1 Input file format

**The first line is an argument line surounded by curly braces and denotes the information for the style used for example Background=FFFFFF,Font=TimesNewRoman,Size=14 the next line in curly brackets how many segmentations will a particular page section have, for example having 3, means that there will be three collumns. The collumns are seperated in the file by use of an ! with no other text in line, after the end of the last section a section break is not needed but rather another curly bracket is expected to indicated the collumns in the next section. For example having:
{Backgound=FFFFFF,Font=Arial,Size=12}
{1}
!ABC
{3}
!D
!E
!f
will create a document, where the first line is the centered text ABC and below it there will be three collumns in the first there will be the text D, in the next E and the last f; all without a bulletpoint. If the text is to be written without a bulletpoint, then the beggining of the line should begin with @. If these symbols are to be used as literals, they must be escaped by using a backslash \. If a line begins with an exclamation mark, then it is unnderstood that that line should be a heading and, by default have bold text and should not contain a bulletpoint.
For the stylization of the text the use of square brackets are used. In the text the contents inside the square brackets declare the stylization options that are used. The formatting is to be applied for the whole line, as such, the style declaration is to be put at the begining of the line, following *!, @* if they were used. For example: `[Bold]`, `![Italic,Size=10pt,Color=red]`

2 Program workflow

The program works by reading the input file, passed by a commandline argument, line by line and interpreting each of the lines and converting the read String, removeing the special characters, and parsing the String to be used as the text in the HTML file. It also splits the data and creates the needed information, to create the Cascadeing Sheets document. After all the lines have been proccessed, it writes the contents to the output files.

3 Headers

3.1 Formatter

This header houses the structure Element which contains a string, for the text that is to be displayed along with a char array that contains it's properties. A function returning a string is defined *parse(char*)*, which gets the read line as an argument and the function goes through it and determines what is this line declaring(Title, bulletpoint, raw text) and what properties does it have(font, typefont, size, color, etc.). The char array is formatted so that the fist bit of the first element of the array indicates wheather or not the text is of the Bold typefont, the second bit, if it is italicized, the third if it is underlined, the fourth and fifth show whether the text is a title, and whether a bulletpoint should be added. The other bits are used incase of functionality expansion.the next char is understood as an unsigned number, that shows the size of the text, the next three chars are understood as the HEX values for the textcolor. The next chars are understood as a C-style string that declares the font used. After that the last char is understood as an ID number, for use in creating the css file. It is expected that the structure element is to be put in a list or other ordered data structure and stored for future parsing.

3.2 HTMLParser

This header contains the HTML structure, which houses the name for the html file, the string, containing it's head information(Title), and an ordered Element array, which houses the Elements which are to be added in sequence. This header also includes a function to dump" the structure into a C string which is the code for the HTML file. A simmilar function exists for crearing the code for the creation of the cascading sheets document.

N.B.

Standard C libraries are used for IO and the creation of the files.