НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМ. Р.Е. АЛЕКСЕЕВА
Кафедра " Вычислительные системы и технологии"

Программирование

Курсовая работа

Реализация кодировщика\декодировщика на основе структуры Машины Тьюринга.

ВЫПОЛНИЛА СТУДЕНТКА ГРУППЫ 19-ИВТ-2
АШАЕВА А.Н.

«18» ИЮНЯ 2020 Г.

Цели и задачи работы

Нынешнее программирование многогранно и используется в таких важных сферах как строительство, бизнес и экономика, медицина, биология и физика. Большой процент физического труда в промышленности заменен на машинный и роботизированный труд, который управляется посредством программного обеспечения, что обеспечивает существенный прирост скорости, точности операций и эффективности производства. Такое богатство разнообразия применений обеспечивается солидным выбором языков программирования, у каждого из которых есть свои плюсы и минусы.

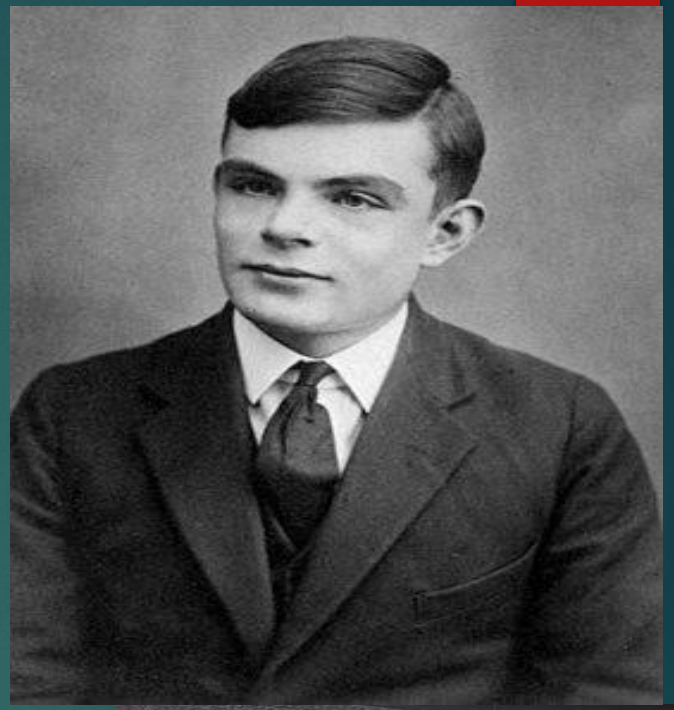
Открытие машины Тьюринга привело к более глубокому познанию цифровых компьютеров и исчислений, включая понимание того, что существуют некоторые вычислительные проблемы, не решаемые на общих пользовательских ЭВМ.

Целью работы является реализация программы кодировщика\декодировщика на основе структуры Машины Тьюринга.

Содержание

Машина Тьюринга и Алан Тьюринг

Концепция абстрактной вычислительной машины была **предложена Аланом Тьюрингом** в 1936 году. Во время Второй мировой войны Тьюринг стал ведущим участником разгадывания шифров немцев. Он работал в Bletchley Park, на станции военного времени GCCS, где сделал пять больших открытий в области криптоанализа, включая разработку электромеханического устройства, используемого в целях расшифровки сигналов шифровальной машины Германии "Enigma".

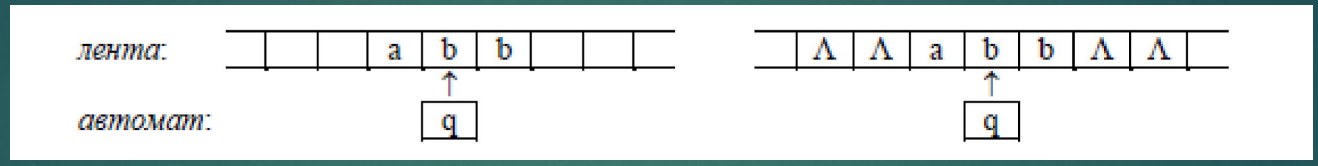


Именно он показал возможность существования универсальной вычислительной машины (машины Тьюринга), способной выполнить любую эффективную процедуру.

Машина Тьюринга — это строгое математическое построение, математический аппарат, созданный для решения определенных задач. Этот математический аппарат был назван “машиной” по той причине, что по описанию его составляющих частей и функционированию он похож на вычислительную машину. Принципиальное отличие машины Тьюринга от вычислительных машин состоит в том, что ее запоминающее устройство представляет собой бесконечную ленту: у реальных вычислительных машин запоминающее устройство может быть как угодно большим, но обязательно конечным. Машину Тьюринга нельзя реализовать именно из-за бесконечности ее ленты. В этом смысле она мощнее любой вычислительной машины.

Структура машины Тьюринга

Машина Тьюринга (МТ) состоит из двух частей – ленты и автомата:



Лента используется для хранения информации. Она бесконечна в обе стороны и разбита на клетки, которые никак не нумеруются и не именуются. В каждой клетке может быть записан один символ или ничего не записано. Содержимое клетки может меняться – в неё можно записать другой символ или стереть находящийся там символ.
Автомат – это активная часть МТ. В каждый момент он размещается под одной из клеток ленты и видит её содержимое; это видимая клетка, а находящийся в ней символ – видимый символ; содержимое же соседних и других клеток автомат не видит. Кроме того, в каждый момент автомат находится в одном из предложенных состояний. Находясь в некотором состоянии, автомат выполняет какую-то определённую операцию (например, перемещается направо по ленте, заменяя все символы b на a), находясь в другом состоянии – другую операцию.
Автомат может выполнять три элементарных действия:
1) записывать в видимую клетку новый символ (менять содержимое других клеток автомат не может);
2) сдвигаться на одну клетку влево или вправо («перепрыгивать» сразу через несколько клеток автомат не может);
3) переходить в новое состояние.
Ничего другого делать автомат не умеет, поэтому все более сложные операции так или иначе должны быть сведены к этим трём элементарным действиям.

Такт работы машины Тьюринга

МТ работает тактами (по шагам), которые выполняются один за другим. На каждом такте автомат МТ выполняет три следующих действия, причем обязательно в указанном порядке:

1) записывает некоторый символ S’ в видимую клетку (в частности, может быть записан тот же символ, что и был в ней, тогда содержимое этой клетки не меняется);

2) сдвигается на одну клетку влево (обозначение – L, от left), либо на одну клетку вправо (обозначение – R, от right), либо остается неподвижным (обозначение – N).

3) переходит в некоторое состояние q’ (в частности, может остаться в прежнем состоянии).
Запись такта для конфигурации называют командой (машины Тьюринга).

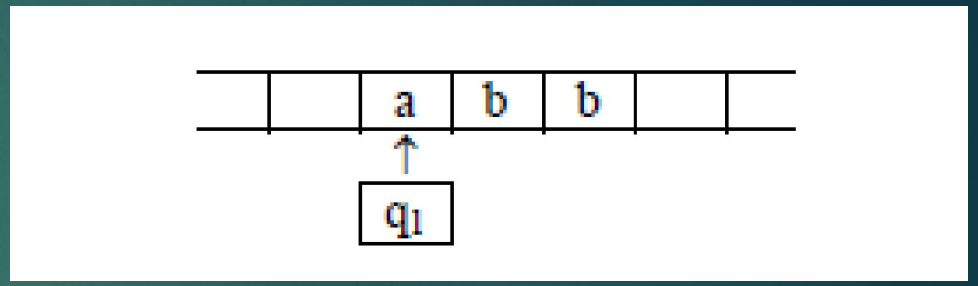
Программа для машины Тьюринга

Сама по себе МТ ничего не делает. Для того чтобы заставить её
работать, надо написать для неё программу. Эта программа
записывается в виде следующей таблицы:
Слева перечисляются все состояния, в которых может находиться автомат, сверху – все символы, которые автомат может видеть на ленте. На пересечениях же (в ячейках таблицы) указываются те такты, которые должен выполнить автомат, когда он находится в соответствующем состоянии и видит на ленте соответствующий символ. В целом таблица определяет действия МТ при всех возможных конфигурациях и тем самым полностью задаёт поведение МТ. Описать алгоритм в виде МТ – значит предъявить такую таблицу.

	S1	S2	…	Si	…	Sn	^
q1							
…							
qi				S,[l,r,n], q			
…							
qm							

Программа для машины Тьюринга

Правила выполнения программы: К началу выполнения программы машина Тьюринга находится в начальной конфигурации. Эта начальная конфигурация определена следующим образом. Во-первых, на ленте записано входное слово, к которому будет применена программа. Входное слово – это конечная последовательность символов, записанных в соседних клетках ленты. Во-вторых, автомат установлен в состояние q1 (указанное в таблице первым) и размещен под его первым (самым левым) символом входного слова:



Если входное слово пустое, то автомат может смотреть в любую клетку, т.к. все они пусты. После этих предварительных действий начинается выполнение программы. В таблице отыскивается ячейка на пересечении первой строки (т.к. автомат находится в состоянии q1) и того столбца, который соответствует первому символу входного слова (это необязательно левый столбец таблицы), и выполняется такт, указанный в этой ячейке.

В результате автомат окажется в новой конфигурации. Теперь такие же действия повторяются, но уже для новой конфигурации: в таблице отыскивается ячейка, соответствующая состоянию и символу этой конфигурации, и выполняется такт из этой ячейки. И так далее. Когда завершается выполнение программы? Введём понятие такта останова. Это такт, который ничего не меняет: автомат записывает в видимую клетку тот же символ, что и был в ней раньше, не сдвигается и остается в прежнем состоянии, т.е. это такт S,N,q для конфигурации (S, q). Попав на такт останова, МТ, по определению, останавливается, завершая свою работу.

Пример

	0	1	^
q1	q2 1 R	q2 0 R	L q0
q2	q1 1 R	q1 0 R	L q0

Данная МТ позволяет заменить все «0» в слове на «1» и наоборот.
Слово: 100011
Запись в ленту: «Λ q1100011 Λ»
Операции:
Λ q1100011 Λ -- Λ 0q200011 Λ -- Λ 01q10011 Λ -- Λ 011q2011 Λ -- Λ 0111q111 Λ -- Λ 01110 q21 Λ -- Λ 011100 q1 Λ -- Λ 01110 q00Λ
Результат: 011100

Описание алгоритма шифрования методом одноалфавитной подстановки

Есть алфавит, состоящий, к примеру, из символов «АБВГДЕ» (при этом важна последовательность символов и они не должны повторяться) и слово, состоящее из символов этого алфавита, например «ГДЕ». Нам необходимо зашифровать слово по некоторому ключу, представляющему собой целое число (для удобства будем брать числа от 1 до 9). Допустим ключ число 2. Тогда каждый символ в слове «ГДЕ» сдвигается на 2 позиции влево относительно соответствующего символа в алфавите и после шифрования представляет собой слово «ЕАБ» (если при смещении символы в алфавите закончились, то отсчет продолжается сначала алфавита). В данном примере, для наглядности, используется небольшой алфавит и однозначный код, но принцип шифрования методом одноалфавитной подстановки полностью соблюдён.

Реализация алгоритма шифрования методом одноалфавитной подстановки при помощи детерминированной машины Тьюринга.

Постановка задачи.

У нас есть алфавит, состоящий из символов «antrkid». Нам необходимо написать программу для ИМТ, позволяющую зашифровать слово «antarktida» по ключу 1 или 2. В данном примере входящим словом для ИМТ будет являться «1antarktida» или «2antarktida» в зависимости от ключа шифрования. На выходе на ленте мы должны получить только зашифрованное слово («ntrnkirdan» и «trktidkant» для ключей 1 и 2 соответственно). «12antrkid» будет являться множеством допустимых входящих символов.

Теперь определимся с состояниями автомата:

а) q1 — автомат определяет, по какому ключу шифруется слово, и переходит в состояние q2 или q3;

б) q2 — автомат шифрует слово по ключу 1;

в) q3 — автомат шифрует слово по ключу 2.

Состояние	Входящие символы									
	1	2	a	n	t	r	k	i	d	B
q1	Bq2R	Bq3R								
q2	nq2R	tq2R	rq2R	kq2R	iq2R	dq2R	aq2R	BSTOPL		
q3	tq3R	rq3R	kq3R	iq3R	dq3R	aq3R	nq3R	BSTOPL		

Проверить работу программы можно в ИМТ, внеся в него следующие команды: 1q1->Bq2R, 2q1->Bq3R, aq2->nq2R, nq2->tq2R, tq2->rq2R, rq2->kq2R, kq2->iq2R, iq2->dq2R, dq2->aq2R, Bq2->BSTOPL, aq3->tq3R, nq3->rq3R, tq3->kq3R, rq3->iq3R, kq3->dq3R, iq3->aq3R, dq3->nq3R, Bq3->BSTOPL.

Описание алгоритма симметричного шифрования методом перестановки.

Данный алгоритм заключается в следующем. У нас есть слово, которое необходимо зашифровать по некоторому ключу. Ключ представляет собой последовательность чисел, первое из которых показывает, какой из символов в исходном слове является первым в зашифрованном, второй показывает, какой из символов в исходном слове является вторым в зашифрованном и т. д. Из этого следует, что длина ключа равна количеству символов в слове. К примеру, у нас есть слово «Привет», которое необходимо зашифровать по ключу «356142». Тогда зашифрованное слово примет вид «иетПвр».

Реализация алгоритма симметричного шифрования методом перестановки при помощи детерминированной машины Тьюринга

Постановка задачи. У нас есть слово «home» его необходимо зашифровать по ключу «3421». Работа МТ выглядит при этом следующим образом. Входным словом является ключ, при этом не обязательно «3421», машина должна работать при любом сочетании этих чисел. По завершении работы на ленте должно остаться только зашифрованное слово (в данном случае слово «meoh»). При этом следует учитывать, что на пути у автомата могут встречаться уже напечатанные символы, которые следует пропускать. Рассмотрим состояния автомата:

а) q1 — автомат определяет, какой символ необходимо напечатать, либо прекращает свою работу, если все символы напечатаны (не осталось символов, составляющих ключ);

б) q2–q5 автомат печатает соответствующий символ;

в) q6 — автомат возвращается в начало слова.

Состояние	Входящие символы								
	1	2	3	4	h	o	m	e	B
q1	Bq2R	Bq3R	Bq4R	Bq5R	hSTOPH	oSTOPH	mSTOPH	eSTOPH	
q2	2q2R	3q2R	4q2R	oq2R	mq2R	eq2R	hq6L		
q3	1q3R	3q3R	4q3R	hq3R	mq3R	eq3R	oq6L		
q4	1q4R	2q4R	4q4R	hq4R	oq4R	eq4R	mq6L		
q5	1q5R	2q5R	3q5R	hq5R	oq5R	mq5R	eq6L		
q6	1q6L	2q6L	3q6L	4q6L	hq6L	oq6L	mq6L	eq6L	Bq1R

При использовании различных ключей, состоящих из символов «1234», будут выдаваться различные зашифрованные слова. Следует отметить, что для реализации шифрования более длинных слов, нужно лишь ввести новые состояния автомата для недостающих символов. А если длина шифруемого слова больше десяти, то ключ следует записывать в системе исчисления, которая позволяет записать каждый номер символа в одной клетке.

Заключение

МТ позволяет в полной мере реализовать простейшие алгоритмы шифрования, однако следует учитывать, что при использовании большого количества входящих символов, требуется вводить дополнительные состояния автомата, что в свою очередь приводит к увеличению размеров программы. Наиболее удобными задачами, решаемыми при помощи МТ, являются задачи обработки символьных последовательностей, к которым можно отнести и описанные выше алгоритмы шифрования.

Задача для реализации кодировщика\декодировщика на основе МТ

Программы кодировщика\декодировщика на основе структуры Машины Тьюринга на английском языке, которая работает в двух режимах: 1 - режим кодирования; 2 – режим декодирования.

Для работы программы нужно:

1. выбрать и ввести режим программы;
2. ввести слово, которое нужно кодировать/декодировать на английском языке;
3. ввести ключ, который имеет входные символы «0», «1», «2», при этом машина должна работать при любом сочетании этих чисел.

Функция Fnull()

Блок-схема

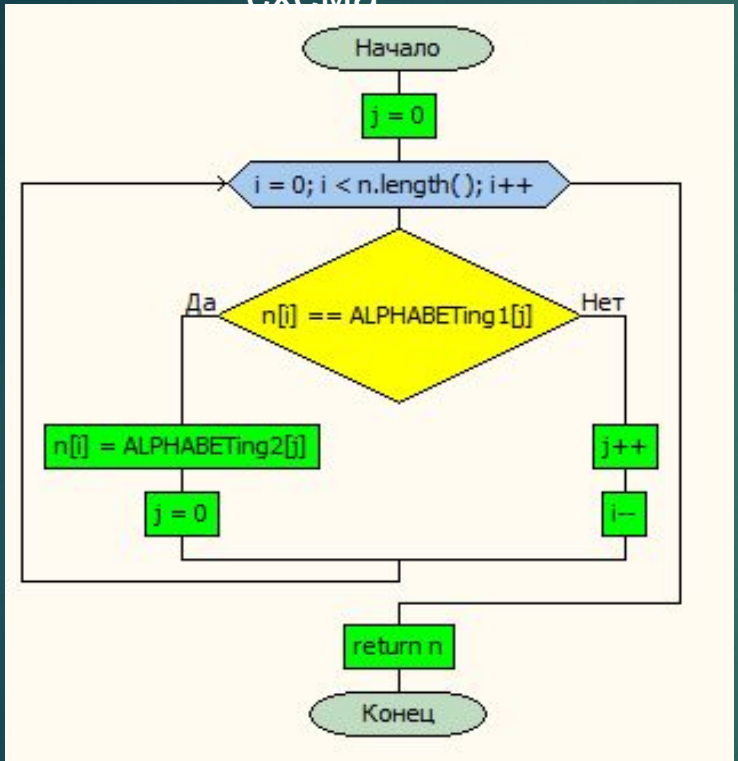


Таблица для кодирования слова word простейшим ключом key (0)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	‘	/	(	)	.	,	!	;	:	-	=	+	@	#	*	%	^	&	?	$	<	>	{	}
0	}	‘	/	(	)	.	,	!	;	:	-	=	+	@	#	*	%	^	&	?	$	<	>	{
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

Функция Fone()

Блок-схема

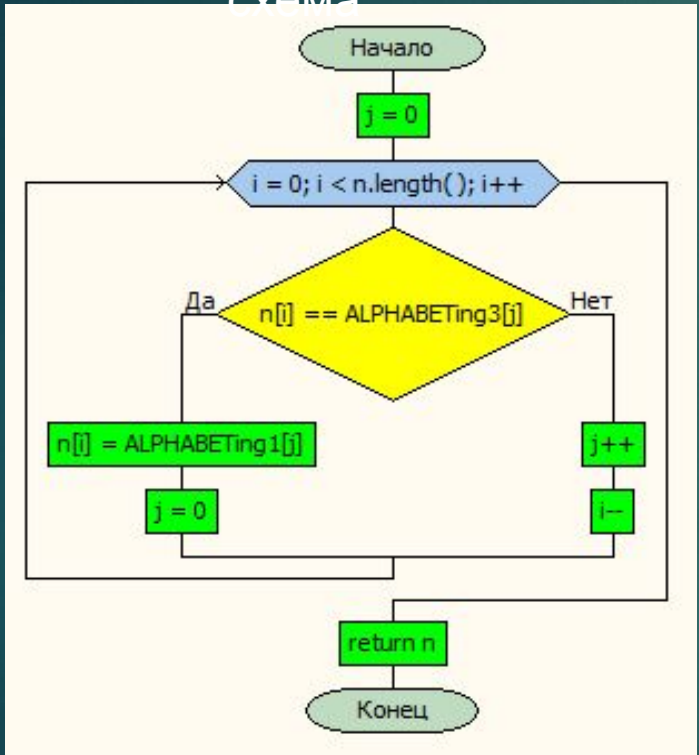


Таблица для декодирования слова word простейшим ключом key (1)

	Y	S	G	R	Z	W	U	K	C	N	M	X	B	J	T	H	O	D	L	E	F	A	I	P	Q	V
1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
\0	\0	\0	\0		\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	y	s	g	r	z	w	u	k	c	n	m	x	b	j	t	h	o	d	l	e	f	a	i	p	q	v
1	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	-	=	+	@	#	*	%	^	&	?	$	<	>	{	}	‘	/	(	)	.	,	!	;	‘
1	‘	/	(	)	.	,	!	;	:	-	=	+	@	#	*	%	^	&	?	$	<	>	{	}
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0		\0	\0	\0	\0

Функция Ftwo ()

Блок-

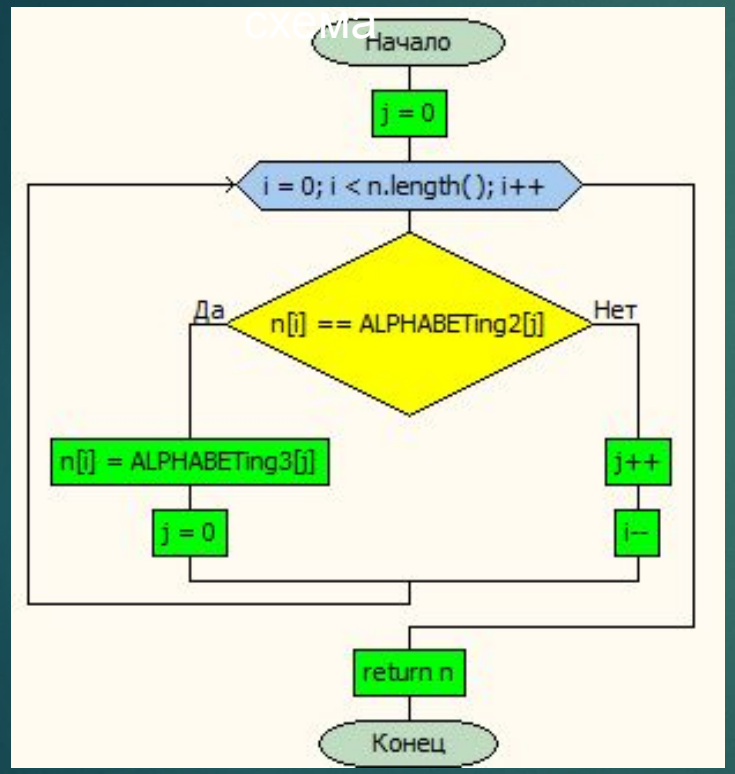


Таблица для кодирования слова word простейшим ключом key (2)

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	Y	S	G	R	Z	W	U	K	C	N	M	X	B	J	T	H	O	D	L	E	F	A	I	P	Q	V
	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
2	y	s	g	r	z	w	u	k	c	n	m	x	b	j	t	h	o	d	l	e	f	a	i	p	q	v
	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	}	‘	/	(	)	.	,	!	;	:	-	=	+	@	#	*	%	^	&	?	$	<	>	{
2	-	=	+	@	#	*	%	^	&	?	$	<	>	{	}	‘	/	(	)	.	,	!	;	‘
	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

Функция FNnull()

Блок-схема

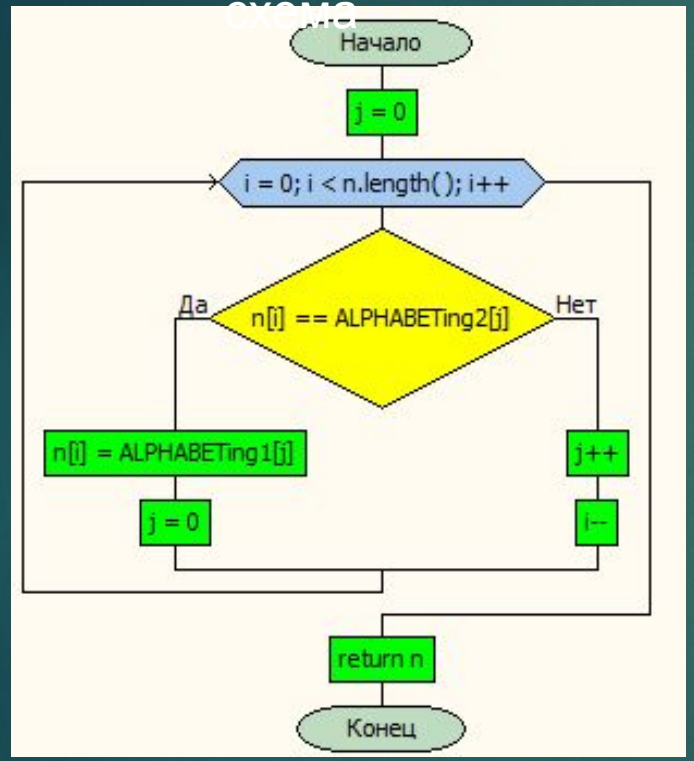


Таблица для декодирования слова word простейшим ключом key (0)

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
0	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	}	‘	/	(	)	.	,	!	;	:	-	=	+	@	#	*	%	^	&	?	$	<	>	{
0	‘	/	(	)	.	,	!	;	:	-	=	+	@	#	*	%	^	&	?	$	<	>	{	}
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

Функция FNone()

Блок-схема

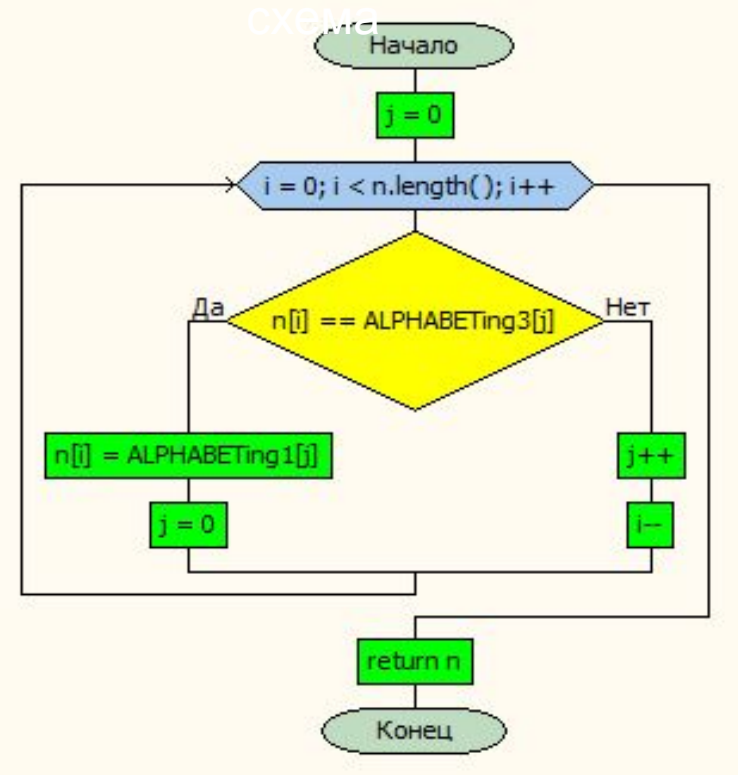


Таблица для декодирования слова word простейшим ключом key (1)

	Y	S	G	R	Z	W	U	K	C	N	M	X	B	J	T	H	O	D	L	E	F	A	I	P	Q	V
1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
\0	\0	\0	\0		\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	y	s	g	r	z	w	u	k	c	n	m	x	b	j	t	h	o	d	l	e	f	a	i	p	q	v
1	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	-	=	+	@	#	*	%	^	&	?	$	<	>	{	}	‘	/	(	)	.	,	!	;	‘
1	‘	/	(	)	.	,	!	;	:	-	=	+	@	#	*	%	^	&	?	$	<	>	{	}
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0		\0	\0	\0	\0

Функция FNtwo ()

Блок-схема

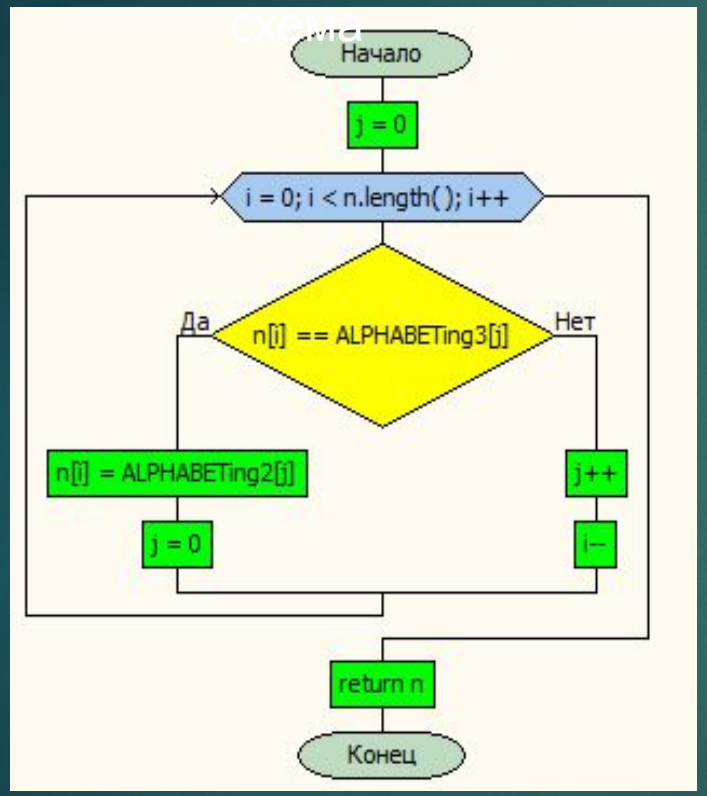


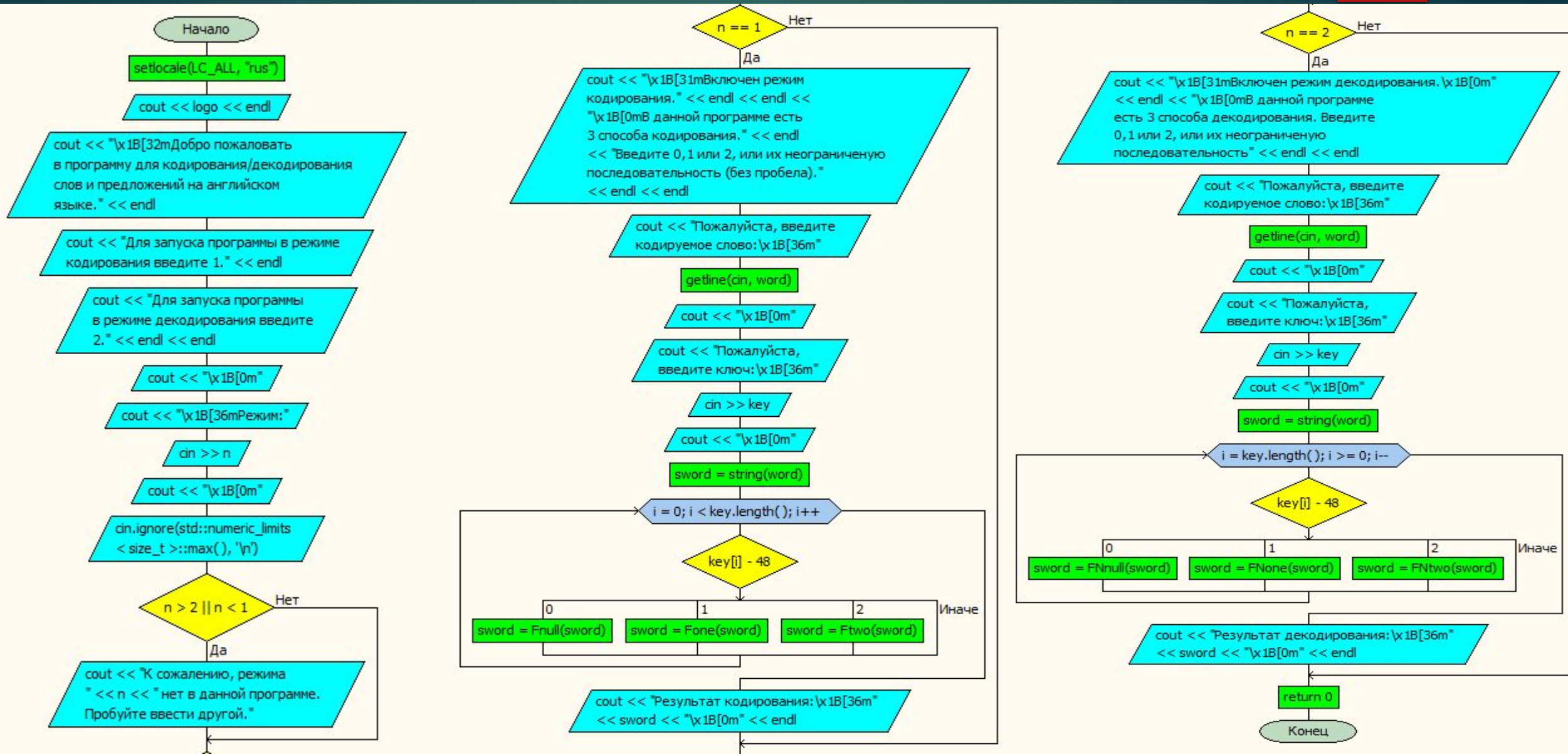
Таблица для декодирования слова word простейшим ключом key (2)

	Y	S	G	R	Z	W	U	K	C	N	M	X	B	J	T	H	O	D	L	E	F	A	I	P	Q	V
2	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	y	s	g	r	z	w	u	k	c	n	m	x	b	j	t	h	o	d	l	e	f	a	i	p	q	v
2	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

	-	=	+	@	#	*	%	^	&	?	$	<	>	{	}	‘	/	(	)	.	,	!	;	‘
2	}	‘	/	(	)	.	,	!	;	:	-	=	+	@	#	*	%	^	&	?	$	<	>	{
\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0	\0

Блок-схема тела программы



Результат работы программы

Режим кодирования:

Вводимое слово: Hi, my name is Anya Ashaeva. I live in the city of Nizhny Novgorod. Education: NGTU named after Alekseev.

Простейший ключ: 0

Результат кодирования: Ij. nz obnf jt Bozb Btibfwb) J mjwf jo uif djuz pg Ojaioz Opwhpspe) Fevdbujpo; OHUV obnfe bgufs Bmfltffw)

```
* Nizhniy Novgorod Technical University *
* *Course work*
* Performed student Ashaeva A.N. 19-IVT-2 *
************************************************************

Добро пожаловать в программу для кодирования/декодирования слов и предложений на
 английском языке.
Для запуска программы в режиме кодирования введите 1.
Для запуска программы в режиме декодирования введите 2.

Режим:1
Включен режим кодирования.

В данной программе есть 3 способа кодирования.
Введите 0,1 или 2, или их неограниченую последовательность (без пробела).

Пожалуйста, введите кодируемое слово:Hi, my name is Anya Ashaeva. I live in the
city of Nizhny Novgorod. Education: NGTU named after Alekseev.
Пожалуйста, введите ключ:0
Результат кодирования:Ij. nz obnf jt Bozb Btibfwb) J mjwf jo uif djuz pg Ojaioz
Opwhpspe) Fevdbujpo; OHUV obnfe bgufs Bmfltffw)

C:\Users\user\Documents\Прога\моя попытка номер 5\Debug\моя попытка номер 5.exe
(процесс 4408) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

Результат работы программы

Режим кодирования:

Вводимое слово: Hi, my name is Anya Ashaeva. I live in the city of Nizhny Novgorod. Education: NGTU named after Alekseev.

Простейший ключ: 1

Результат кодирования: Kc# bq jybz cl Yjqy Ylkyzay@ C xcaz cj ekz gceq tw Jcvkjq Jtautdtr@ Zrfgyectj^ JUEF jybzr ywezd Yxzmlzza@

```
* Nizhniy Novgorod Technical University *
* *Course work*
* Performed student Ashaeva A.N. 19-IVT-2 *
**********************************************************

Добро пожаловать в программу для кодирования/декодирования слов и предложений на
 английском языке.
Для запуска программы в режиме кодирования введите 1.
Для запуска программы в режиме декодирования введите 2.

Режим:1
Включен режим кодирования.

В данной программе есть 3 способа кодирования.
Введите 0,1 или 2, или их неограниченую последовательность (без пробела).

Пожалуйста, введите кодируемое слово:Hi, my name is Anya Ashaeva. I live in the
city of Nizhny Novgorod. Education: NGTU named after Alekseev.
Пожалуйста, введите ключ:1
Результат кодирования:Kc# bq jybz cl Yjqy Ylkyzay@ C xcaz cj ekz gceq tw Jcvkjq
Jtautdtr@ Zrfgyectj^ JUEF jybzr ywezd Yxzmlzza@

C:\Users\user\Documents\Прога\моя попытка номер 5\Debug\моя попытка номер 5.exe
(процесс 13172) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

Результат работы программы

Режим кодирования:
Вводимое слово: Hi, my name is Anya Ashaeva. I live in the city of Nizhny Novgorod. Education: NGTU named after Alekseev.
Простейший ключ: 2

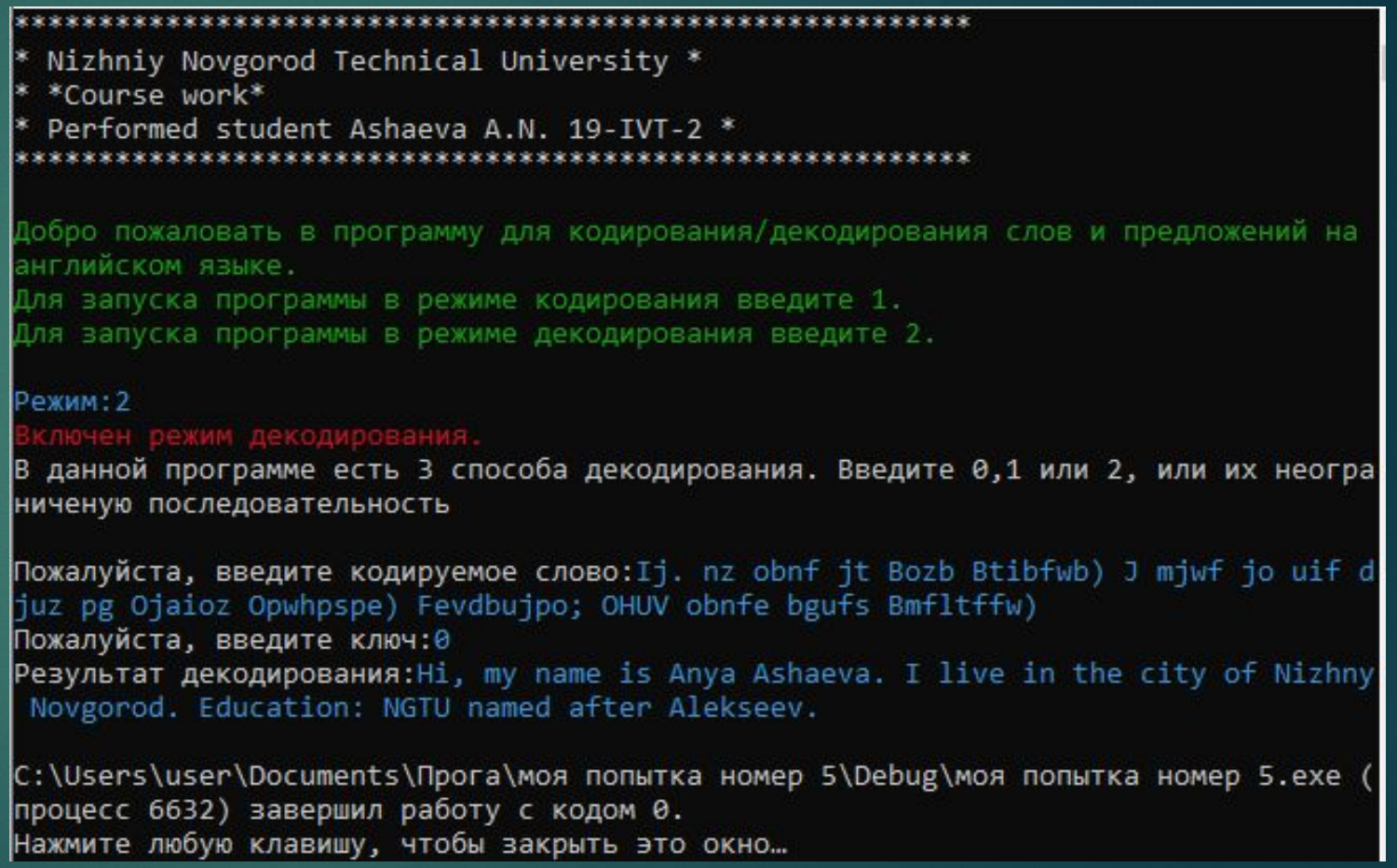
Результат кодирования: Uk* xp bvxr kd Vbpv Vduvrfv# K mkfr kb lur sklp jz Bkqubp Bjfwjojg# Rgesvlkjb& BWLE bvxrg vzlro Vmrndrrf#

```
* Nizhniy Novgorod Technical University *
* *Course work*
* Performed student Ashaeva A.N. 19-IVT-2 *
************************************************************

Добро пожаловать в программу для кодирования/декодирования слов и предложений на
 английском языке.
Для запуска программы в режиме кодирования введите 1.
Для запуска программы в режиме декодирования введите 2.

Режим:1
Включен режим кодирования.

В данной программе есть 3 способа кодирования.
Введите 0,1 или 2, или их неограниченую последовательность (без пробела).

Пожалуйста, введите кодируемое слово:Hi, my name is Anya Ashaeva. I live in the
city of Nizhny Novgorod. Education: NGTU named after Alekseev.
Пожалуйста, введите ключ:2
Результат кодирования:Uk* xp bvxr kd Vbpv Vduvrfv# K mkfr kb lur sklp jz Bkqubp
Bjfwjojg# Rgesvlkjb& BWLE bvxrg vzlro Vmrndrrf#

C:\Users\user\Documents\Прога\моя попытка номер 5\Debug\моя попытка номер 5.exe
(процесс 13492) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

Результат работы программы

Режим декодирования:
Вводимое слово: Ij. nz obnf jt Bozb Btibfwb) J mjwf jo uif djuz pg Ojaioz Opwhpspe) Fevdbujpo; OHUV obnfe bgufs Bmfltffw)
Простейший ключ: 0

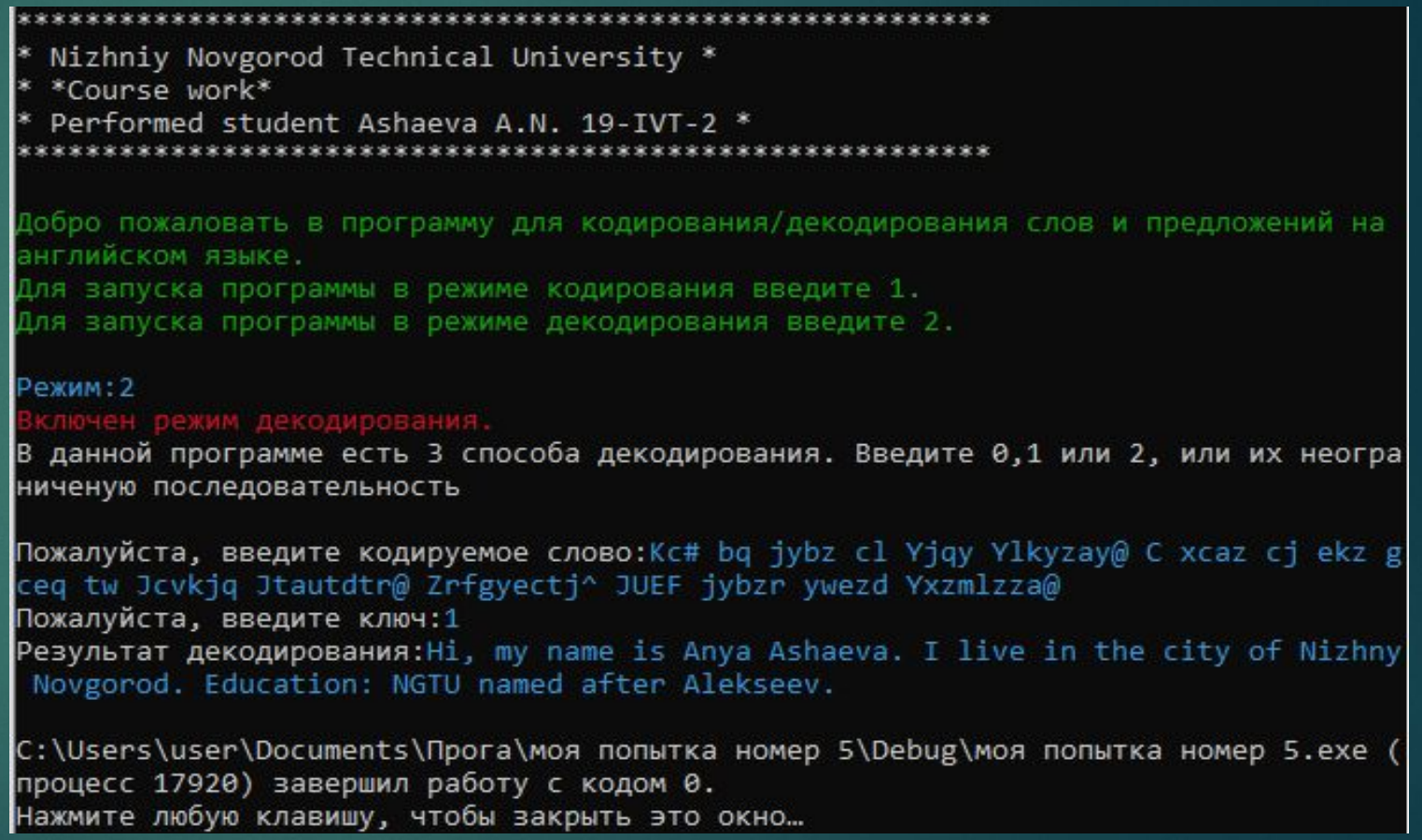
Результат декодирования: Hi, my name is Anya Ashaeva. I live in the city of Nizhny Novgorod. Education: NGTU named after Alekseev.



Результат работы программы

Режим декодирования:
Вводимое слово: Kc# bq jybz cl Yjqy Ylkyzay@ C xcaz cj ekz gceq tw Jcvkjq Jtautdtr@ Zrfgyectj^ JUEF jybzr ywezd Yxzmlzza@
Простейший ключ: 1

Результат декодирования: Hi, my name is Anya Ashaeva. I live in the city of Nizhny Novgorod. Education: NGTU named after Alekseev.



Результат работы программы

Режим декодирования:
Вводимое слово: Kc# bq jybz cl Yjqy Ylkyzay@ C xcaz cj ekz gceq tw Jcvkjq Jtautdtr@ Zrfgyectj^ JUEF jybzr ywezd Yxzmlzza@
Простейший ключ: 1

Результат декодирования: Hi, my name is Anya Ashaeva. I live in the city of Nizhny Novgorod. Education: NGTU named after Alekseev.

```
**********************************************************
* Nizhniy Novgorod Technical University *
* *Course work*
* Performed student Ashaeva A.N. 19-IVT-2 *
**********************************************************

Добро пожаловать в программу для кодирования/декодирования слов и предложений на
английском языке.
Для запуска программы в режиме кодирования введите 1.
Для запуска программы в режиме декодирования введите 2.

Режим:2
Включен режим декодирования.
В данной программе есть 3 способа декодирования. Введите 0,1 или 2, или их неогра
ниченую последовательность

Пожалуйста, введите кодируемое слово:Kc# bq jybz cl Yjqy Ylkyzay@ C xcaz cj ekz g
ceq tw Jcvkjq Jtautdtr@ Zrfgyectj^ JUEF jybzr ywezd Yxzmlzza@
Пожалуйста, введите ключ:1
Результат декодирования:Hi, my name is Anya Ashaeva. I live in the city of Nizhny
 Novgorod. Education: NGTU named after Alekseev.

C:\Users\user\Documents\Прога\моя попытка номер 5\Debug\моя попытка номер 5.exe (
процесс 17920) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

Литература

1. https://moluch.ru/conf/tech/archive/88/4317/
2. http://fmoit.arabaev.kg/wp-content/uploads/2019/04/Teor.alg.-okonchat.pdf
3. Ломакина Л.С. Алгоритмы и теория сложности. Методические материалы. Электронный ресурс, НГТУ им. Р.Е.Алексеева, Нижний Новгород
4. Ломакина Л.С. Методические материалы к изучению курса Теоретические основы алгоритмизации 2020, Электронный ресурс, НГТУ им. Р.Е.Алексеева, Нижний Новгород