Playing with Strings and Vectors

Implement the following four functions.

```
std::string strip(const std::string &str);
std::string join(const std::string &sep, const std::vector<std::string> &strings);
std::vector<std::string> split(const std::string &str, const std::string &sep);
std::string swapcase(std::string str);
```

- `strip(str)`: Returns the string obtained from `str` by removing all the leading and trailing whitespaces. A character `c` is a whitespace if `std::isspace(c)` is true. For example,

  ```
  assert(strip("   wefafwefw    \n") == "wefafwefw");
  ```

  The assertion above should succeed.

- `join(sep, strings)`: Concatenate the strings `strings`, during which the string `sep` is inserted in between each given string. Then return the result. For example:

  ```
  std::vector<std::string> strings = {"hello", "world", "cxx23"};
  assert(join(", ", strings) == "hello, world, cxx23");
  ```

  The assertion above should succeed.

  If the vector `strings` is empty, return an empty string.

- `split(str, sep)`: Using `sep` as the delimiter, split the string `str` into a vector of strings and return that vector. For example,

  ```
  std::vector<std::string> ans{"", "aaa", "", "bbb", "cdefg"};
  assert(split("xaaaxxbbbxcdefg", "x") == ans);
  ans = {"", "x"};
  assert(split("xxx", "xx") == ans);
  ```

  The assertions above should succeed. It is guaranteed that `sep` is not an empty string.

- `swapcase(str)`: Returns the string obtained from `str` with cases swapped, i.e. lowercase letters are changed to their uppercase forms, and uppercase letters are changed to their lowercase forms. Other characters remain unchanged. For example, `swapcase("123..abcDEF")` is equal to `"123..ABCdef"`.

Submission

Submit your code containing these four functions to OJ.