Лекция 02

БГТУ, ФИТ, ПОИТ, 2 семестр, Языки программирования

Структура языка программирования

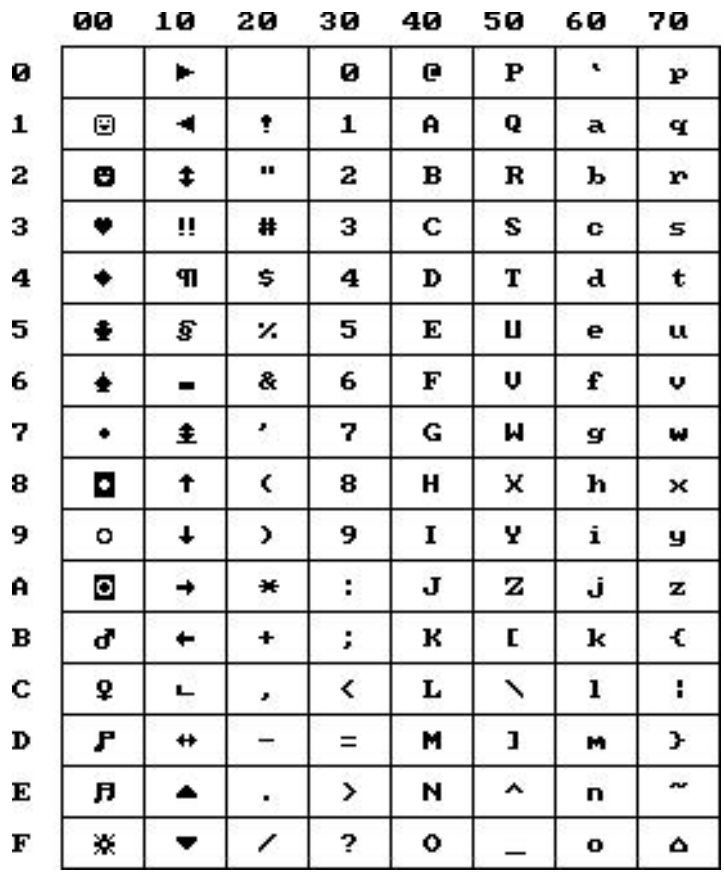
План лекции:

- понятие кодировки;
- кодировка ASCII;
- кодировка Unicode;
- прямой (LE) и обратный (BE) порядок байт;
- маркер последовательности байтов (BOM).

1. **Спецификация системы программирования:** набор требований к системе программирования, достаточный для ее разработки.

2. **Алфавит языка:** набор символов, разрешенных к использованию языком. Основывается на одной из кодировок.
   Согласно RFC 2278 кодировка (*charset*) определяется как комбинация набора символов и схемы кодирования.

3. **ASCII: American Standard Code for Information Interchange —** американский стандартный код для обмена информацией. ASCII — 8-битная кодировка для представления десятичных цифр, латинского и национального алфавитов, знаков препинания и управляющих символов.



4. **ASCII:** альтернативная кодировка (CP866)

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
8.	А 410	Б 411	В 412	Г 413	Д 414	Е 415	Ж 416	З 417	И 418	Й 419	К 41A	Л 41B	М 41C	Н 41D	О 41E	П 41F
9.	Р 420	С 421	Т 422	У 423	Ф 424	Х 425	Ц 426	Ч 427	Ш 428	Щ 429	Ъ 42A	Ы 42B	Ь 42C	Э 42D	Ю 42E	Я 42F
A.	а 430	б 431	в 432	г 433	д 434	е 435	ж 436	з 437	и 438	й 439	к 43A	л 43B	м 43C	н 43D	о 43E	п 43F
B.	▒ 2591	▓ 2592	▓ 2593	┃ 2502	┤ 2524	╡ 2561	╢ 2562	╖ 2556	╕ 2555	╣ 2563	║ 2551	╗ 2557	╝ 255D	╜ 255C	╛ 255B	┐ 2510
C.	└ 2514	┴ 2534	┬ 252C	├ 251C	─ 2500	┼ 253C	╞ 255E	╟ 255F	╚ 255A	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256C	╧ 2567
D.	╨ 2568	╤ 2564	╥ 2565	╙ 2559	╘ 2558	╒ 2552	╓ 2553	╫ 256B	╪ 256A	┘ 2518	┌ 250C	▉ 2588	▄ 2584	▌ 258C	▐ 2590	▀ 2580
E.	р 440	с 441	т 442	у 443	ф 444	х 445	ц 446	ч 447	ш 448	щ 449	ъ 44A	ы 44B	ь 44C	э 44D	ю 44E	я 44F
F.	Ё 401	ё 451	Є 404	є 454	Ї 407	ї 457	Ў 40E	ў 45E	° B0	∙ 2219	· B7	√ 221A	№ 2116	¤ A4	■ 25A0	A0

5. **ASCII:** русская Windows-кодировка (синоним CP1251, Windows-1251)

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
8.	Ђ 402	Ѓ 403	‚ 201A	ѓ 453	„ 201E	… 2026	† 2020	‡ 2021	€ 20AC	‰ 2030	Љ 409	‹ 2039	Њ 40A	Ќ 40C	Ћ 40B	Џ 40F
9.	ђ 452	‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014		™ 2122	љ 459	› 203A	њ 45A	ќ 45C	ћ 45B	џ 45F
A.	A0	Ў 40E	ў 45E	J 408	¤ A4	Ґ 490	¦ A6	§ A7	Ё 401	© A9	Є 404	« AB	¬ AC	AD	® AE	Ї 407
B.	° B0	± B1	I 406	і 456	ґ 491	µ B5	¶ B6	· B7	ё 451	№ 2116	є 454	» BB	j 458	S 405	s 455	ї 457
C.	А 410	Б 411	В 412	Г 413	Д 414	Е 415	Ж 416	З 417	И 418	Й 419	К 41A	Л 41B	М 41C	Н 41D	О 41E	П 41F
D.	Р 420	С 421	Т 422	У 423	Ф 424	Х 425	Ц 426	Ч 427	Ш 428	Щ 429	Ъ 42A	Ы 42B	Ь 42C	Э 42D	Ю 42E	Я 42F
E.	а 430	б 431	в 432	г 433	д 434	е 435	ж 436	з 437	и 438	й 439	к 43A	л 43B	м 43C	н 43D	о 43E	п 43F
F.	р 440	с 441	т 442	у 443	ф 444	х 445	ц 446	ч 447	ш 448	щ 449	ъ 44A	ы 44B	ь 44C	э 44D	ю 44E	я 44F

6. **ASCII:** Visual Studio хранит содержимое cpp-файлов и h-файлов в кодировке Windows-1251 (CP1251).

```
// LP_Lab03.cpp: определяет точку входа для консольного приложения.

#include "stdafx.h"
#include <stdlib.h>
#include <iostream>

int  x = 0x12345678;

int _tmain(int argc, _TCHAR* argv[])
{
    printf("%x%x%x%x\n", *((char*)&x),*((char*)&x+1), *((char*)&x+2), *((char*)&x+3));
    system("pause");
    return 0;
}
```

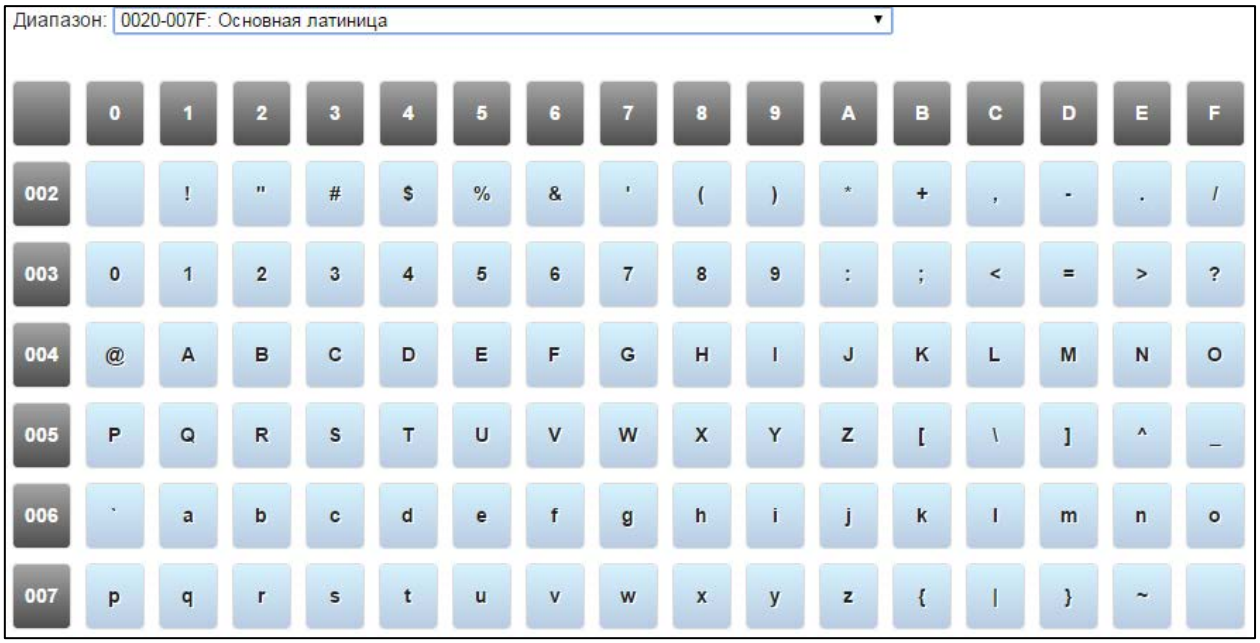
00000000	00 01 02 03	04 05 06 07	08 09 0a 0b	0c 0d 0e 0f	
00000000	2f 2f 20 4c	50 5f 4c 61	62 30 33 2e	63 70 70 3a	// LP_Lab03.cpp:
00000010	20 ee ef f0	e5 e4 e5 eb	ff e5 f2 20	f2 ee f7 ea	?????????? ????
00000020	f3 20 e2 f5	ee e4 e0 20	e4 eb ff 20	ea ee ed f1	? ????????? ????
00000030	ee eb fc ed	ee e3 ee 20	ef f0 e8 eb	ee e6 e5 ed	??????? ????????
00000040	e8 ff 2e 0d	0a 0d 0a 23	69 6e 63 6c	75 64 65 20	??.....#include
00000050	22 73 74 64	61 66 78 2e	68 22 0d 0a	23 69 6e 63	"stdafx.h"..#inc
00000060	6c 75 64 65	20 3c 73 74	64 6c 69 62	2e 68 3e 0d	lude <stdlib.h>.
00000070	0a 23 69 6e	63 6c 75 64	65 20 3c 69	6f 73 74 72	.#include <iostr
00000080	65 61 6d 3e	0d 0a 0d 0a	69 6e 74 20	20 78 20 3d	eam>....int x =
00000090	20 30 78 31	32 33 34 35	36 37 38 3b	0d 0a 0d 0a	0x12345678;....
000000a0	69 6e 74 20	5f 74 6d 61	69 6e 28 69	6e 74 20 61	int _tmain(int a
000000b0	72 67 63 2c	20 5f 54 43	48 41 52 2a	20 61 72 67	rgc, _TCHAR* arg
000000c0	76 5b 5d 29	0d 0a 7b 0d	0a 09 70 72	69 6e 74 66	v[])..{...printf
000000d0	28 22 25 78	25 78 25 78	25 78 5c 6e	22 2c 20 2a	("%x%x%x%x\n", *
000000e0	28 28 63 68	61 72 2a 29	26 78 29 2c	2a 28 28 63	((char*)&x),*((c
000000f0	68 61 72 2a	29 26 78 2b	31 29 2c 20	2a 28 28 63	har*)&x+1), *((c
00000100	68 61 72 2a	29 26 78 2b	32 29 2c 20	2a 28 28 63	har*)&x+2), *((c
00000110	68 61 72 2a	29 26 78 2b	33 29 29 3b	0d 0a 09 73	har*)&x+3));...s
00000120	79 73 74 65	6d 28 22 70	61 75 73 65	22 29 3b 0d	ystem("pause");.
00000130	0a 09 72 65	74 75 72 6e	20 30 3b 0d	0a 7d 0d 0a	..return 0;..}..

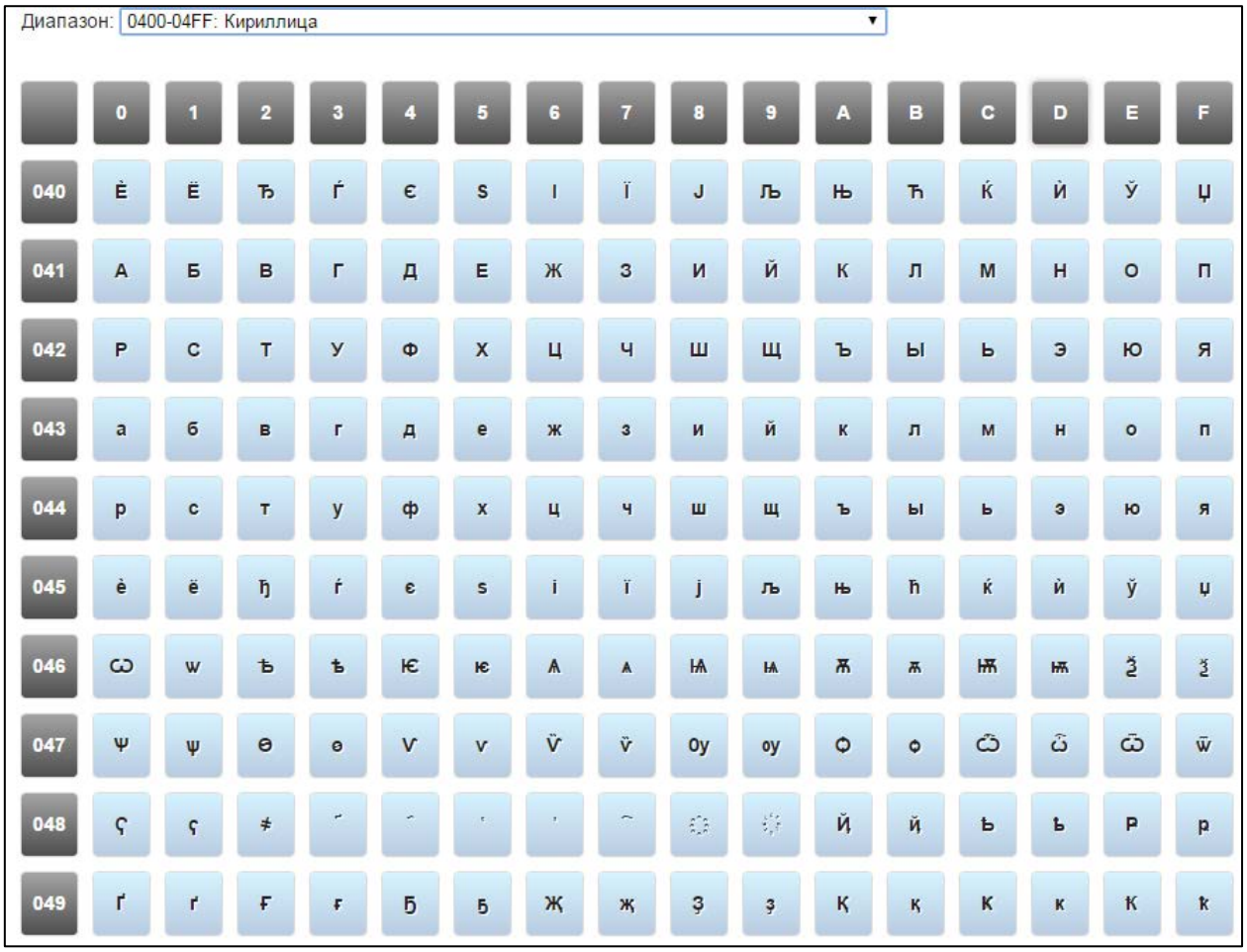
7. **UNICODE:** Стандарт предложен в 1991 году некоммерческой организацией Unicode Consortium, 1991, ISO/IEC 10646, последняя версия 10.0.0 (2017).
   Юникод – это стандарт кодирования символов, позволяющий представить знаки почти всех письменных языков, состоит из 2х разделов:
   - UCS - universal character set (универсальный набор символов);
   - UTF - Unicode transformation format (семейство кодировок).

   Принято обозначение U+xxx, где xxx- число в шестнадцатеричном формате.

8. **UNICODE:** UCS расположены в 17 плоскостях (0-16), 2^{16} (65 536) символов в каждой плоскости, плоскость 0 – основная (основные символы), 1-14 – дополнительные, 15-16 – для частного использования.

9. **UNICODE:** http://foxtools.ru/Unicode



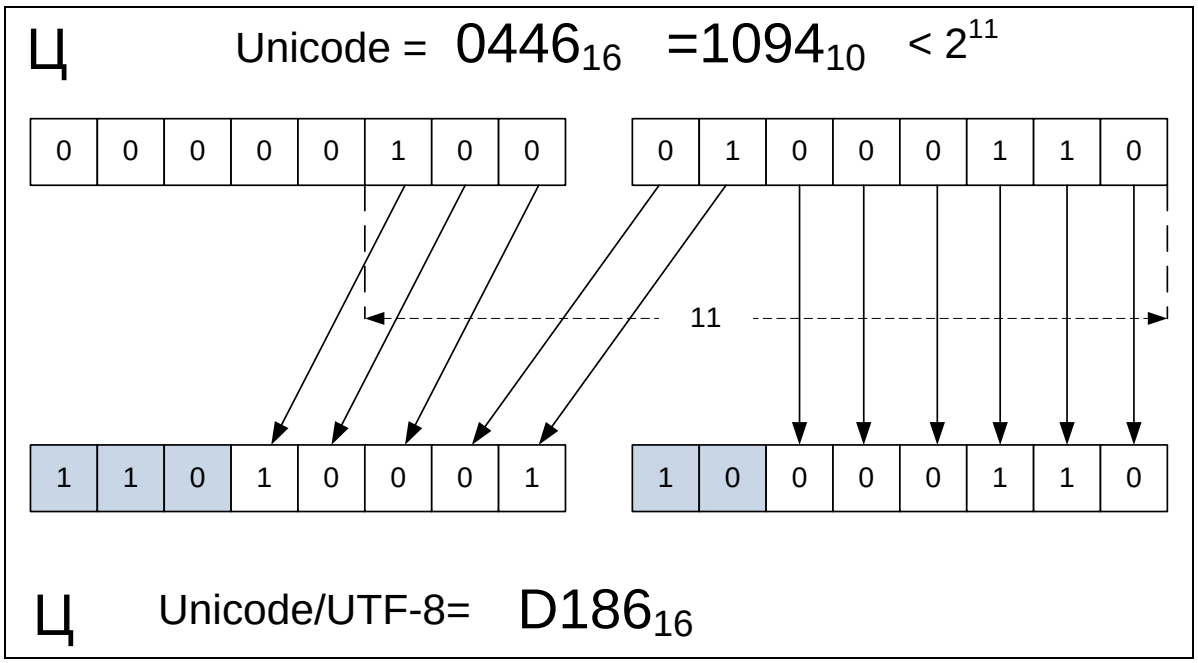


10. **UNICODE:** UTF-8

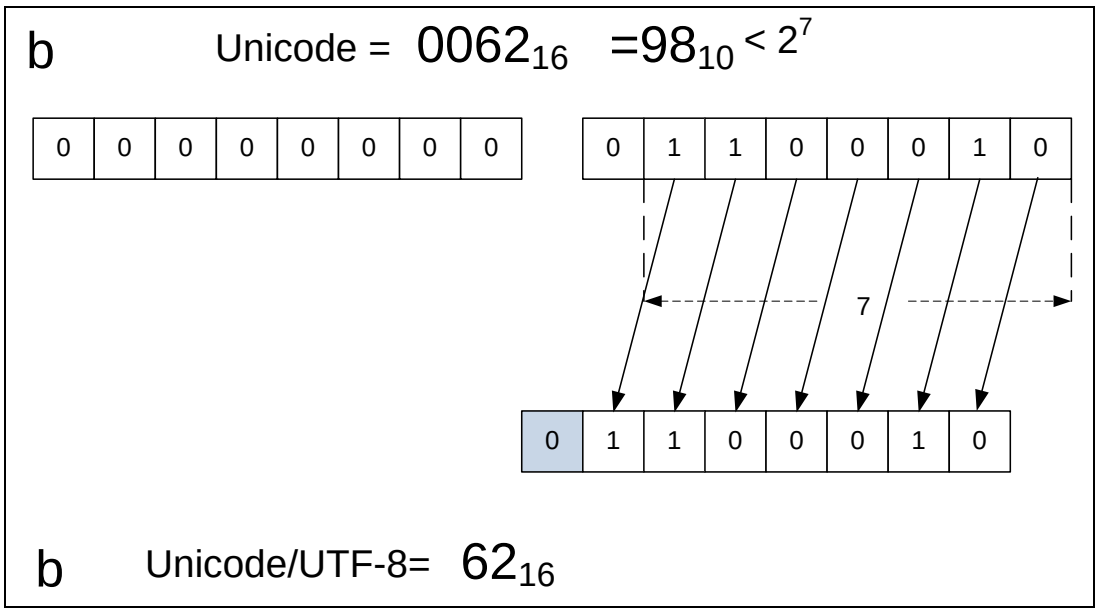
UTF-8 — представление Юникода, обеспечивающее совместимость со старыми системами, использовавшими 8-битные символы.

Алгоритм кодирования в UTF-8:

1) определить количество октетов (октет: 8 битов или 1 байт) – в какой диапазон значений попадает количество значащих символов (7, 11, 16, 21, 26, 31);
2) подготовить старшие биты первого октета:
    a. 0xxxxxxx для одного октета;
    b. 110xxxxx – двух;
    c. 1110xxxx - трех и т.д..
    d. 10xxxxxx - для остальных октетов;
3) заполнить оставшиеся биты (обозначены как x) в октетах кодом символа Юникода в двоичном виде. Начать с младших битов, поставив их в младшие биты последнего октета кода. И так далее, пока все биты кода символа не будут перенесены в свободные биты октетов.



0446_{16} = 4*16² + 4*16 + 6 = 1094_{10}



11. **UNICODE:** UTF-8

0x00000000—0x0000007F: 0xxxxxxx

0x00000080—0x000007FF: 110xxxxx 10xxxxxx

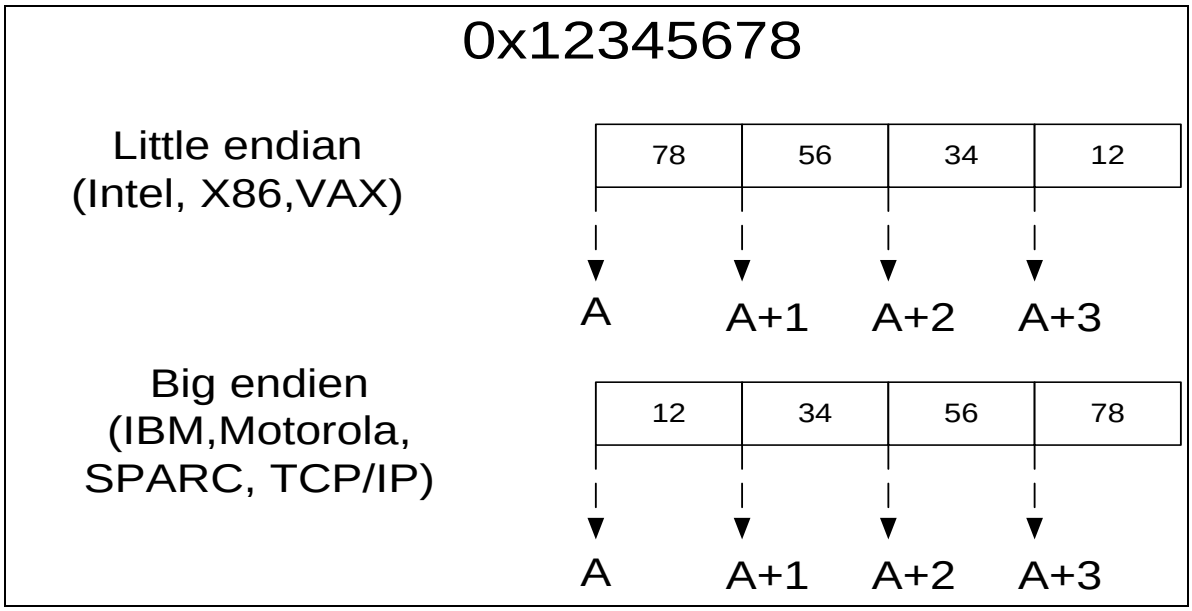
0x00000800—0x0000FFFF: 1110xxxx 10xxxxxx 10xxxxxx

0x00010000—0x001FFFFF: 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

12. **UNICODE:** UTF-16

    В UTF-16 символы кодируются двухбайтовыми словами с использованием всех возможных диапазонов значений (от 0 до FFFF16). При этом можно кодировать символы Unicode в диапазонах 000016..D7FF16 и E00016..10FFFF16. Исключенный отсюда диапазон D80016..DFFF16 используется для кодирования так называемых суррогатных пар — символов, которые кодируются двумя 16-битными словами.

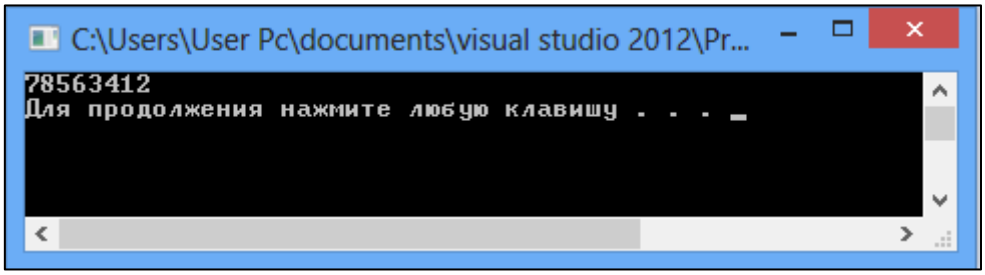
13. **LE** (Little endian order, прямой порядок, от младшего к старшему), BE (Big endian order, обратный порядок, от старшего к младшему).

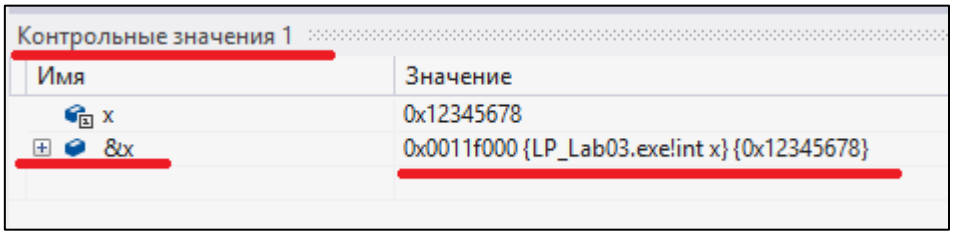


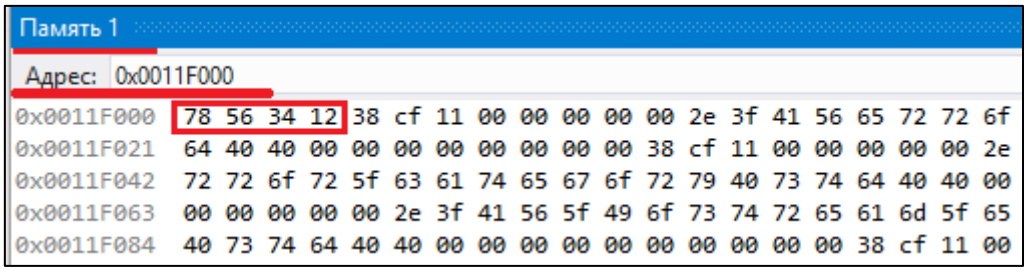
```c
#include "stdafx.h"
#include <stdlib.h>
#include <iostream>

int  x = 0x12345678;

int _tmain(int argc, _TCHAR* argv[])
{
    printf("%x%x%x%x\n", *((char*)&x),*((char*)&x+1), *((char*)&x+2), *((char*)&x+3));
    system("pause");
    return 0;
}
```







14. **UNICODE: BOM** (Byte Order Mark)

    Для определения формата представления Юникода в начало текстового файла записывается сигнатура (обозначение) — символ U+FEFF — маркер последовательности байтов.

Представление кодировки маркера последовательности байтов:

Кодировка	Представление (hex)
UTF-8	EF BB BF
UTF-16 (BE)	FE FF
UTF-16 (LE)	FF FE
UTF-32 (BE)	00 00 FE FF
UTF-32 (LE)	FF FE 00 00

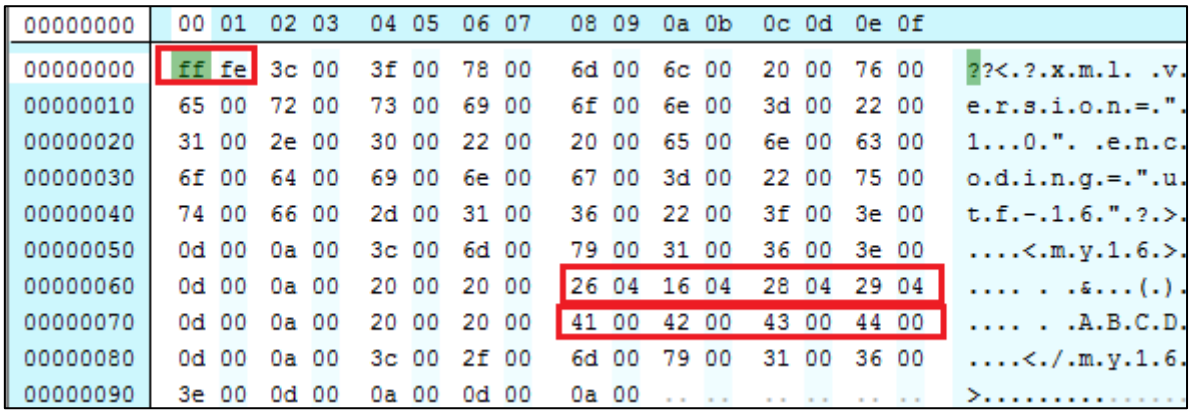
15. **UNICODE:** UTF-8

```
<?xml version="1.0" encoding="utf-8"?>
<my>
    ц абвгд
    b abcde
</my>
```

```
  00 01 02 03  04 05 06 07  08 09 0a 0b  0c 0d 0e 0f
  ef bb bf 3c  3f 78 6d 6c  20 76 65 72  73 69 6f 6e     <?xml version
  3d 22 31 2e  30 22 20 65  6e 63 6f 64  69 6e 67 3d  ="1.0" encoding=
  22 75 74 66  2d 38 22 3f  3e 0d 0a 3c  6d 79 3e 0d  "utf-8"?>..<my>.
  0a 09 d1 86  20 d0 b0 d0  b1 d0 b2 d0  b3 d0 b4 0d  .. ц  а б в г д.
  0a 09 62 20  61 62 63 64  65 0d 0a 3c  2f 6d 79 3e  ..b abcde..</my>
  .. .. .. ..  .. .. .. ..  .. .. .. ..  .. .. .. ..  ................
```

16. **UNICODE:** UTF-16

```
<?xml version="1.0" encoding="utf-16"?>
<my16>
  ЦЖШЩ
  ABCD
</my16>
```



00000000	00 01	02 03	04 05	06 07	08 09	0a 0b	0c 0d	0e 0f	
00000000	ff fe	3c 00	3f 00	78 00	6d 00	6c 00	20 00	76 00	??<.?.x.m.l. .v.
00000010	65 00	72 00	73 00	69 00	6f 00	6e 00	3d 00	22 00	e.r.s.i.o.n.=."
00000020	31 00	2e 00	30 00	22 00	20 00	65 00	6e 00	63 00	1...0.". .e.n.c.
00000030	6f 00	64 00	69 00	6e 00	67 00	3d 00	22 00	75 00	o.d.i.n.g.=."u.
00000040	74 00	66 00	2d 00	31 00	36 00	22 00	3f 00	3e 00	t.f.-.1.6."?.>.
00000050	0d 00	0a 00	3c 00	6d 00	79 00	31 00	36 00	3e 00	....<.m.y.1.6.>.
00000060	0d 00	0a 00	20 00	20 00	26 04	16 04	28 04	29 04	.... . .&...(.).
00000070	0d 00	0a 00	20 00	20 00	41 00	42 00	43 00	44 00	.... . .A.B.C.D.
00000080	0d 00	0a 00	3c 00	2f 00	6d 00	79 00	31 00	36 00	....<./.m.y.1.6.
00000090	3e 00	0d 00	0a 00	0d 00	0a 00	.. ..	.. ..	.. ..	>..............

```
<?xml version="1.0" encoding="utf-16BE"?>
<my16>
  ЦЖШЩ
  ABCD
</my16>
```

000000	00 01	02 03	04 05	06 07	08 09	0a 0b	0c 0d	0e 0f	
000000	fe ff	00 3c	00 3f	00 78	00 6d	00 6c	00 20	00 76	??.<.?.x.m.l. .v
000010	00 65	00 72	00 73	00 69	00 6f	00 6e	00 3d	00 22	.e.r.s.i.o.n.=."
000020	00 31	00 2e	00 30	00 22	00 20	00 65	00 6e	00 63	.1...0.". .e.n.c
000030	00 6f	00 64	00 69	00 6e	00 67	00 3d	00 22	00 75	.o.d.i.n.g.=."u
000040	00 74	00 66	00 2d	00 31	00 36	00 42	00 45	00 22	.t.f.-.1.6.B.E."
000050	00 3f	00 3e	00 0d	00 0a	00 3c	00 6d	00 79	00 31	.?.>.....<.m.y.1
000060	00 36	00 3e	00 0d	00 0a	00 20	00 20	04 26	04 16	.6.>..... . .&..
000070	04 28	04 29	00 0d	00 0a	00 20	00 20	00 41	00 42	.(.)..... . .A.B
000080	00 43	00 44	00 0d	00 0a	00 3c	00 2f	00 6d	00 79	.C.D.....<./.m.y
000090	00 31	00 36	00 3e	00 0d	00 0a	00 0d	00 0a	00 0d	.1.6.>..........
0000a0	00 0a	00 0d	00 0a	00 0d	00 0a	.. ..	.. ..	.. ..	....