Balanced Parentheses Project

In this project, you will write a function that checks for balanced parentheses symbols. Your function will have two string arguments and returns a string. The arguments are:

parens: a string with parens[2*i] a left parenthesis symbol and parens[2*i+1] the corresponding matching right parenthesis symbol.

Example: **`"()[]{}"`**

expr: a string containing parenthesis symbols

Example: "{x = (a+B[i])*(b+c)-A[i-1])}"

If your function finds an unmatched symbol, it will return the string of unmatched parenthesis symbols in same order that they appear in expr up to the first unmatched symbol.

For example, our previous example would return an empty string. However, below we show example expr strings with unmatched parenthesis symbols and the return value for those strings

expr	return string
`"{x = (a+B[i)*(b+c)-A[i-1)}"`	**`"([)"`**.
"**`{x = {(a+B[i])*(b+c)-A[i-1)]"`**	**`"{[)"`**

You should use a stack of characters that is used to hold left parenthesis symbols.

Your function should do no output; it should just return a string. The prototype is

```
string checkBalancedParens(string parens, string expr);
```

You will be provided with files **`balanced.h`** and **`balanced.cpp`**. The **`.h`** file will just contain the above prototype. The **`.cpp`** is where you will provide the code for the above function. This is the only file you will submit is **`balanced.cpp.`**