

Include <fstream>

```
#include <fstream>
```

Table 1: ofstream基本语句

语句	含义
<code>ofstream outfile("文件名");</code>	输出一个文件，名字为“文件名”。
<code>outfile << ("内容");</code>	将“内容”写入“文件名”文件。
<code>outfile.close();</code> <code>outfile.clear();</code>	关闭输出流。

Table 2: ifstream基本语句

语句	含义
<code>ifstream infile("文件名");</code>	读取“文件名”文件，此时“文件名”与infile已形成绑定关系。
<code>string s;</code> <code>while (infile >> s, !infile.eof);</code> <code>infile.close();</code> <code>infile.clear();</code>	将“文件名”的“内容”写入字符串变量s中。
<code>string s;</code> <code>ifstream infile(s.c_str());</code> <code>infile.close();</code> <code>infile.clear();</code>	同上，不同的写法
<code>ifstream infile;</code> <code>string s;</code> <code>infile.open(s.c_str());</code> //若是指针，用 <code>it->c_str();</code>	同上，不同写法。目的是不在开端绑定infile和文件。

往向量里塞文件

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>

int main()
{
    vector<string> files;
    files.push_back("文件1");
    files.push_back("文件2");
    files.push_back("文件3");
    vector<string>::const_iterator itr = files.begin();
    while (itr != files.end())
```

```
{
    ifstream infile(itr->c_str); //C语言的传参风格，历史遗留问题
    if (!infile)
    {
        cerr << "Error" << *itr << endl;
        infile.clear();
        ++itr;
        continue;
    }
    string s;
    while (infile >> s)
        process s; //在头文件定义的void()类型函数，等同于cout
    infile.close();
    infile.clear();
}
return 0;
}
```