プログラミング C++　第 6 回クイズ

学生番号：s1270151　　　　氏名：上田健登

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

```
#include <iostream>
#include <string>
#include <list>
#include <algorithm>

struct Tag {
   std::string tag;
   std::string element;
};

int main(){
   std::string xml = "<noun>I</noun> <verb>talk</verb> <adverb>about</adverb>
<noun>running</noun>";

   std::string open_tag_begin = "<";
```

```
    std::string open_tag_end = ">";
    std::string close_tag_begin = "</";
    std::string close_tag_end = ">";

    std::list<Tag> words;

    std::string::iterator head = xml.begin();
    const std::string::iterator end = xml.end();
    std::string::iterator i, j;
    while(head != end){
        Tag word;
        i = std::search(head, end, open_tag_begin.begin(), open_tag_begin.end());
        head = i + 1;

        j = std::search(head, end, open_tag_end.begin(), open_tag_end.end());
        head = j + 1;

        std::copy(i + 1, j, std::back_inserter(word.tag));

        i = std::search(head, end, close_tag_begin.begin(), close_tag_begin.end());
        head = i + 2;

        std::copy(j + 1, i, std::back_inserter(word.element));

        words.push_back(word);

        j = std::search(head, end, close_tag_end.begin(), close_tag_end.end());
        head = j + 1;
    }

    for (std::list<Tag>::iterator it = words.begin(); it != words.end(); it++) {
        std::cout << it -> tag << ": " << it -> element << '¥n';
    }
    return 0;
}
```