

第5回演習

演習 1: スタック

- ▶ 以下のプログラムを実行し入力逆順に出力されることを確認せよ。
- ▶ 実行例：echo 1 2 3 4 5 | ./a.out

```
// ex05-1.cpp
#include <iostream>
#include <stack>
int main()
{
    std::stack<int> s;
    for (int x; std::cin >> x; )
        s.push(x);

    while (!s.empty()) {
        std::cout << s.top() << " ";
        s.pop();
    }
    std::cout << "\n";
}
```

演習 2: キュー

- ▶ 講義資料 p.27 の `simulate` 関数を引数なしで `n` 回呼び出し、結果の平均を求めるプログラムを作成せよ。
- ▶ `n` はコマンド引数で与えることとする。
- ▶ ファイル名 `ex05-2.cpp`
- ▶ 実行例（毎回変わるが 4 に近い値となる）：

```
$ ./a.out 10
4.1
$ ./a.out 100
3.31
$ ./a.out 200
3.685
```

演習 3: スタック

- ▶ 括弧の対応をスタックで調べるプログラムを作れ
 - ▶ 次のスライドの `match()` 関数を作成する
 - ▶ `ch` に閉じ括弧が指定されるので、対応する開き括弧を決めて `top` と比較する
- ▶ 標準入力から文字が入ってくる
- ▶ かぎ括弧、丸括弧、波カッコの対応を調べる
- ▶ 入れ子形を含めてプログラムや数式の括弧のように対応付けがされているかを調べる
- ▶ `echo` コマンドで入力を与える場合は、`echo` 実行の前にシェルがの記号を処理しないように単一引用符 (`'`) で囲む

```
$ echo '[()]{ }{[( )]( )}' | ./a.out
ok
$ echo '[()])' | ./a.out
unmatch ]
```

プログラムの雛形

```
#include <iostream> // ex05-3.cpp
#include <stack>
bool match(std::stack<char>& stk, char ch);
int main() {
    std::stack<char> stk;
    bool flag{true};
    char ch;
    while (flag && std::cin >> std::noskipws >> ch) {
        switch (ch) {
            case '[': case '{': case '(':
                stk.push(ch);
                break;
            case ']': case '}': case ')':
                flag = match(stk, ch);
                break;
        }
    }
    if (flag) { std::cout << "ok\n"; }
    else { std::cout << "unmatch " << ch << "\n"; }
}
```

演習 4: スタックとキュー

キューは挿入した順に、スタックは逆順にデータを取り出せる。この特徴を使って回文を判定する関数を作成せよ。ただし、以下の main 関数から使用できること。

```
int main() { // ex05-4.cpp
    vector<vector<string>> v {
        {"た","け","や","ぶ","や","け","た"},
        {"し","ん","か","ん","せ","ん"},
        {"し","ん","ぶ","ん","し"} };
    for (const auto& s:v) {
        if (is_palindrome(s)) {
            for (auto& e:s) cout << e;
            cout << "\n";
        }
    }
}
```

```
$ ./a.out
たけやぶやけた
しんぶんし
```

演習 5: スタック

- ▶ 次の URL の Web ページを読み、逆ポーランド記法の数式を計算するプログラムを作成せよ。

<https://ja.wikipedia.org/wiki/逆ポーランド記法>

- ▶ 数式は四則演算 (+-*/) とし、標準入力から与えられる。
- ▶ echo コマンドで入力を与える場合は、echo 実行の前にシェルが '*' の記号を処理しないように単一引用符 (') で囲む
- ▶ ヒント (厳密には型が異なる):
 - ▶ 数字判定 `bool std::isdigit(char), #include<cctype>`
 - ▶ 数値変換 `int std::stoi(const string&)`

```
$ ./a.out
13 17 + 4 1 - *          <-- Enter の後に Ctrl-D
90
$ echo '3 4 + 1 2 - *' | ./a.out
-7
```

雛形

```
int main() { // ex05-5.cpp
    std::stack<int> stk;
    for (std::string token; std::cin >> token; ) {
        if (std::isdigit(token[0])) // <cctype>
            stk.push( std::stoi(token) );
        else {
            if (stk.size() < 2) { break; } // error
            int x ... // 操作 2 個
            int y ... // 操作 2 個
            if (token == "+") // 操作 1 個
            else if (token == "-") // 操作 1 個
            else if (token == "*") // 操作 1 個
            else if (token == "/" && x!=0) // 操作 1 個
            else { break; } // error
        }
    }
    if (stk.size() == 1) {
        std::cout << stk.top() << "\n"; stk.pop(); }
}
```