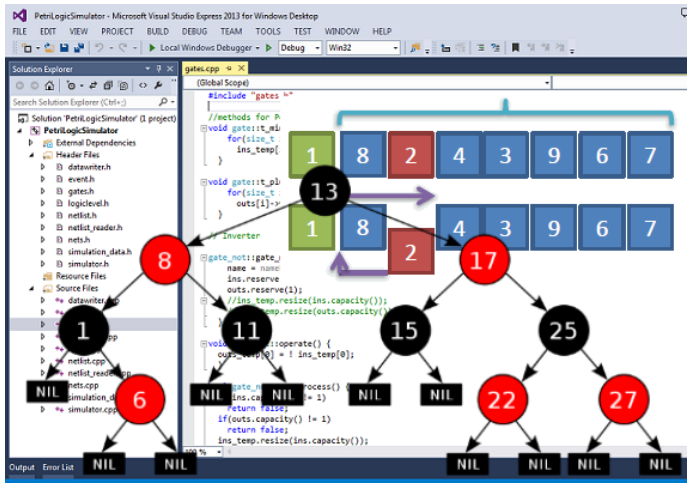
Национальный исследовательский университет «МИЭТ»

Институт интегральной электроники (группы ЭН-24-25, каф. ПКИМС)

Теория алгоритмов



Лекция 7

Коды. Кодирование данных

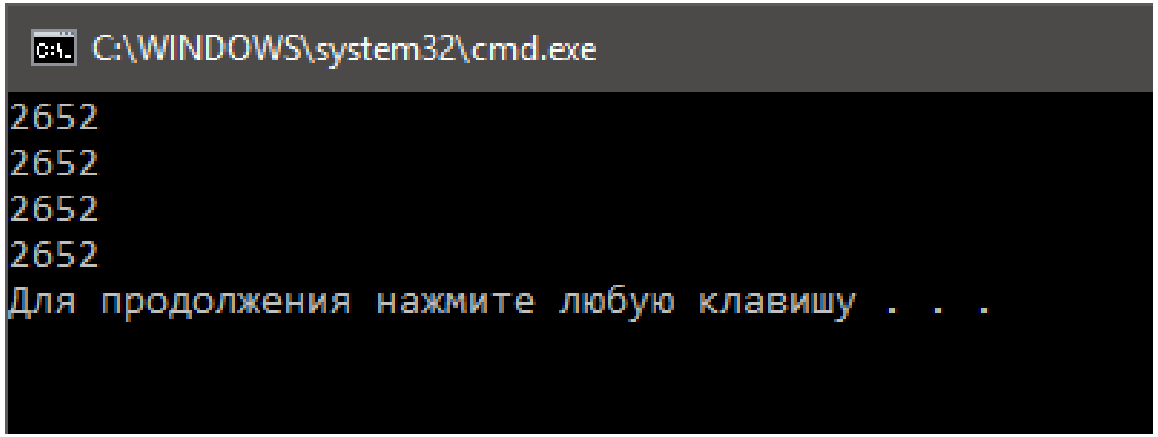
Азбука Морзе: сравнение латиницы и кириллицы

А	A	· –
Б	B	– · · ·
В	W	· – –
Г	G	– – ·
Д	D	– · ·
Е (также и Ё)	E	·
Ж	V	· · · –
З	Z	– – · ·
И	I	· ·
Й	J	· – – –
К	K	– · –
Л	L	· – · ·
М	M	– –
Н	N	– ·
О	O	– – –

П	P	· – – ·
Р	R	· – ·
С	S	· · ·
Т	T	–
У	U	· · –
Ф	F	· · – ·
Х	H	· · · ·
Ц	C	– · – ·
Ч	Ö	– – – ·
Ш	CH	– – – –
Щ	Q	– – · –
Ъ	Ñ	– – · – –
Ы	Y	– · – –
Ь (также и Ъ)	X	– · · –
Э	É	· · – · ·
Ю	Ü	· · – –
Я	Ä	· – · –

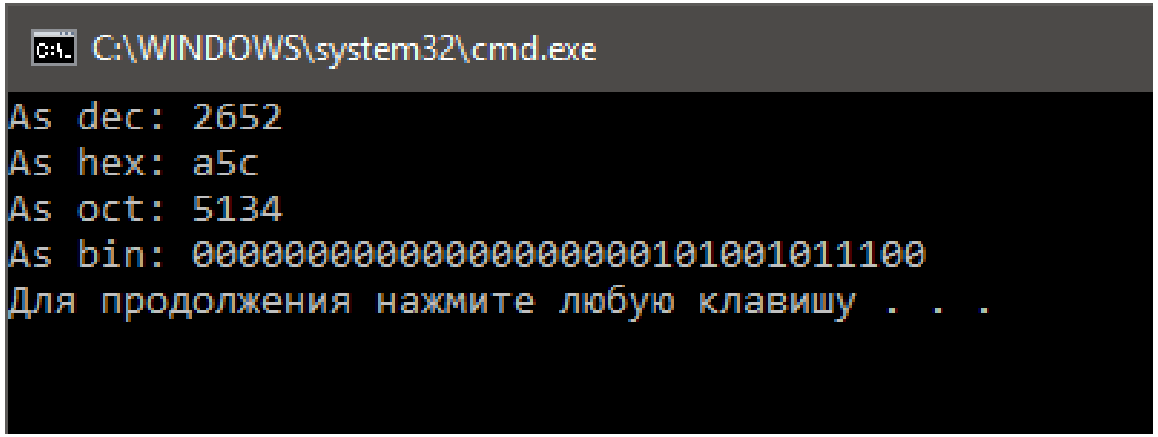
Запись чисел в различных системах счисления в C++

```cpp
#include <iostream>

int main(int argc, char *argv[]) {

  int var_dec = 2652;
  int var_oct = 05134;
  int var_hex = 0xA5C;
  int var_bin = 0b101001011100;

  std::cout << var_dec << std::endl;
  std::cout << var_oct << std::endl;
  std::cout << var_hex << std::endl;
  std::cout << var_bin << std::endl;

  return 0;
}
```



Вывод чисел в различных системах счисления в C++ (1)

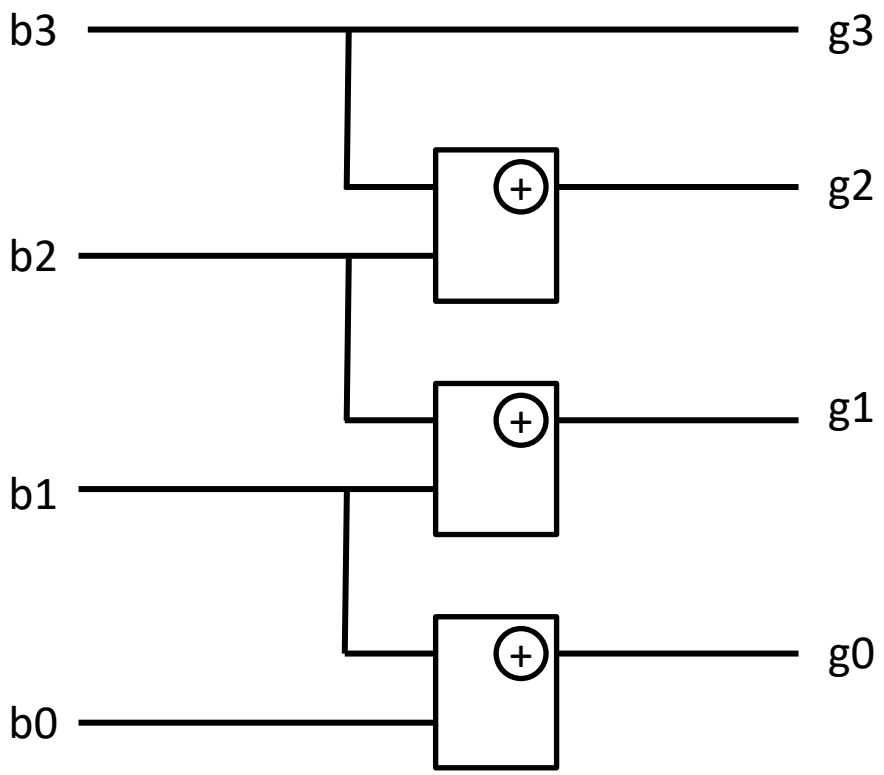
```
#include <iostream>
#include <bitset>

int main(int argc, char *argv[]) {

  int var = 2652;

  std::cout << "As dec: " << std::dec << var << std::endl;
  std::cout << "As hex: " << std::hex << var << std::endl;
  std::cout << "As oct: " << std::oct << var << std::endl;

  std::bitset<8 * sizeof(int)> bits(var);
  std::cout << "As bin: " << bits << std::endl;

  return 0;
}
```



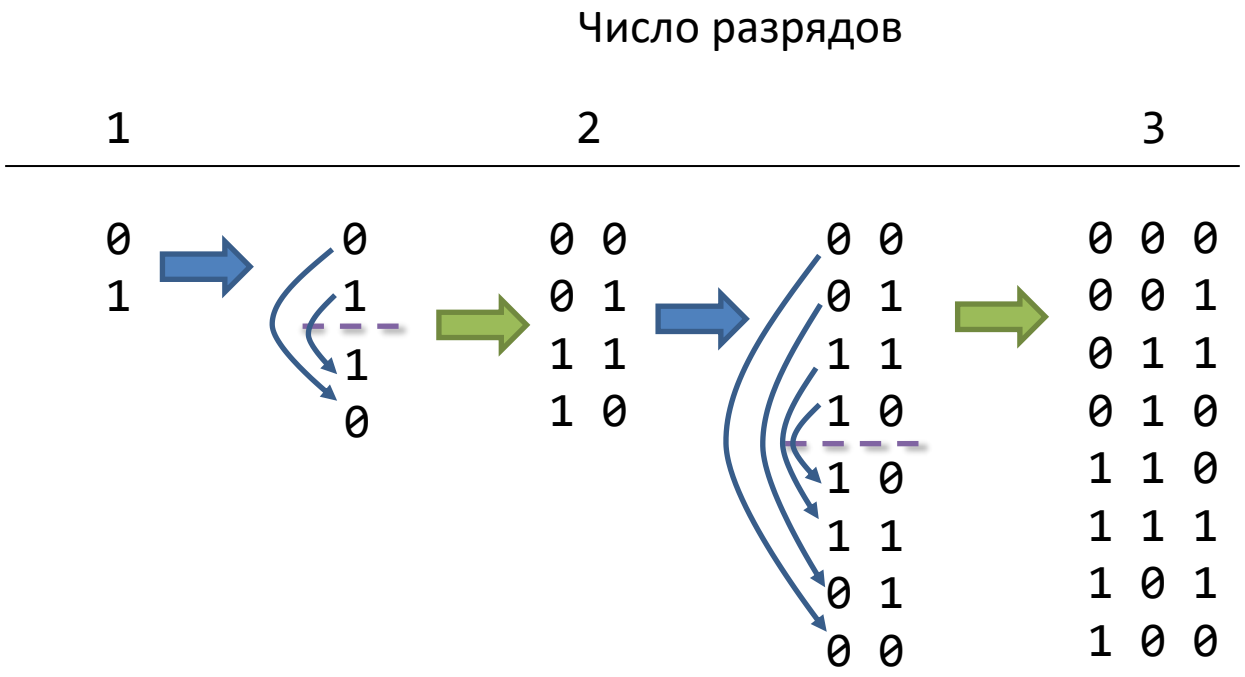
Альтернативные системы кодирования: коды Грэя

Двоичный код	
0	**0 0 0 0**
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0
15	1 1 1 1

Двоичный код Грэя	
0	**0 0 0 0**
1	0 0 0 1
2	0 0 1 1
3	0 0 1 0
4	0 1 1 0
5	0 1 1 1
6	0 1 0 1
7	0 1 0 0
8	1 1 0 0
9	1 1 0 1
10	1 1 1 1
11	1 1 1 0
12	1 0 1 0
13	1 0 1 1
14	1 0 0 1
15	1 0 0 0



Построение зеркального кода Грэя

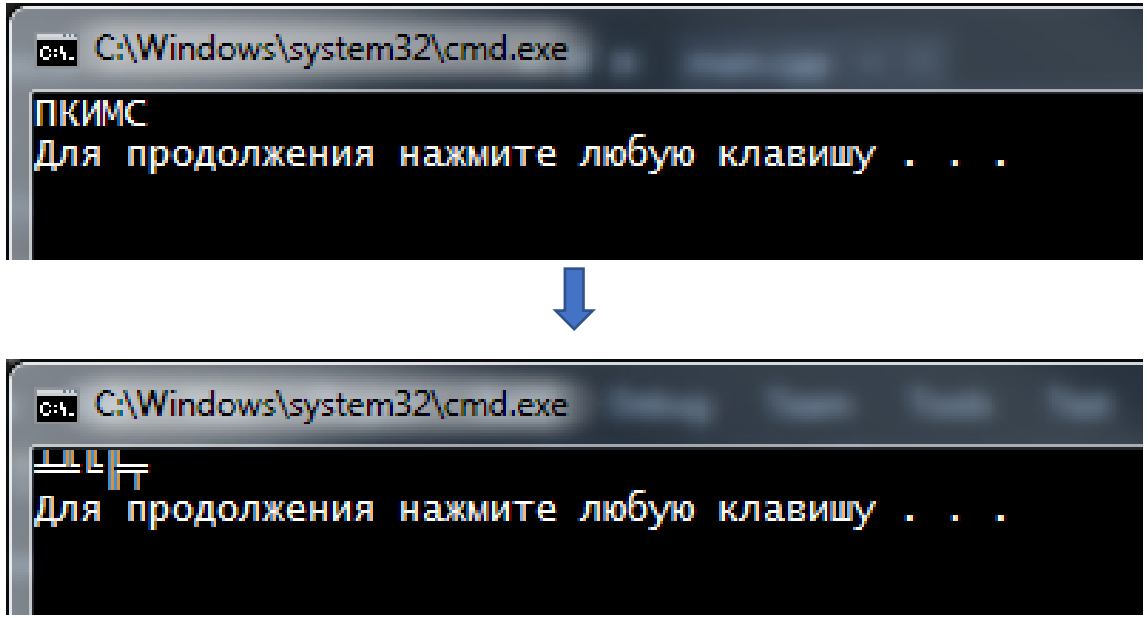


Двоичный код Грэя

0	**0 0 0 0**
1	0 0 0 1
2	0 0 1 1
3	0 0 1 0
4	0 1 1 0
5	0 1 1 1
6	0 1 0 1
7	0 1 0 0
8	1 1 0 0
9	1 1 0 1
10	1 1 1 1
11	1 1 1 0
12	1 0 1 0
13	1 0 1 1
14	1 0 0 1
15	1 0 0 0

Работа с локалями (1)

```
#include <iostream>
using namespace std;

int main() {
  std::cout << "ПКИМС" << std::endl;
  return EXIT_SUCCESS;
}
```



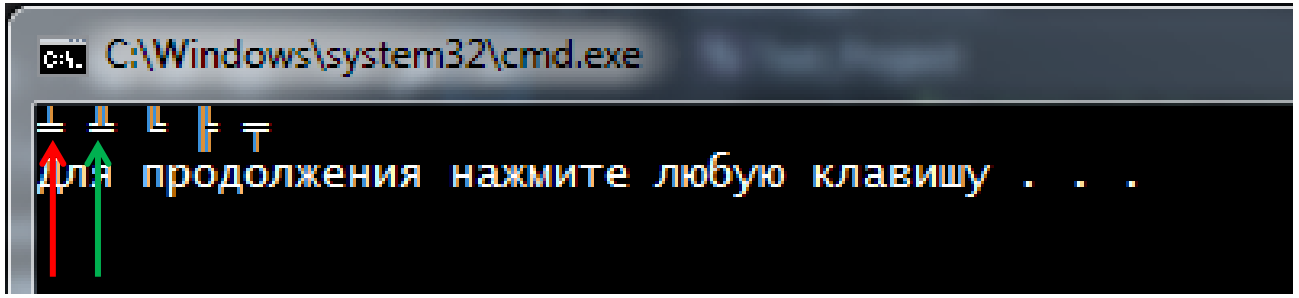
Работа с локалями (2)

Код пишется под Windows
(кодировка CP1251)

Ѓ 128	ѓ 129	, 130	ѐ 131	„ 132	… 133	† 134	‡ 135	€ 136	‰ 137	Љ 138	‹ 139	њ 140	Ќ 141	Ћ 142	Џ 143
ђ 144	‘ 145	’ 146	“ 147	” 148	• 149	– 150	— 151	ѓ 152	™ 153	љ 154	› 155	ќ 156	ћ 157	џ 158	159
nbsp 160	Ў 161	ў 162	Ј 163	¤ 164	Ґ 165	¦ 166	§ 167	Ё 168	© 169	Є 170	« 171	¬ 172	shy 173	® 174	Ї 175
° 176	± 177	І 178	і 179	ґ 180	µ 181	¶ 182	· 183	ё 184	№ 185	є 186	» 187	ј 188	Ѕ 189	ѕ 190	ї 191
А 192	Б 193	В 194	Г 195	Д 196	Е 197	Ж 198	З 199	И 200	Й 201	К 202	Л 203	М 204	Н 205	О 206	П 207
Р 208	С 209	Т 210	У 211	Ф 212	Х 213	Ц 214	Ч 215	Ш 216	Щ 217	Ъ 218	Ы 219	Ь 220	Э 221	Ю 222	Я 223
а 224	б 225	в 226	г 227	д 228	е 229	ж 230	з 231	и 232	й 233	к 234	л 235	м 236	н 237	о 238	п 239
р 240	с 241	т 242	у 243	ф 244	х 245	ц 246	ч 247	ш 248	щ 249	ъ 250	ы 251	ь 252	э 253	ю 254	я 255

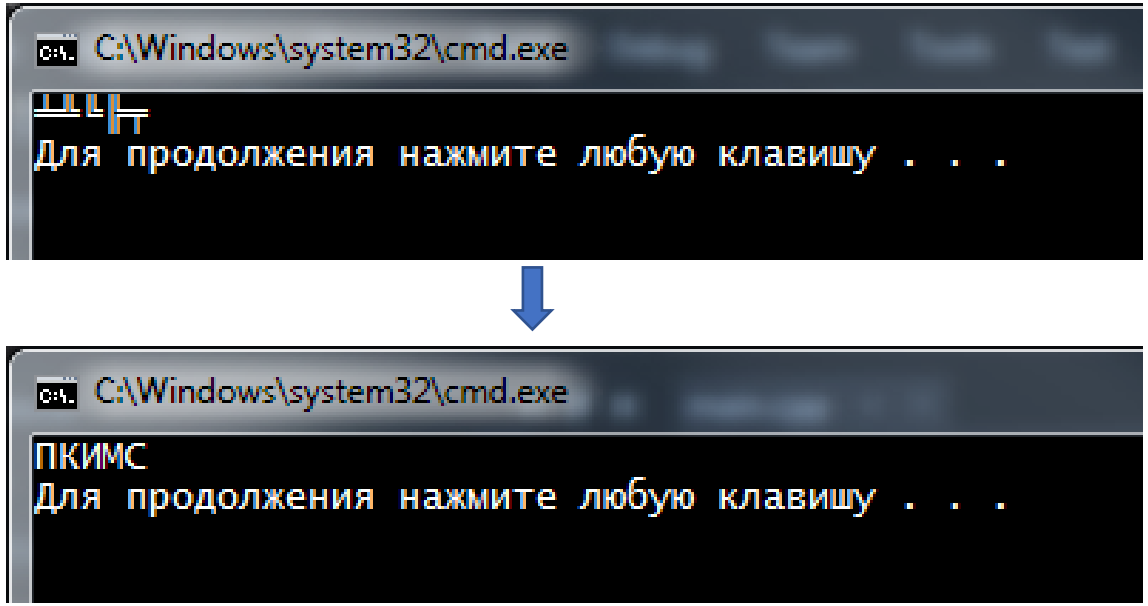
Код работает под MS-DOS
(кодировка CP866)

А 128	Б 129	В 130	Г 131	Д 132	Е 133	Ж 134	З 135	И 136	Й 137	К 138	Л 139	М 140	Н 141	О 142	П 143
Р 144	С 145	Т 146	У 147	Ф 148	Х 149	Ц 150	Ч 151	Ш 152	Щ 153	Ъ 154	Ы 155	Ь 156	Э 157	Ю 158	Я 159
а 160	б 161	в 162	г 163	д 164	е 165	ж 166	з 167	и 168	й 169	к 170	л 171	м 172	н 173	о 174	п 175
▒ 176	▓ 177	▓ 178	│ 179	┤ 180	╡ 181	╢ 182	╖ 183	╕ 184	╣ 185	║ 186	╗ 187	╝ 188	╜ 189	╛ 190	┐ 191
└ 192	┴ 193	┬ 194	├ 195	─ 196	┼ 197	╞ 198	╟ 199	╚ 200	╔ 201	╩ 202	╦ 203	╠ 204	═ 205	╬ 206	╧ 207
╨ 208	╤ 209	╥ 210	╙ 211	╘ 212	╒ 213	╓ 214	╫ 215	╪ 216	┘ 217	┌ 218	▉ 219	▄ 220	▌ 221	▐ 222	▀ 223
р 224	с 225	т 226	у 227	ф 228	х 229	ц 230	ч 231	ш 232	щ 233	ъ 234	ы 235	ь 236	э 237	ю 238	я 239
Ё 240	ё 241	Є 242	є 243	Ї 244	ї 245	Ў 246	ў 247	° 248	∙ 249	· 250	√ 251	№ 252	¤ 253	■ 254	nbsp 255



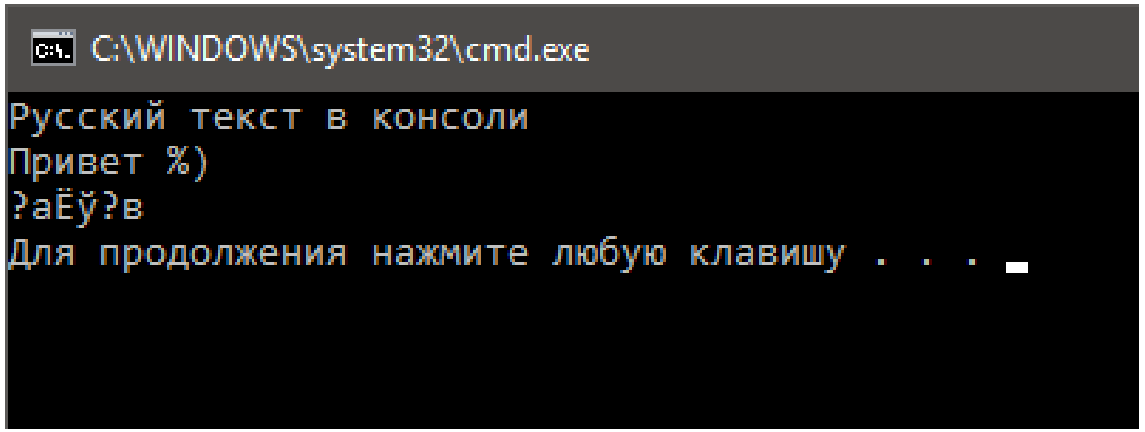
Работа с локалями (3)

```
#include <iostream>
#include <clocale>
using namespace std;

int main() {
  setlocale(LC_CTYPE, "rus");
  std::cout << "ПКИМС" << std::endl;
  return EXIT_SUCCESS;
}
```



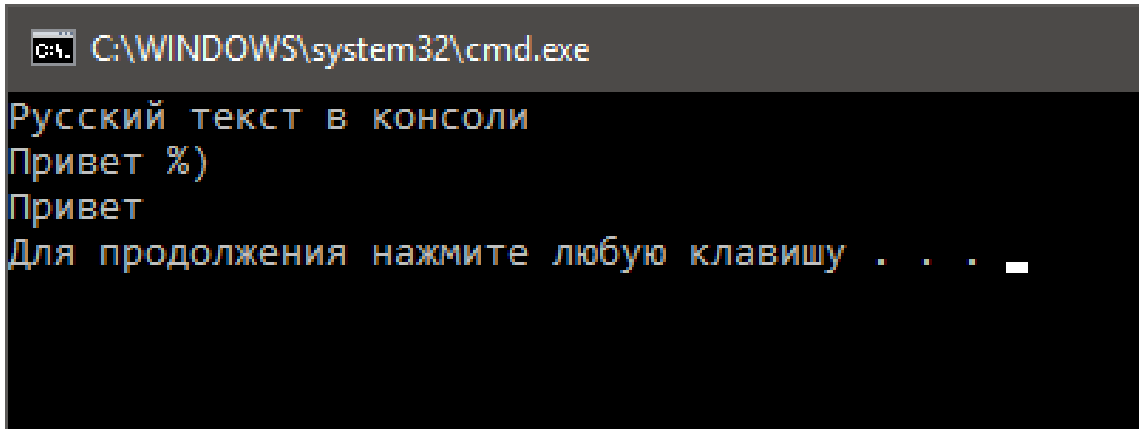
Чтение русского текста в консоли (1)

```
#include <iostream>
#include <string>

int main(int argc, char *argv[]) {
  setlocale(LC_ALL, "Russian");
  std::cout << "Русский текст в консоли" << std::endl;

  std::string str;
  std::cin >> str;

  std::cout << str << std::endl;

  return 0;
}
```



Чтение русского текста в консоли (2)

```
#include <Windows.h>
#include <iostream>
#include <string>

int main(int argc, char *argv[]) {
  SetConsoleOutputCP(1251);
  SetConsoleCP(1251);
  std::cout << "Русский текст в консоли" << std::endl;

  std::string str;
  std::cin >> str;

  std::cout << str << std::endl;

  return 0;
}
```



Многобайтная кодировка Unicode

Обозначение номера символа в таблице Unicode: U+XXXXXX

0410	А
0430	а
0411	Б
0431	б
0412	В

0466	Ѧ
0467	ѧ
0468	Ѩ
0469	ѩ
046A	Ѫ

046E	Ѯ
046F	ѯ
0470	Ѱ
0471	ѱ
0472	Ѳ

Буква латиницы О с акутом

Óó

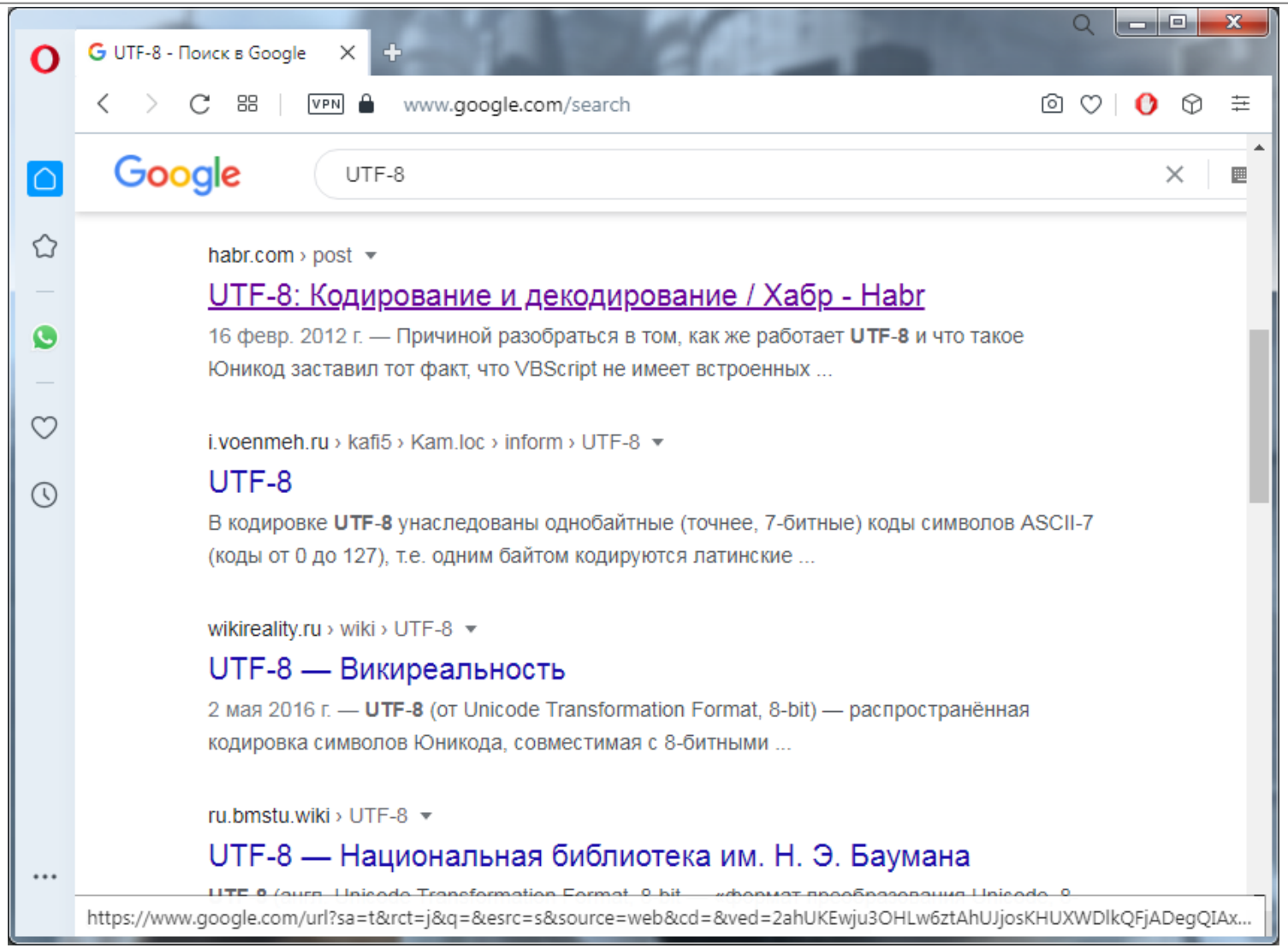
Изображение [показать]

◄ Ï Ð Ñ Ò **Ó** Ô Õ Ö × ►
◄ ï ð ñ ò **ó** ô õ ö ÷ ►

Характеристики

Название	Ó: LATIN CAPITAL LETTER O WITH ACUTE ó: LATIN SMALL LETTER O WITH ACUTE
Юникод	Ó: U+00D3 ó: U+00F3
HTML-код	Ó: `&#211;` *или* `&#xd3;` ó: `&#243;` *или* `&#xf3;`
UTF-16	Ó: 0xD3 ó: 0xF3
URL-код	Ó: %C3%93 ó: %C3%B3
Мнемоника	Ó: `&Oacute;` ó: `&oacute;`

URL-кодирование (1)



https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwju3OHLw6ztAhUJjosKHUXWDlkQFjADegQIAxAC&url=https%3A%2F%2Fhabr.com%2Fru%2Fpost%2F138173%2F&usg=AOvVaw38VoofJOwoNswAJJ1hh5mp

URL-кодирование (2)

<протокол>://[//[<логин>[:<пароль>]@]<хост>[:<порт>]][/<URL-путь>][?<параметры>][#<якорь>]

https://www.google.com/url?source=web&url=https%3A%2F%2Fhabr.com%2Fru%2Fpost%2F138173%2F&usg=AOvVaw38VoofJOwoNswAJJ1hh5mp

https://www.google.com/url?source=web&url=https://habr.com/ru/post/138173/&usg=AOvVaw38VoofJOwoNswAJJ1hh5mp

Кодирование специальных символов:

!	#	$	%	&	'	(	)	*	+	,	/	:	;	=	?	@	[	]
%21	%23	%24	%25	%26	%27	%28	%29	%2A	%2B	%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D

Кодирование Base64 (1)

Позволяет закодировать любой символ набором из 64 ASCII-символов:

- латинские буквы верхнего регистра [A-Z] (26)
- латинские буквы нижнего регистра [a-z] (26)
- цифры [0-9] (10)
- два спец. символа [+/]

Обычный вид результата кодирования:

```
CtCX0LTRgNCw0LLRgdGC0LLRg9C50YLQtSwg0YMg0LzQtdC90Y8g0LLQvtC30L3QuNC60LvQuCDR
gtGA0YPQtNC90L7RgdGC0Lgg0L/RgNC4INCy0YvQv9C+0LvQvdC10L3QuNC4INC70LDQsdC+0YDQ
sNGC0L7RgNC90L7QuSwg0YHQstGP0LfQsNC90L3QvtC5INGBIEdUSy4g0K8g0YPRgdGC0LDQvdC+
0LLQuNC7INCy0YHQtSDQvdC10L7QsdGF0L7QtNC40LzQvtC1LCDQvdC/0YDQuCDQutC+0LzQv9C4
v9C40LvRj9GG0LjQuCDQu9C40L3QutC+0Log0YDRg9Cz0LDQtdGC0YHRjyDQvdCwINGB
0LvQtdC00YPRjtGJ0LjQtSDRhNGD0L3QutGG0LjQuCA6CjE+bWFpbi5vYmogOiBlcnJvciBMTksy
MDE5OiB1bnJlc29sdmVkIGV4dGVybmFsIHN5bWJvbCBnZGtfZHJhd2luZ19jb250ZXh0X2dldF9j
YWlyb19jb250ZXh0IHJlZmVyZW5jZWQgaW4gZnVuY3Rpb24gImludCBfX2NkZWNsIG9uX2RyYXco
c3RydWN0IF9HdGtXaWRnZXQgKixzdHJ1Y3QgX0dka0V2ZW50RXhwb3NlICosdm9pZCAqKSIgKD9v
bl9kcmF3QEBZQUhQRUFVX0d0a1dpZGdldEBAUEVBVV9HZGtFdmVudEV4cG9zZUBAUEVBWEBaKQox
Pm1haW4ub2JqIDogZXJyb3IgTE5LMjAxOTogdW5yZXNvbHZlZCBleHRlcm5hbCBzeW1ib2wgZ2Rr
X3dpbmRvd19iZWdpbl9kcmF3X2ZyYW1lIHJlZmVyZW5jZWQgaW4gZnVuY3Rpb24gIml=
```

Кодирование Base64 (2)

Алгоритм кодирования:

1. все символы, которые необходимо кодировать, выписываются в двоичном виде группами по 3 байта, в итоге получаются цепочки из 24 бит;
2. если для триплета байт не хватает байт, в недостающие позиции битов дописываются нули;
3. цепочка из 24 бит кодируется согласно таблице:

Символ	Значение				Символ	Значение				Символ	Значение				Символ	Значение			
	10	2	8	16		10	2	8	16		10	2	8	16		10	2	8	16
A	0	000000	00	00	Q	16	010000	20	10	g	32	100000	40	20	w	48	110000	60	30
B	1	000001	01	01	R	17	010001	21	11	h	33	100001	41	21	x	49	110001	61	31
C	2	000010	02	02	S	18	010010	22	12	i	34	100010	42	22	y	50	110010	62	32
D	3	000011	03	03	T	19	010011	23	13	j	35	100011	43	23	z	51	110011	63	33
E	4	000100	04	04	U	20	010100	24	14	k	36	100100	44	24	0	52	110100	64	34
F	5	000101	05	05	V	21	010101	25	15	l	37	100101	45	25	1	53	110101	65	35
G	6	000110	06	06	W	22	010110	26	16	m	38	100110	46	26	2	54	110110	66	36
H	7	000111	07	07	X	23	010111	27	17	n	39	100111	47	27	3	55	110111	67	37
I	8	001000	10	08	Y	24	011000	30	18	o	40	101000	50	28	4	56	111000	70	38
J	9	001001	11	09	Z	25	011001	31	19	p	41	101001	51	29	5	57	111001	71	39
K	10	001010	12	0A	a	26	011010	32	1A	q	42	101010	52	2A	6	58	111010	72	3A
L	11	001011	13	0B	b	27	011011	33	1B	r	43	101011	53	2B	7	59	111011	73	3B
M	12	001100	14	0C	c	28	011100	34	1C	s	44	101100	54	2C	8	60	111100	74	3C
N	13	001101	15	0D	d	29	011101	35	1D	t	45	101101	55	2D	9	61	111101	75	3D
O	14	001110	16	0E	e	30	011110	36	1E	u	46	101110	56	2E	+	62	111110	76	3E
P	15	001111	17	0F	f	31	011111	37	1F	v	47	101111	57	2F	/	63	111111	77	3F

Кодирование Base64: пример (1)

ПКИМС (в кодировке CP866)

П	К	И	М	С
143	138	136	140	145

А 128	Б 129	В 130	Г 131	Д 132	Е 133	Ж 134	З 135	И 136	Й 137	К 138	Л 139	М 140	Н 141	О 142	П 143
Р 144	С 145	Т 146	У 147	Ф 148	Х 149	Ц 150	Ч 151	Ш 152	Щ 153	Ъ 154	Ы 155	Ь 156	Э 157	Ю 158	Я 159
а 160	б 161	в 162	г 163	д 164	е 165	ж 166	з 167	и 168	й 169	к 170	л 171	м 172	н 173	о 174	п 175
▒ 176	▓ 177	▓ 178	│ 179	┤ 180	╡ 181	╢ 182	╖ 183	╕ 184	╣ 185	║ 186	╗ 187	╝ 188	╜ 189	╛ 190	┐ 191
└ 192	┴ 193	┬ 194	├ 195	─ 196	┼ 197	╞ 198	╟ 199	╚ 200	╔ 201	╩ 202	╦ 203	╠ 204	═ 205	╬ 206	╧ 207
╨ 208	╤ 209	╥ 210	╙ 211	╘ 212	╒ 213	╓ 214	╫ 215	╪ 216	┘ 217	┌ 218	█ 219	▄ 220	▌ 221	▐ 222	▀ 223
р 224	с 225	т 226	у 227	ф 228	х 229	ц 230	ч 231	ш 232	щ 233	ъ 234	ы 235	ь 236	э 237	ю 238	я 239
Ё 240	ё 241	Є 242	є 243	Ї 244	ї 245	Ў 246	ў 247	° 248	∙ 249	· 250	√ 251	№ 252	¤ 253	■ 254	nbsp 255

П	К	И	М	С
143	138	136	140	145
1 0 0 0 1 1 1 1	1 0 0 0 1 0 1 0	1 0 0 0 1 0 0 0	1 0 0 0 1 1 0 0	1 0 0 1 0 0 0 1

1	0	0	0	1	1	1	1	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	1	0	0	1	0	0	1	0	0	0	1	0	0

Кодирование Base64: пример (2)

П	К	И	М	С
143	138	136	140	145
1 0 0 0 1 1 1 1	1 0 0 0 1 0 1 0	1 0 0 0 1 0 0 0	1 0 0 0 1 1 0 0	1 0 0 1 0 0 0 1

1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	1	0	0	1	0	0	1	0	0	0	1	0	0

j 4 q I j J E

Символ	Значение			
	10	2	8	16
A	0	000000	00	00
B	1	000001	01	01
C	2	000010	02	02
D	3	000011	03	03
E	4	000100	04	04
F	5	000101	05	05
G	6	000110	06	06
H	7	000111	07	07
I	8	001000	10	08
J	9	001001	11	09
K	10	001010	12	0A
L	11	001011	13	0B
M	12	001100	14	0C
N	13	001101	15	0D
O	14	001110	16	0E
P	15	001111	17	0F

Символ	Значение			
	10	2	8	16
Q	16	010000	20	10
R	17	010001	21	11
S	18	010010	22	12
T	19	010011	23	13
U	20	010100	24	14
V	21	010101	25	15
W	22	010110	26	16
X	23	010111	27	17
Y	24	011000	30	18
Z	25	011001	31	19
a	26	011010	32	1A
b	27	011011	33	1B
c	28	011100	34	1C
d	29	011101	35	1D
e	30	011110	36	1E
f	31	011111	37	1F

Символ	Значение			
	10	2	8	16
g	32	100000	40	20
h	33	100001	41	21
i	34	100010	42	22
j	35	100011	43	23
k	36	100100	44	24
l	37	100101	45	25
m	38	100110	46	26
n	39	100111	47	27
o	40	101000	50	28
p	41	101001	51	29
q	42	101010	52	2A
r	43	101011	53	2B
s	44	101100	54	2C
t	45	101101	55	2D
u	46	101110	56	2E
v	47	101111	57	2F

Символ	Значение			
	10	2	8	16
w	48	110000	60	30
x	49	110001	61	31
y	50	110010	62	32
z	51	110011	63	33
0	52	110100	64	34
1	53	110101	65	35
2	54	110110	66	36
3	55	110111	67	37
4	56	111000	70	38
5	57	111001	71	39
6	58	111010	72	3A
7	59	111011	73	3B
8	60	111100	74	3C
9	61	111101	75	3D
+	62	111110	76	3E
/	63	111111	77	3F

Помехоустойчивое кодирование: контрольные суммы (CRC)

CRC – Cyclic Redundancy Check – циклическая избыточная проверка

Один из методов проверки целостности информации

CRC представляет собой остаток от деления входящего сообщения на полином

Есть стандартные полиномы:

Название	Полином
CRC1	$x + 1$
CRC8	$x^8 + x^7 + x^6 + x^4 + x^2 + 1$
CRC16	$x^{16} + x^{15} + x^2 + 1$

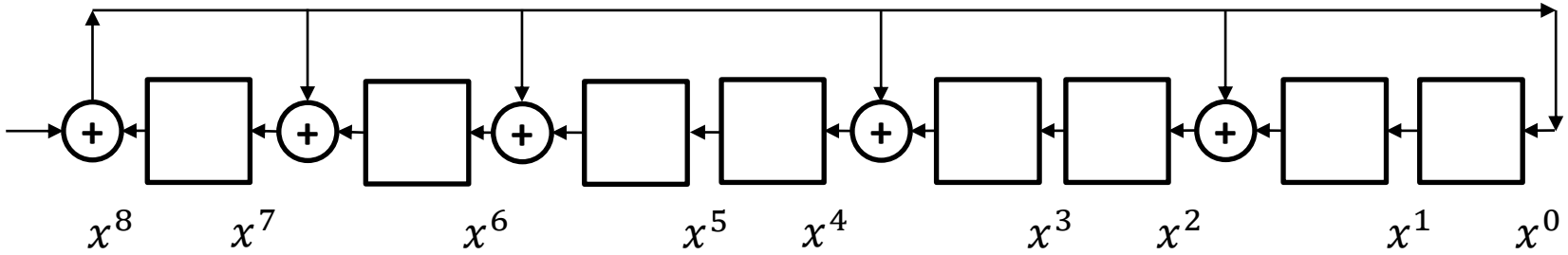
Пример подсчёта CRC8 (1)

Кодируемое сообщение: «ЭН»

Полином: $x^8 + x^7 + x^6 + x^4 + x^2 + 1$

Битовое представление сообщения: 11011101 11001101 00000000

Регистр для вычисления остатка над полиномом:



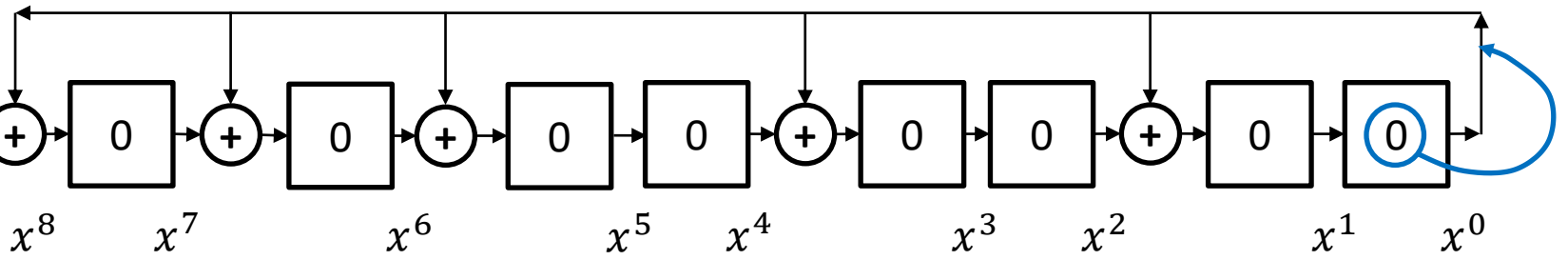
Битовое кодируемое сообщение дополняется нулями по степени полинома

Регистр изначально инициализируется нулевыми значениями

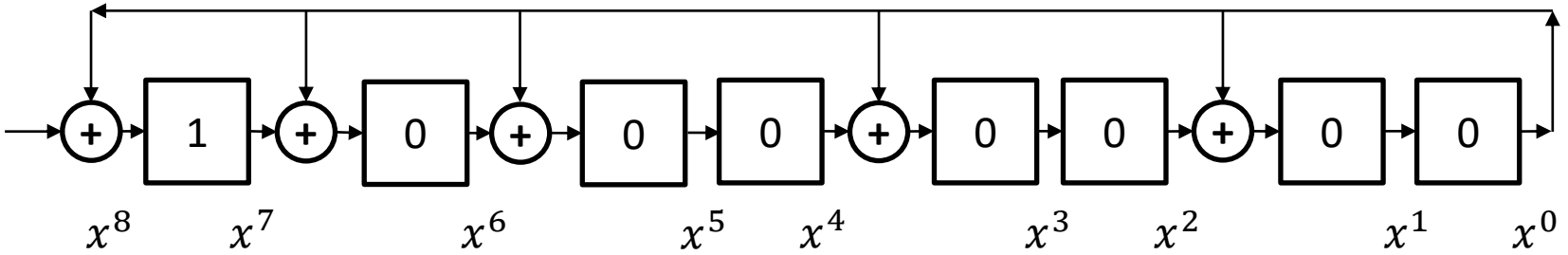
Пример подсчёта CRC8 (2)

Данные, для которых считается CRC

1 1 0 1 1 1 0 1 1 1 0 0 1 1 0 1 **0 0 0 0 0 0 0 0 0**

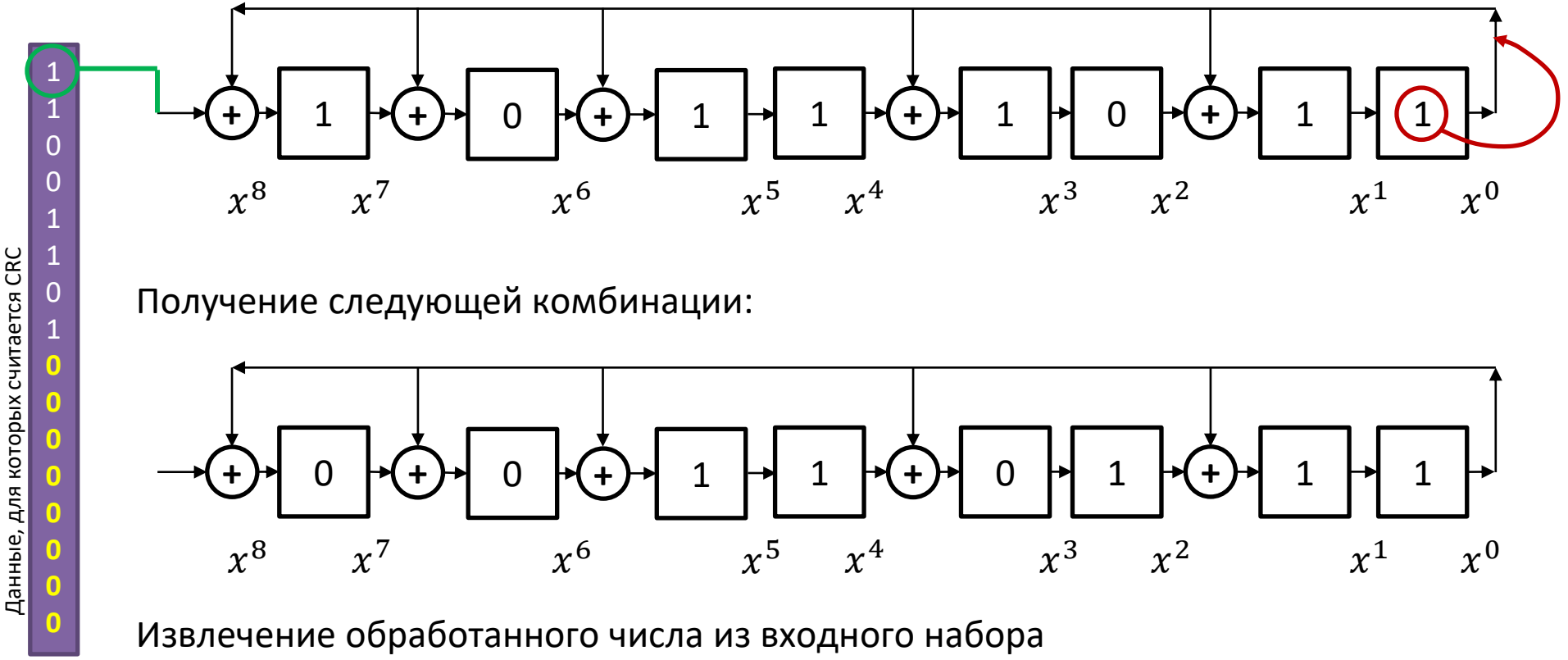


Получение следующей комбинации:



Извлечение обработанного числа из входного набора

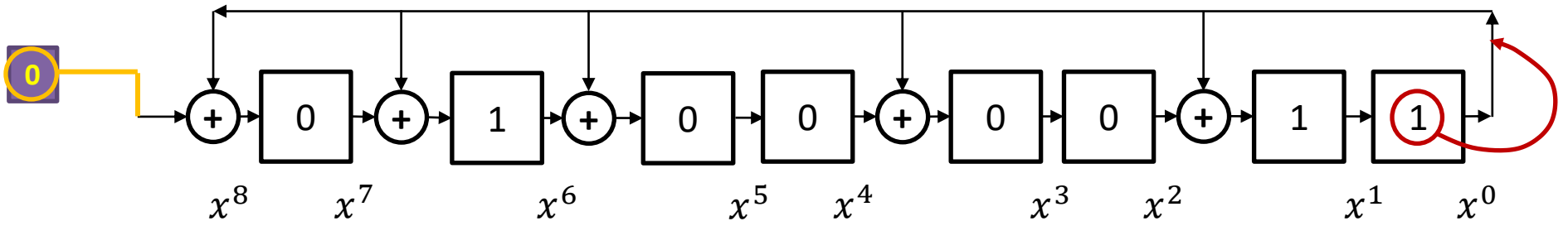
Пример подсчёта CRC8 (10)



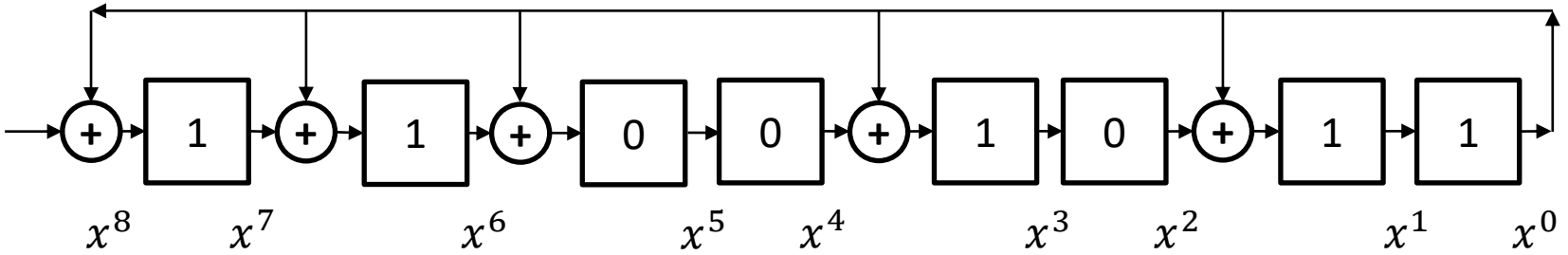
Получение следующей комбинации:

Извлечение обработанного числа из входного набора

Пример подсчёта CRC8 (25)



Получение следующей комбинации:

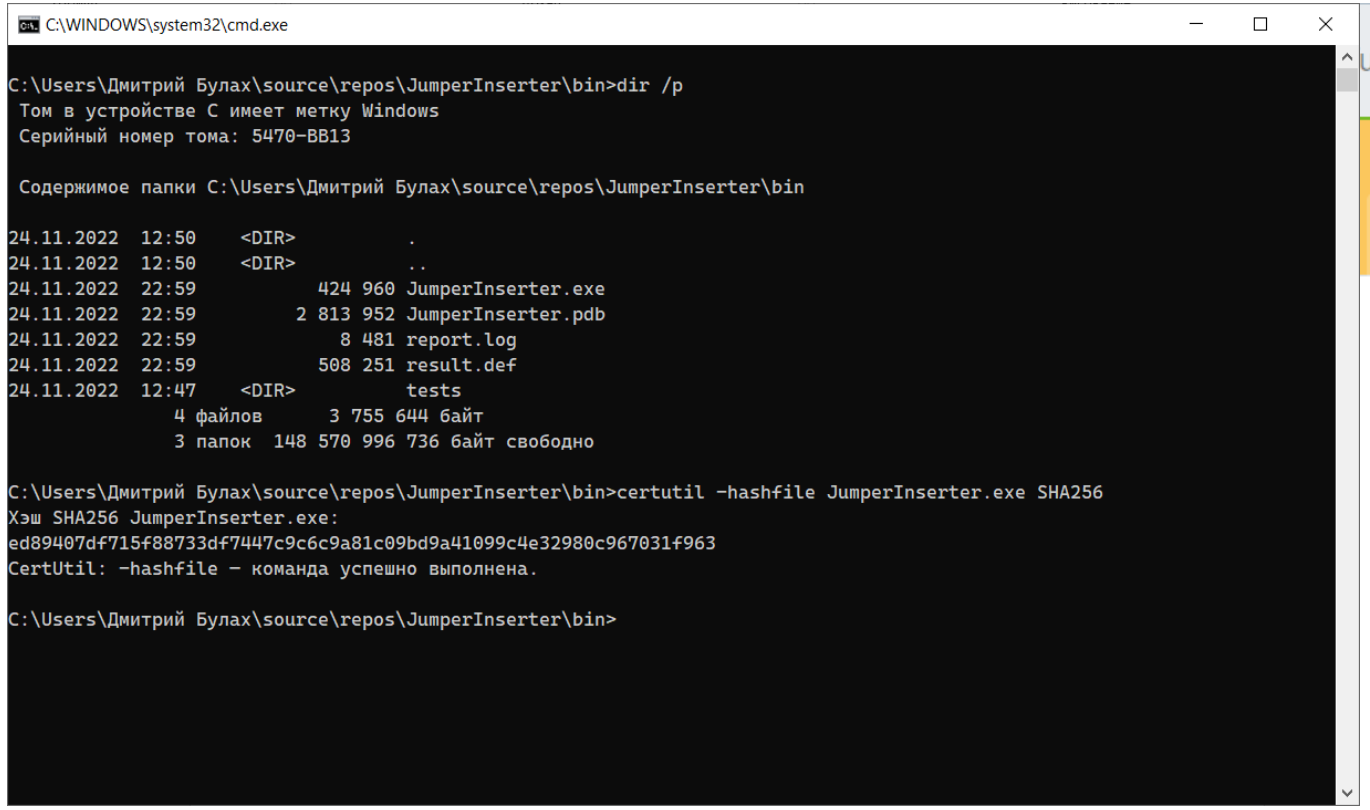


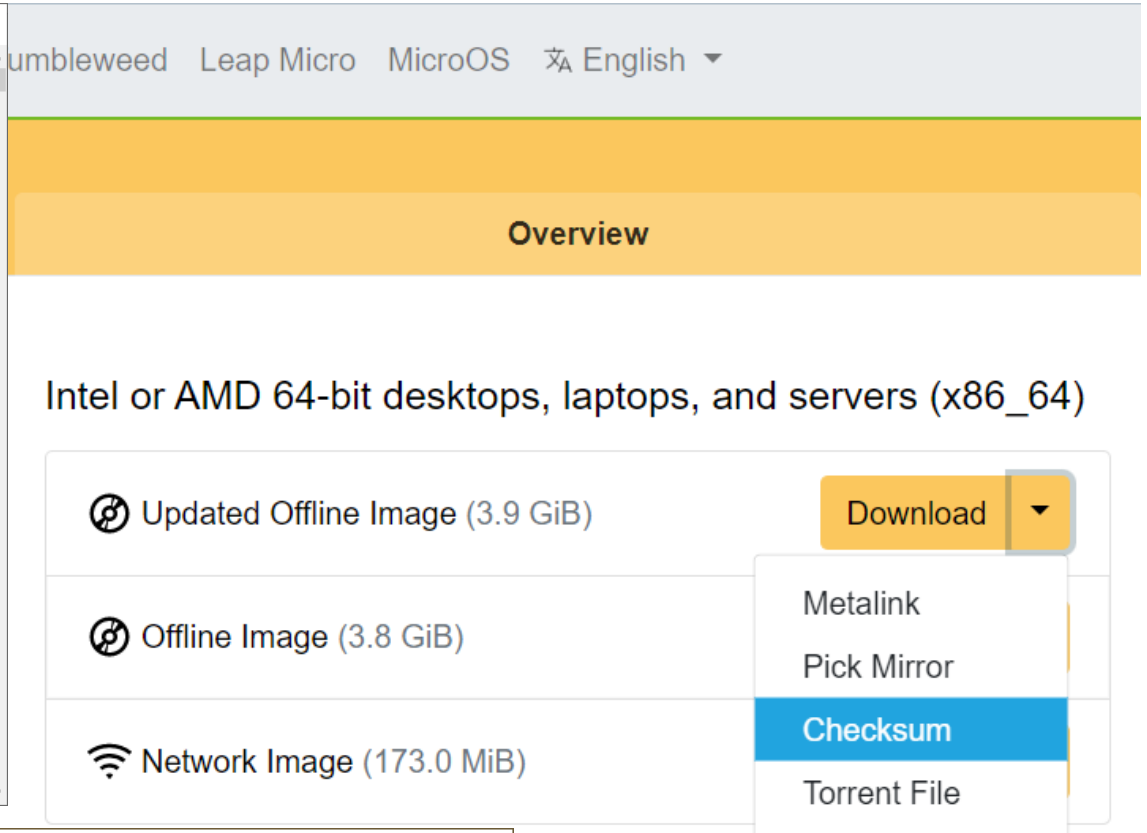
Полученная контрольная сумма:

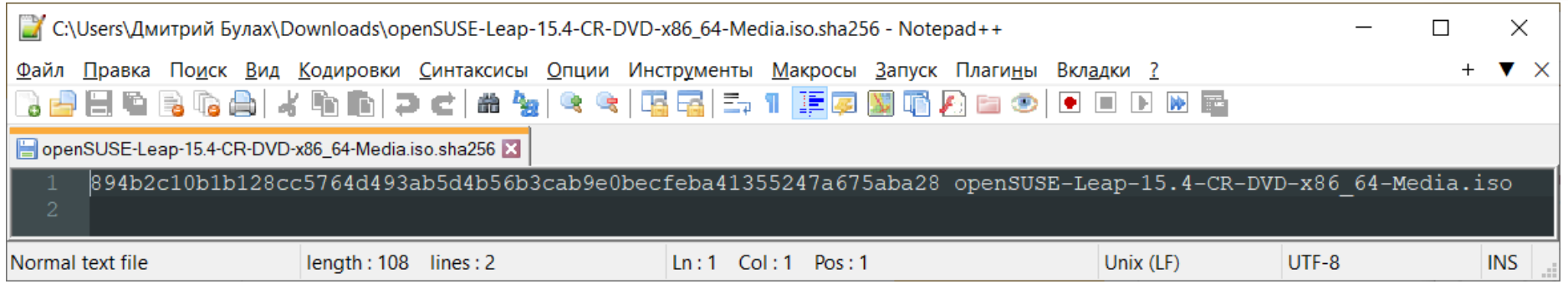
11001011 = 203 = 0xCB

Данные, для которых считается CRC

Хэширование данных







Корректирующие коды: код Хэмминга (1)

Кодируемое сообщение: «ЭН»

Битовое представление сообщения: 11011101 11001101

1	1	0	1	1	1	0	1	1	1	0	0	1	1	0	1

Расширяем последовательность, добавляя пустые ячейки на позиции, равные степеням двойки:

2^0	2^1	2^2	2^3				2^4												
		1		1	0	1		1	1	0	1	1	1	0		0	1	1	0	1

Инициализируем эти значения нулями:

0	**0**	1	**0**	1	0	1	**0**	1	1	0	1	1	1	0	**0**	0	1	1	0	1

Корректирующие коды: код Хэмминга (2)

0	0	1	0	1	0	1	0	1	1	0	1	1	1	0	0	0	1	1	0	1

В каждой из добавленных позиций (контрольных разряд, КР) считаем его значение:

1. КР считает число единиц в подконтрольных ему позициях с определённым шагом (отмечено зелёным);
2. шаг проверки – номер шага, начиная с позиции КР;
3. записываем в КР значение «1», если число единиц нечётно

			1		1	0	1		1	1	0	1	1	1	0		0	1	1	0	1
Шаг 1																					
Шаг 2																					
Шаг 3																					
Шаг 4																					

Корректирующие коды: код Хэмминга (3)

Исходное сообщение:

1	0	1	0	1	0	1	1	1	1	0	1	1	1	0	1	0	1	1	0	1

Вносим в него ошибку (ошибка вносится в 13 бит):

1	0	1	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	1	0	1

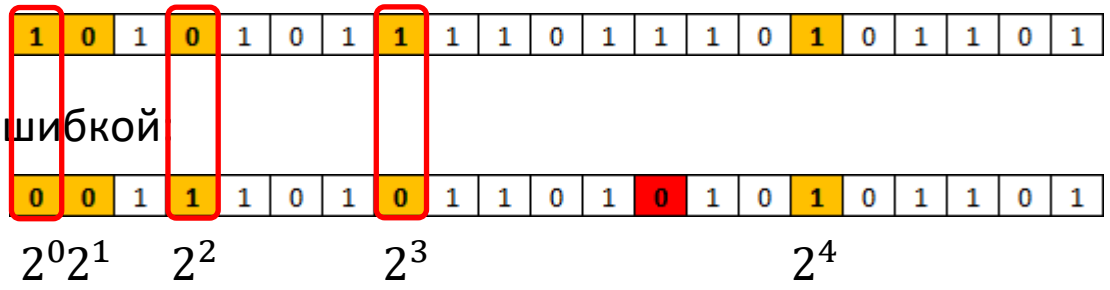
Пересчитываем значения контрольных разрядов:

			1		1	0	1		1	1	0	1	0	1	0		0	1	1	0	1
Шаг 1																					
Шаг 2																					
Шаг 3																					
Шаг 4																					
Шаг 5																					

Корректирующие коды: код Хэмминга (4)

Исходное сообщение:

Сообщение с ошибкой:



Выясняем номер бита, в котором произошла ошибка:

$$2^0 + 2^2 + 2^3 = 1 + 4 + 8 = 13$$