Coding Challenge Searching Algorithm.

Analysis of the product data

- Product name field consist by manufacturer, model, and family.
- Model field is unique at same manufacturer group.
- Model field can search multiple times caused by similarity. For example "Nikon_Coolpix_**P7000**" and "Nikon_Coolpix_**700**", "Olympus_**FE-5010**" and "Olympus_Stylus_**5010**"

Analysis of the listing data

- **Listing data has manufacturer field. (It means program can make small target for searching)**
- Some title has or has not the information of family.

Current searching part are shown below.

```
std::string model = " " + makeStringLower(keyModel) + " ";
if(title.find(model) != std::string::npos)
{
  bRet = true;
}
```

And the following was my first version of searching (including the family field into search).

```
std::string model = " " + makeStringLower(keyModel) + " ";
if(!keyFamily.empty())
{
  if(title.find(model) != std::string::npos &&
     title.find(keyFamily) != std::string::npos)
  {
    bRet = true;
  }
}
else if(title.find(model) != std::string::npos)
{
  bRet = true;
}
```

Finally, model field is unique at same manufacturer group, and the checking family would be useless for the result of searching. I removed the part of searching with family.

Additionally, title field of listing has variant format (i.e. some parts are normal, some parts are capital, the others are lower capitals). Therefore, I should make all comparing text to lower capital before running search (*makeStringLower* function).