

TA によるユーティリティのすゝめ

先週の課題3はこんな関数を作って解くことができます(※話を簡単にする為にエラーチェックはしてません)。

```
std::vector<std::string> comma_split(std::string str)
{
    std::vector<std::string> result;
    int pos = str.find(",");
    std::string s1 = str.substr(0, pos);
    std::string s2 = str.substr(pos + 1, str.size() - pos - 1);

    result.push_back(s1);
    result.push_back(s2);

    return result; // 実は C の配列と違って vector は値として return できる
}
```

この関数をカンマだけではなく、ユーザが指定した区切り文字列で区切れるようにすると便利そうなので変更してみましょう

```
std::vector<std::string> any_split(std::string str, std::string sep)
{
    std::vector<std::string> result;
    int pos = str.find(sep);
    /* 以下同文 */
    return result;
}
```

更に、文字列を何個にでも区切れるようにしてみましょう。

(例えば "a,b,c" を "," で区切ると "a" と "b" と "c" に区切れるようにします)

```
std::vector<std::string> split(std::string str, std::string sep)
{
    std::vector<std::string> dst;
    int head = 0;
    int tail = str.find(sep);
    int sp_len = sep.length();

    while ( tail != std::string::npos ) {
        dst.push_back(str.substr(head, tail - head) );
        head = tail + sp_len;
        tail = str.find( sep, head );
    }

    if ( head <= str.length() )
        dst.push_back( str.substr(head) );

    return dst;
}
```

これにより、「任意の文字列を任意の区切り文字列で区切ったものを返す関数」ができました。“任意の”という抽象的な言葉がよく出てきますね。これだけ抽象的であれば課題以外にも使える気がしませんか？

実際、この関数は CSV ファイルから読み込んだデータを要素ごとに取り出す場合や、姓と名を分けるとき等によく登場しますし、Java や Ruby などでは標準で組み込まれているほど有名な関数です。

このように、「こうしたらもっと便利になるんじゃないか？」と少し想像力を働かせると、ラクができる素敵な便利ツール作ることができ(る場合があります)ます。そして、この便利ツールをユーティリティといいます。

ユーティリティ作って(あるいは見つけて)しまえば、これ以降で同じような処理をペチペチ書くことも、書くついでにバグを埋め込んでしまうことも(ほぼ)なくなります。つまり、時間的、精神的にラクができます。**怠け者にとってこれほど素晴らしいことはありません。**

皆さんもラクを求めてみませんか？

※この改良版が <http://ideone.com/ULEhj> に up してあります