3. Минимальная единица хранения информации-бит

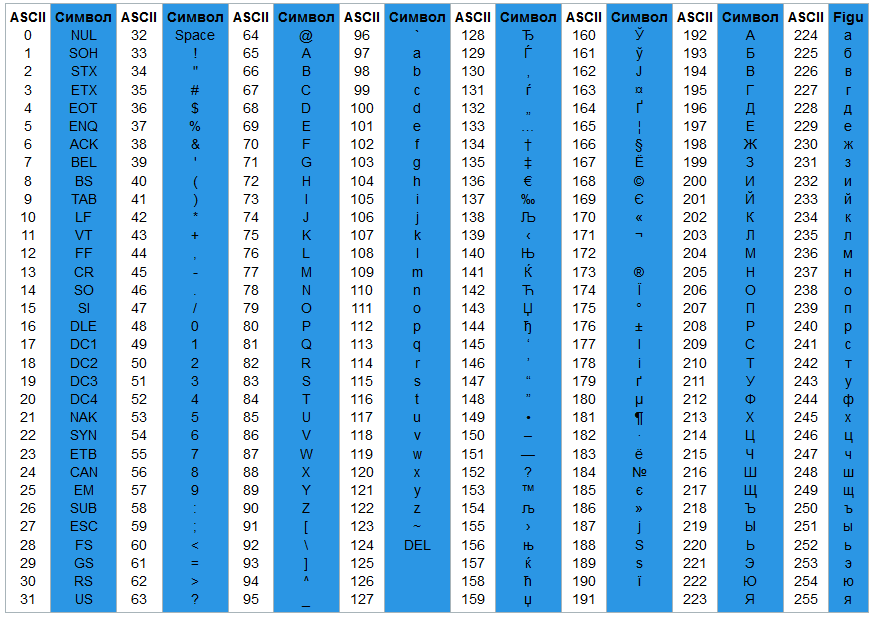
Минимальная адресуемая единица хранения информации-байт

4. Бит может принимать значения 0 и 1

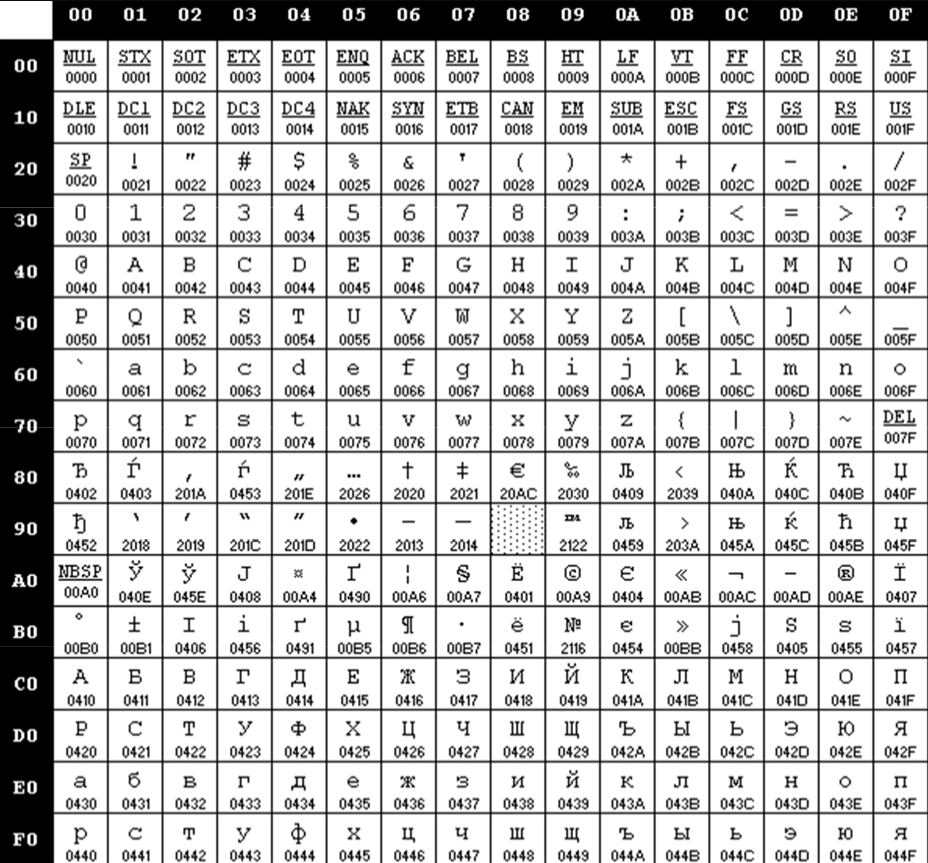
5. В одном байте 8 битов

6.

|  |  |  |
| --- | --- | --- |
| десятичная | двоичная | шестнадцатеричная |
| 0 | 0000 0000 | 0 |
| 1 | 0000 0001 | 1 |
| 2 | 0000 0010 | 2 |
| 3 | 0000 0011 | 3 |
| 4 | 0000 0100 | 4 |
| 5 | 0000 0101 | 5 |
| 6 | 0000 0110 | 6 |
| 7 | 0000 0111 | 7 |
| 8 | 0000 1000 | 8 |
| 9 | 0000 1001 | 9 |
| 10 | 0000 1010 | A |
| 11 | 0000 1011 | B |
| 12 | 0000 1100 | C |
| 13 | 0000 1101 | D |
| 14 | 0000 1110 | E |
| 15 | 0000 1111 | F |
| 16 | 0010 000 | 10 |
| 17 | 0010 001 | 11 |
| 18 | 0010 010 | 12 |
| 19 | 0010 011 | 13 |
| 20 | 0010 100 | 14 |

7.

Ascii



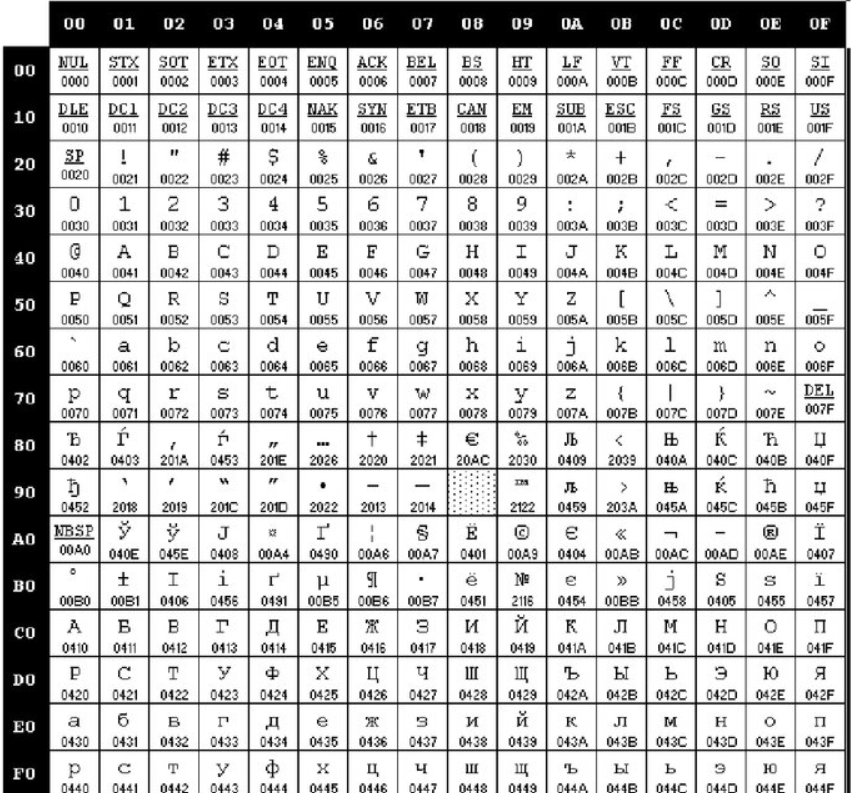
Win-1251

16. в какой кодировке представлен файл (\*.cpp) транслятора C++. Utf-8

17. 0042 0059 0053 004b 004f

0062 0079 0073 006k 006f

18. С помощью функций tolower/toupper можно переводить символы из нижнего в верхний реестр и наоборот. После этого используя память можно посмотреть код элемента или с помощью setlocale  
также можно к номеру прибавить/отнять 20 и получить заглавную/строчную букву



Windows-1251

9.