プログラミング C++　第 6 回クイズ

学生番号：s1290034　　　　　　　　　　　氏名：関口由登

以下の string 型の文字列 xml と構造体 Tag がある．このとき「＜」から「＞」までの間の部分（タグ）を構造体の tag に保存し（例えば noun），その後の「＜／」までの部分（例えば I）を構造体の element に分解して，全体を変数 words に保存するプログラムを書け.

```
struct Tag {
  string tag;
  string element;
}
std::string xml = “<noun>I</noun> <verb>talk</verb> <adverb>about</adverb>
<noun>running</noun>”;
Std::list<Tag> words;
```

We have the above struct Tag, string object of named xml, and lists. Write a c++ source code for
(1) Extract a string between “<” and “>” and store it in tag in the struct.
(2) Extract a string between “>” and “<” and store it in element in the struct..
(3) Put the struct to lists.

解答欄 / Answer
ソースコード

```
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;

struct Tag {
  string tag;
  string element;
};

int main() {
    string s;
    vector<Tag> words;
    words.clear();
    while(cin >> s) {
        int tab_start = 0, tab_end = 0, element_start = 0, element_end = 0;
        Tag word;
        for(int i = 0; i < s.size(); i++) {
            if(s.at(i) == '<' && tab_start == 0) {
                tab_start = i + 1;
                continue;
            }
            if(s.at(i) == '>' && tab_end == 0 && element_start == 0) {
                tab_end = i - 1;
                element_start = i + 1;
                continue;
            }
            if(s.at(i) == '<' && s.at(i + 1) == '/' && element_end == 0) {
                element_end = i - s.substr(tab_start, tab_end).size() - 2;
                continue;
            }
        }
        if(tab_start != 0 && tab_end != 0 && element_start != 0 && element_end != 0) {
            word.tag = s.substr(tab_start, tab_end);
            word.element = s.substr(element_start, element_end);
        }
        words.push_back(word);
    }

    for(int i = 0; i < words.size(); i++) {
        cout << "Tag：" << words.at(i).tag << "　" << "Element：" << words.at(i).element << endl;
    }
}
```

実行結果

```
Tag：noun　Element：I
Tag：verb　Element：talk
Tag：adverb　Element：about
Tag：noun　Element：running
```