

第12回演習

演習 1:string::find_first_of() ex12-1.cpp

p.10 や 9 回目 p.20 のプログラムを参考に、対象となる文字列 `src` から文字列 `sub` 中のどれかの文字を探し、見つけた文字と `src` 中の添字の値を出力する関数を作成せよ。

```
#include <iostream>
using std::cout, std::string;
void find_all(string src, string sub) { /* here */ }
int main() {
    string a {"acbaabebcdacbac"};
    find_all(a, "ac");
    find_all(a, "b");
    find_all(a, "cb");
}
```

```
$ ./a.out
a:0 c:1 a:3 a:4 c:8 a:10 c:11 a:13 c:14
b:2 b:5 b:7 b:12
c:1 b:2 b:5 b:7 c:8 c:11 b:12 c:14
```

演習 2:正規表現 ex12-2.cpp

以下のプログラムをコンパイルし、例のように実行せよ。

```
int main(int argc, char *argv[]) {
    if (argc < 2) {
        cout <<"usage: " << argv[0] <<" regex\n";
        return 1;
    }
    string rs { argv[1] };
    std::regex r { rs };
    std::vector<string> vs { "ac", "abc", "acc",
        "a c", "a0c", "a1c", "abd", "a3d", "axc", "a\nc" };
    for (auto e:vs)
        if (std::regex_match(e, r))
            cout <<"\" << e <<"\" ";
    cout <<"\n";
}
```

```
$ ./a.out 'a[bc]c'
"abc" "acc"
```

演習 3:正規表現 (プログラム作成なし)

前の演習のプログラムを使って、以下の結果が出力されるようにコマンド引数を考えて実行せよ。

1. a で始まり, 2 番目に数字、c で終わる
 - ▶ "a0c" "a1c"
2. a で始まり, 2 番目に改行以外の文字、c で終わる
 - ▶ "abc" "acc" "a c" "a0c" "a1c" "axc"
3. c で終わる
 - ▶ "ac" "abc" "acc" "a c" "a0c" "a1c" "axc"
4. a で始まり c で終わる。間に b が入っても良い。
 - ▶ "ac" "abc"
5. a で始まり, 2 番目にホワйтスペース以外の文字, c で終わる
 - ▶ "abc" "acc" "a0c" "a1c" "axc"

演習 4:正規表現 ex12-4.cpp

資料 p.11 のプログラムの正規表現をコマンド引数で与えられるように変更せよ。

```
$ ./a.out 'a.*c'
ac, aaac, ac ccc,
ac, aaac,  aac , ac ccc, anc ,
[okam@csz840 ex4]$ ./a.out 'a[a-z]c'

aaac,  aac , anc ,
[okam@csz840 ex4]$ ./a.out 'a[a-z]*c'
ac, aaac,
ac, aaac,  aac , ac ccc, anc ,
```

演習 5:正規表現 ex12-5.cpp

資料 p.26 の data の値を "acaaac aac akbac cccancnkp" とし、正規表現をコマンド引数から受け付けるように変更する。下記の実行例は正規表現が一部抜けているので、これを補って結果が同じになるように実行せよ。

```
$ ./a.out 'ak(      *)c'
data: acaaac aac akbac cccancnkp
m.prefix():acaaac aac
m.suffix():ancnkp
m[0]:akbac ccc, m.position(0):11
m[1]:bac cc, m.position(1):13
$ ./a.out '  3}( w)'
data: acaaac aac akbac cccancnkp
m.prefix():ac
m.suffix(): aac akbac cccancnkp
m[0]:aaac, m.position(0):2
m[1]:c, m.position(1):5
```

演習 6:正規表現 ex12-6.cpp

資料 p.32 の smatch 用イテレータを使って、前の演習と同じ機能のプログラムを作成せよ。

```
$ ./a.out '\b(aac).*k'
data: acaaac aac akbac cccancnkp
m.prefix():acaaac
m.suffix():p
m[0]:aac akbac cccancnk, m.position(0):7
m[1]:aac, m.position(1):7
$ ./a.out 'a.*(a.c).*(\1).*c'
data: acaaac aac akbac cccancnkp
m.prefix():
m.suffix():nkp
m[0]:acaaac aac akbac cccanc, m.position(0):0
m[1]:aac, m.position(1):3
m[2]:aac, m.position(2):7
```