# **Лабораторная работа 2**2

#### ***2***2***.1 [#20]***

Модифицируйте программу Encryption так, чтобы она шифровала пробелы между словами.   
Вместо заполнения остатка строки Msg пробелами отслеживайте длину входной строки   
с помощью переменной типа 1..Len

#### ***2***2***.2 [#30]***

Модифицируйте программу Encryption так, чтобы она считывала шифр из файла.

Формат файла определите самостоятельно.

*Учтите, что в файле шифра могут быть неправильные данные (может не хватать какого-то символа, а может быть символ, выходящий за пределы массива). Обрабатывайте эти ситуации корректно.*

#### ***2***2***.3 [#20]***

Напишите программу Decryption, которая расшифровывает зашифрованные программой   
Encryption сообщения, считывая шифр из файла в том же формате, что в и задании 22.2.

PROGRAM Encryption(INPUT, OUTPUT);

{Переводит символы из INPUT в код согласно Chiper

и печатает новые символы в OUTPUT}

CONST

Len = 20;

TYPE

Str = ARRAY [1 .. Len] OF 'A' .. 'Z';

Chiper = ARRAY ['A' .. 'Z'] OF CHAR;

VAR

Msg: Str;

Code: Chiper;

PROCEDURE Initialize(VAR Code: Chiper);

{Присвоить Code шифр замены}

BEGIN {Initialize}

Code['A'] := 'Z';

Code['B'] := 'Y';

Code['C'] := 'X';

Code['D'] := '#';

Code['E'] := 'V';

Code['F'] := 'U';

Code['G'] := 'T';

Code['H'] := 'S';

Code['I'] := 'I';

Code['J'] := 'Q';

Code['K'] := 'P';

Code['L'] := '!';

Code['M'] := 'N';

Code['N'] := 'M';

Code['O'] := '2';

Code['P'] := 'K';

Code['Q'] := '$';

Code['R'] := 'D';

Code['S'] := 'H';

Code['T'] := '\*';

Code['U'] := 'F';

Code['V'] := 'E';

Code['W'] := 'T';

Code['X'] := 'C';

Code['Y'] := 'B';

Code['Z'] := 'A';

END; {Initialize}

PROCEDURE Encode(VAR S: Str);

{Выводит символы из Code, соответствующие символам из S}

VAR

Index: 1 .. Len;

BEGIN {Encode}

FOR Index := 1 TO Len

DO

IF S[Index] IN ['A' .. 'Z']

THEN

WRITE(Code[S[Index]])

ELSE

WRITE(S[Index]);

WRITELN

END; {Encode}

BEGIN {Encryption}

{Инициализировать Code}

Initialize(Code);

WHILE NOT EOF

DO

BEGIN

{читать строку в Msg и распечатать ее}

I := 0;

WHILE NOT EOLN AND (I < Len)

DO

BEGIN

I := I + 1;

READ(Msg[I]);

WRITE(Msg[I])

END;

READLN;

WRITELN;

{дополнить Msg пробелами}

FOR I := I+1 TO Len

DO

BEGIN

Msg[I] := ' ';

END

{распечатать кодированное сообщение}

Encode(Msg)

END

END. {Encryption}