Purpose:

It is requested to design a library that processes the arguments received by command line using a callback

The design must follow the next steps:

a. It must can processes options. An option consist in two arguments, the first begin with "-" and is called key, the second is called value. An example with two options could be: The key maxclients with the value 4 and the key path with the value C:/web, in this case it could be: webserver -maxclients 4 -path C:\WEB
b. It must can processes parameters, a parameter is an isolated argument that do not begin with an "-" . An example with an option and two parameters could be: copyfile -isverify 1 archivo1.c archivo2.c
c. The command line parser must follow this prototype:
   typedef int (*pCallback) (char *, char*, void *);
   int parseCmdLine(int argc, char *argv[], pCallback p, void *userData);
d. For the user of parseCmdLine () to use it, it must first define a callback function. This function is called every time parseCmdLine () finds an option or a parameter.
   The callback function should return 1 if the interpretation of the options or parameters was correct, and 0 to indicate that the processing should stop because an invalid option or parameter was found. The prototype is:

   int parseCallback(char *key, char *value, void *userData);

e. The parser is executed with the parseCmdLine () function. Every time the parser finds an option, it must call the callback passing in key the value of the found key, and in value the value found.
   Every time you find a parameter, you must pass the value NULL in key, and the value of the found parameter in value. If the processing of all options and parameters was successful and no form errors were found (see limit cases below), you must return on your behalf the sum of the number of options and parameters found. Otherwise (-1).
f. It is recommended to use typedef for function pointers.
g. A test bench must be made and delivered to test the module to limit cases (what is a limit case?). An example: what happens if we send an option with a key but no value? An idea: we must think about all possible situations and know how to deal with them. Discuss all possible limit cases in class. How should parseCmdLine () behave in such cases?