CSVdelim.exe - Change CSV file delimiter MS Windows cmd tool
(c) Markus Gnam 2023. **Version 1.0.4.0** 20231004
USAGE: CSVdelim [options] <file>

Change an input field delimiter to an output field delimiter
according to CSV rules (quoted with double quotes if needed).
The input file has to comply with CSV rules.
The output file is formatted according to CSV rules.

E.g. change ; to , with output file ending .csv:
CSVdelim --from=; --to=, inputfile.ssv => inputfile.csv
The Quote char sign (for in and out) is double quote.
The default for tab separated files is to use no quotes.
For all options and more details type "CSVdelim --help"

OPTIONS: Details about all options and their defaults:

-f, **--from**=FIELDSEP	Field separator of the input file [default: auto]
	E.g. commma: **--from=,** or --from=csv
	E.g. semicolon: **--from=;** or --from=ssv
	E.g. tabulator: **--from=\t** or --from=tab
	auto: Assume FIELDSEP from file ending, see Notes.
-t, **--to**=FIELDSEP	Field separator of the output file [default: auto]
	E.g. tabulator: **--to=\t** or --to=tab
	E.g. semicolon: **--to=;** or --to=ssv
	E.g. commma: **--to=,** or --to=csv
	auto: Assume FIELDSEP from file ending, see Notes.
-i, **--inquote**={0,1}	The input unquoting method [default: auto]
	--inquote=0: Don't use any quotes (don't unquote)
	--inquote=1: Use quotes (unquote)
-q, **--quote**={0,1,2}	The output quoting method [default: auto]
	--quote=0: Don't use any quotes (don't quote)
	--quote=1: Smart quotes (quote only if needed)
	--quote=2: Quote always (quote all output fields)
-o, **--out**=FILE	Output file [default: auto]
	If specified, use this output file, else use auto
--trim={0,1}	Trim the output fields [default: 0]
-?, **--help**	Display this help and exit

Notes:
You can use -, -- or / as first option character(s).
E.g. -inquote=1, --inquote=1, /inquote=1 are all valid alternatives.
{0,1} means: Use "0" for No (False) or "1" for Yes (True).
For these {0,1} options omitting these values means Yes.
E.g. option -i, --inquote={0,1}
You can use either --inquote=1 or simply --inquote
For the short option you can use -i, -i=1 or -i 1

Auto settings:
These settings can be changed at any time by setting them manually.

--from:	File ending == csv => --from=,
	File ending == ssv => --from=;
	File ending == tab or txt or tsv => --from=\t
--to:	File ending == csv => --to=\t
	File ending == ssv => --to=\t
	File ending == tab or txt or tsv => --to=,
--out:	Use input file base name with the ending:
	--to=\t => Ending .txt; --to=, => Ending .csv
	--to=; => Ending .ssv; others: => Ending .out

Explanation:
The CSV rules are given in **RFC 4180**: https://tools.ietf.org/html/rfc4180

A good explanation can be found here:
http://dictionnaire.sensagent.leparisien.fr/Comma%20separated%20lists/en-en/

<Begin citation>CSV is a delimited data format that has fields/columns separated by the comma character and records/rows separated by newlines. **Fields that contain a special character (comma, newline, or double quote), must be enclosed in double quotes. [...] If a field's value contains a double quote character it is escaped by placing another double quote character next to it.** [...] Fields are separated by commas (although in locales where the comma is used as a decimal point, the semicolon is used instead as a delimiter, inducing some drawbacks when CSV files are exchanged e.g. between France and USA).
1997,Ford,E350

[...] Fields may always be enclosed within double-quote characters, whether necessary or not. "1997","Ford","E350"<End citation>

Examples:
Example 1:
https://raw.githubusercontent.com/uniVocity/univocity-parsers/master/src/test/resources/examples/example.csv

example.csv:
```
Year,Make,Model,Description,Price
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,"Venture ""Extended Edition""","",4900.00
1996,Jeep,Grand Cherokee,"MUST SELL!
air, moon roof, loaded",4799.00
1999,Chevy,"Venture ""Extended Edition, Very Large""",,5000.00
,,"Venture ""Extended Edition""","",4900.00
```

:: example1.cmd: Change comma to semicolon according to CSV rules:
```
CSVdelim example.csv --from=, --to=; --inquote=1 --quote=1 --out=out.ssv
```

⇨ Out file "out.ssv" successfully created for FS=, and OFS=;

out.ssv:
```
Year;Make;Model;Description;Price
1997;Ford;E350;ac, abs, moon;3000.00
1999;Chevy;"Venture ""Extended Edition""";;4900.00
1996;Jeep;Grand Cherokee;"MUST SELL!
air, moon roof, loaded";4799.00
1999;Chevy;"Venture ""Extended Edition, Very Large""";;5000.00
;;"Venture ""Extended Edition""";;4900.00
```

Example 2:
Although this program supports multiline fields, it isn't recommended to use them. There are problems with other programs that don't support them. The Tab sign (\t) is the recommended unambiguous output delimiter which doesn't need any quoting since a tab should not appear inside any field.

:: example2.cmd: Change comma to TAB (\t) with no output quotes:
```
CSVdelim names.csv --from=, -to=\t --inquote=1 --quote=0 --out=out.tab
```

⇨ Out file "out.tab" successfully created for FS=, and OFS=\t