz do szyfrowania:
4
Tutaj dostajesz swój zaszyfrowany tekst: OeqmpJchexsTmwdi

-----
Process exited after 22.79 seconds with return value 0
Press any key to continue . . .
```

 C:\Users\Asus\Desktop\TiTPM_31003\TiTPM01.exe

```
Tutaj wpisz przykładowy tekst:
KamilFydatoPisze
Tutaj wprowadz klucz do szyfrowania:
-5
Tutaj dostajesz swój zaszyfrowany tekst: FvhdgAtyvojKdnuz

-----
Process exited after 12.3 seconds with return value 0
Press any key to continue . . .
```

Opis teoretyczny:

Na początku musimy ustalić długość naszego kodu, przedział szyfru musi obejmować liczby od -26 do 26, jeśli nie mieści się w tych granicach to program kończy działanie. Jeżeli klucz jest prawidłowy to następuje dodanie przesunięcia każdej litery z osobna o zadaną wcześniej wartość klucza. Jeśli mamy dodatni klucz to zabezpieczamy go przed wyjściem poza ostatnią literę alfabetu (z) a jeżeli jest ujemny to ograniczamy go pierwszą literą (a). Jeżeli wyjdziemy poza zakres to wracamy do przeciwnej strony zakresu alfabetu.

Co wykonywałem po kolei:

1. Wprowadziłem tekst składający się z liter o dowolnej wielkości,
2. Zadeklarowałem klucz, który odpowiada za przesunięcie znaków,
3. Wprowadziłem dane,
4. Wykonało się szyfrowanie tekstu,
5. Odpowiednie funkcje if oraz else sprawdzają czy jest duża czy mała litera,
6. Na końcu otrzymałem zadowalający mnie efekt (co warto zaznaczyć użyłem kodu ASCII).

Wniosek:

Algorytm szyfrowania zastosowany w kodzie Cezara bywa fragmentem bardziej złożonych systemów szyfrowania, takich jak szyfr Vigenère'a. Współcześnie szyfru Cezara używa się z przesunięciem 13 (ROT13), będącego prostym i szybkim sposobem na ukrycie treści. Obecnie szyfr Cezara, jak każda technika podmieniająca pojedyncze litery alfabetu na inne, nie oferuje żadnego bezpieczeństwa komunikacji.