

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Hexadecimal-to-ASCII conversion table

```
// expand Huffman-encoded input from standard input and write to standard output
public static void expand() {

    // read in Huffman trie from input stream
    Node root = readTrie();

    // number of bytes to write
    int length = BinaryStdIn.readInt();

    // decode using the Huffman trie
    for (int i = 0; i < length; i++) {
        Node x = root;
        while (!x.isLeaf()) {
            boolean bit = BinaryStdIn.readBoolean();
            if (bit) x = x.right;
            else x = x.left;
        }
        BinaryStdOut.write(x.ch);
    }
    BinaryStdOut.close();
}
```