**Код программы:**

#include <stdio.h>

#include <locale.h>

#define FileIn "InputFile.txt"//Файл с входными данными

//Файл с закодированным текстом

#define FileEncrip "EncriptedFile.txt"

//Выходной файл с раскодированным текстом

#define FileDecrrip "DecriptedFile.txt"

const unsigned char BigKirill[33] = { 'А','Б','В','Г','Д','Е','Ё','Ж','З','И','Й','К','Л','М','Н','О','П','Р','С','Т','У','Ф','Х',

'Ц','Ч','Ш','Щ','Ъ','Ы','Ь','Э','Ю','Я' };

const unsigned char SmallKirill[] = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя";

int Encryption(int shift); //Функция кодирования текста

int Decryption(int shift); //Функция декодирования текста

int main()

{

setlocale(LC\_ALL, "Rus");

int shift; //Переменная сдвига

printf("Enter how many positions to shift \n");

//scanf("%d", &shift); //Ввод сдвига

if (scanf("%d", &shift) == 0)

{

printf("Incorrect shift\n");

return 0;

}

if (shift < 0)

{

printf("Incorrect shift\n");

return 0;

}

//Провекри на выходные данные функций

if (Encryption(shift) != 0)

{

printf("Error of Encryption!\n");

return 0;

}

printf("Decreption of file \n");

if (Decryption(shift) != 0)

{

printf("Error of Decryption!\n");

return 0;

}

printf("%c", '\n');

return 0;

}

//Функция кодирования текста

int Encryption(int shift)

{

FILE\* in;

FILE\* out;

in = fopen(FileIn, "r");

out = fopen(FileEncrip, "w");

unsigned char\* str;

str = (unsigned char\*)malloc(256);

long size;

int flag;

int i = 0;

int j = 0;

int NewPos;

if (out == NULL)

{

perror("File not found");

return 1;

}

if (in != NULL)

{

if (fseek(in, 0, SEEK\_END) != 0)

{

printf("Error of displacement in file");

fclose(in);

fclose(out);

remove(out);

return 1;

}

size = ftell(in); //Поиск размера файла

if( fseek(in, 0, SEEK\_SET) != 0)

{

printf("Error of displacement in file");

fclose(in);

fclose(out);

remove(out);

return 1;

}

if (size == 0)

{

fclose(in);

fclose(out);

remove(out);

printf("File is empty\n");

return 1;

}

while (fgets(str, 256, in) != NULL)

{

i = 0;

while ((str[i] != '\0')) //Чтение по строкам

{

flag = 0;

for (j = 0; j < 33; j++) //КИРИЛЛИЦА

{

if (str[i] == SmallKirill[j])

{

str[i] = BigKirill[j];

}

if (str[i] == BigKirill[j])

{

NewPos = j + (shift % 33);

if (NewPos > 32)

{

while (NewPos > 32)

{

NewPos -= 32;

}

NewPos -= 1;

}

str[i] = BigKirill[NewPos];

fprintf(out, "%c", str[i]);

flag = 1;

break;

}

}

/\*Проверка на соответсвие прописному символу

соответствующего диапазона символов с кодами ASCII\*/

if (str[i] >= 'A' && str[i] <= 'Z')

{

str[i] = str[i] + (shift % 26);

if (str[i] > 'Z')

{

str[i] = 'A' + (str[i] - 'Z') - 1;

}

// printf("%d " , (int)str[i]);

fprintf(out, "%c", str[i]);

flag = 1;

}

/\*Проверка на соответсвие строчному символу

соответствующего диапазона символов с кодами ASCII\*/

if (str[i] >= 'a' && str[i] <= 'z')

{

str[i] = str[i] + (shift % 26);

if (str[i] > 'z')

{

str[i] = 'a' + (str[i] - 'z') - 1;

}

fprintf(out, "%c", str[i]);

flag = 1;

}

else if (flag == 0)//Если в строке символ - не буква

{

fprintf(out, "%c", str[i]);

}

i++;

}

}

if (!feof(in))

{

fclose(in);

fclose(out);

remove(out);

printf("error of reading\n");

return 1;

}

fclose(in);

fclose(out);

}

else

{

perror("File not found");

return 1;

}

return 0;

}

int Decryption(int shift) //Функцияя декодирования текста

{

FILE\* in;

FILE\* out;

in = fopen(FileEncrip, "r");

out = fopen(FileDecrrip, "w");

unsigned char str[256];

long size;

int flag;

int i = 0;

int j = 0;

int NewPos;

printf("%s - дешифрование \n", FileEncrip);

if (out == NULL)

{

perror("File not

found");

return 2;

}

if (in != NULL)

{

if (fseek(in, 0, SEEK\_END) != 0)

{

printf("Error of displacement in file");

fclose(in);

fclose(out);

remove(out);

return 2;

}

size = ftell(in);

if (fseek(in, 0, SEEK\_SET) != 0)

{

printf("Error of displacement in file");

fclose(in);

fclose(out);

remove(out);

return 2;

}

if (size == 0)

{

fclose(in);

fclose(out);

remove(out);

printf("File is empty\n");

return 2;

}

while (fgets(str, 256, in) != NULL)

{

i = 0;

while ((str[i] != '\0'))

{

flag = 0;

if (str[i] >= 'A' && str[i] <= 'Z')

{

str[i] = str[i] - (shift % 26);

if (str[i] < 'A')

{

str[i] = 'Z' - ('A' - str[i]) + 1;

}

printf("%c", str[i]);

fprintf(out, "%c", str[i]);

flag = 1;

}

else if (str[i] >= 'a' && str[i] <= 'z')

{

str[i] = str[i] - (shift % 26);

if (str[i] < 'a')

{

str[i] = 'z' - ('a' - str[i]) + 1;

}

printf("%c", str[i]);

fprintf(out, "%c", str[i]);

flag = 1;

}

else

{

for (j = 0; j < 33; j++) //КИРИЛЛИЦА

{

if (str[i] == BigKirill[j])

{

NewPos = j - (shift % 33);

if (NewPos < 0)

{

while (NewPos < 0)

{

NewPos += 32;

}

NewPos += 1;

}

str[i] = BigKirill[NewPos];

printf("%c", str[i]);

fprintf(out, "%c", str[i]);

flag = 1;

break;

}

}

}

if (flag == 0)

{

printf("%c", str[i]);

fprintf(out, "%c", str[i]);

}

i++;

}

}

if (!feof(in))

{

fclose(in);

fclose(out);

printf("error of reading\n");

return 2;

}

fclose(in);

fclose(out);

}

else

{

perror("File not found");

return 2;

}

return 0;

}