
プログラミング実習における コンピュータを使わない 教材とその効果

玉木徹

新潟大学
工学部情報工学科・
大学院自然科学研究科

本学科のプログラミング授業

合計30週

- 通年（二学期）
- 講義 45 時間：週 1.5 時間
- 実習 90 時間：週 3 時間
- C 言語

プログラミング基礎
(講義)

1 年次後期

プログラミング
基礎実習

プログラミング 1
(講義)

2 年次前期

プログラミング
実習 1

しかし：卒研配属後プログラムが組めない
本当に理解している？

学生のプログラミング能力の現状

学生約 100 名に対する小テスト (2003/6)

- 講義・実習を一年受講している
- 教科書・参考書の持ち込み可
- 15分で5問

▶ 問:
「右の数式を C 言語の代入式
に書き直せ」

$$x = \frac{a+b}{c - \frac{d}{a+2}}$$

▶ 正解:

$$x = (a+b) / (c - d / (a+2))$$

▶ 正当率: 74%

学生のプログラミング能力の現状

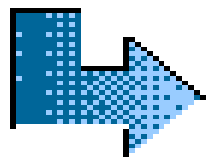
- ▶ 問 : 「配列 a[128] の全ての要素の和を求めるように下線部を埋めよ」

```
int i, sum=0;  
for(____;____;____)  
    sum += a[i];  
printf("%d\n", sum);
```

- ▶ 正解:

```
for(i=0;i<128;i++)
```

- ▶ 正当率 : 41%



- 基礎知識の欠如
- 基礎訓練の必要性

「基礎」とは何か？

従来の教材と実習

C 言語の最初の
サンプルプログラム

```
#include <stdio.h>
int main(int argc, char *argv[]){
    printf("Hello World!\n");
    return 0;
}
```

問題：

- プログラムを入力し、実行する必要がある
- 多数の記号と暗黙の規則がある
- 他の手続き型言語と共通する部分がない

サンプルプログラムは「基礎」ではない！

C 言語の実習課題

「〇〇を計算して表示するプログラムを作成する」

問題：

- コンパイルをする手順が複雑かつ環境依存
- 構文エラーが多発 + 多量のエラーメッセージ

実行は「基礎」ではない！

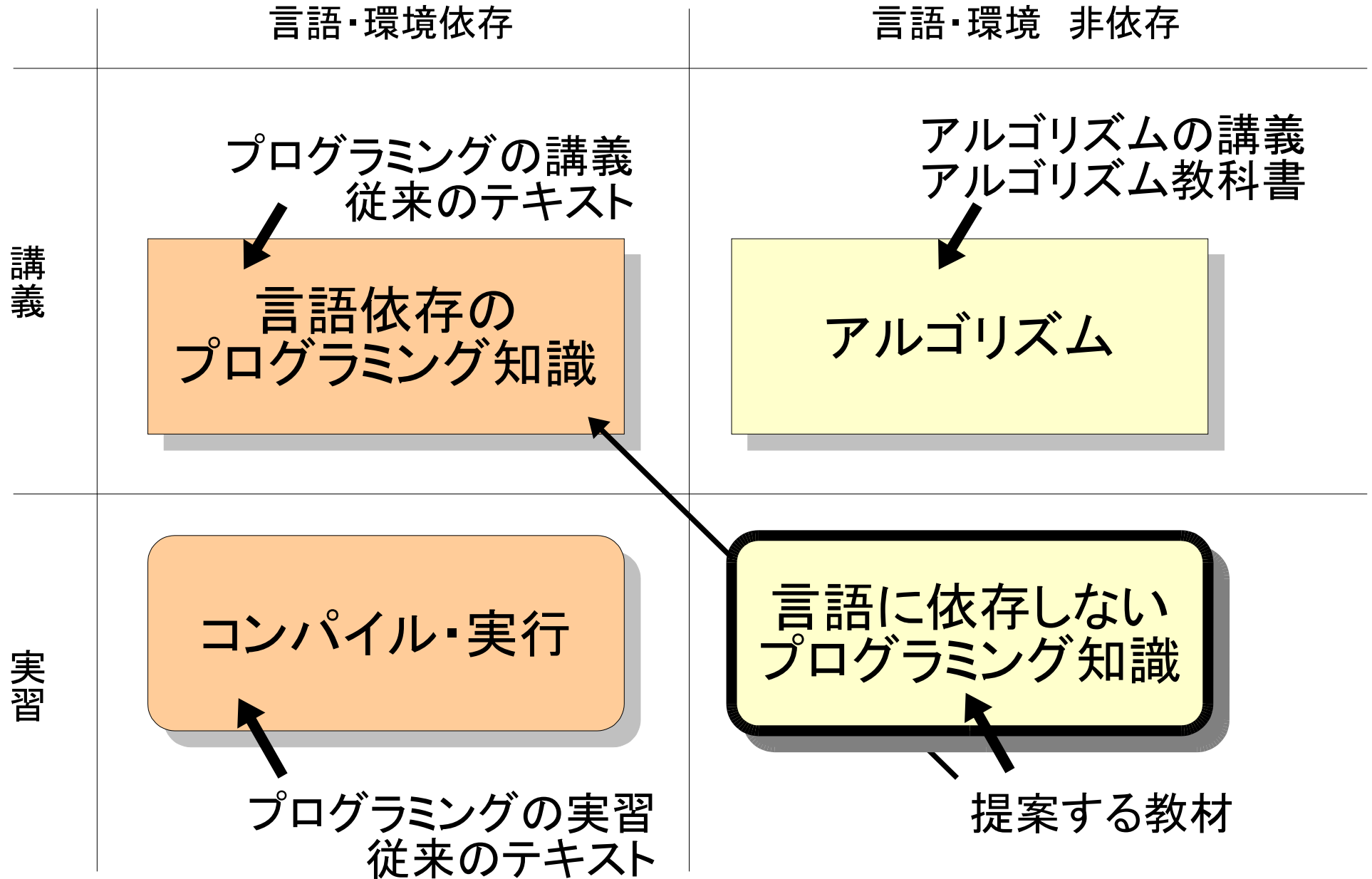
従来のテキスト・講義・実習

- テキスト(教科書)
 - 環境の導入(インストール等)
 - 簡単なサンプルプログラム(hello.c)
 - 各項目の詳細
- 講義
 - テキストに沿って授業
- 実習
 - サンプルプログラムの入力・実行
 - 課題プログラムの作成

B. W. カーニハン, D. M. リッチー, プログラミング言語 C, 第 2 版, 共立出版, 1989



プログラミングに必要な要素



提案する教材

プログラミング・ドリル

- ✓ **プログラミングの基礎から開始**
 - ・ 全ての手続き型言語に共通
 - ・ C言語に基づく
 - ・ Pascal/Fortran/C++ 等にも適用可
- ✓ **コンピュータを使用しない**
 - ・ 紙に解答を書く
 - ・ コンピュータではなく頭で計算結果を考える
- ✓ **プログラミングではなく数学の知識が必要**
 - ・ 基礎的な数学の関数 (ex: \sin / \cos / $\sqrt{\quad}$)
 - ・ 数学記号 (ex. e)
- ✓ **一つの項目について繰り返し解く**

プログラミング・ドリルの内容

問1:

「評価」とは計算することである。(例: $1+1$ の評価結果は 2)
次の C 言語の式を評価せよ。

式:

$1+2$

$22*3.3$

$2-1$

$10.2+5.1$

$10/5$

答:

3

42.6

1

15.3

2

プログラミング・ドリルの内容

問2:

「評価」とは(大小や同異等を)判定することである。判定結果は正しい(真)ならば1、間違っている(偽)ならば0で表す。(例: $1 > 0$ の評価結果は1(真)) 次のC言語の式を評価せよ。

式:

$0 < 1$

$1.0 \neq 10.0$

$-1 \geq 3$

$3 < -1.5$

答:

1

1

0

0

プログラミング・ドリルの内容

問3:

「評価」は左から行う。次の C 言語の式を評価せよ。

式 :

$-2+4-3.5$

$3*8/4$

$10/2/5$

$1.1 + 0.1 < 1.1$

$0 < 3 < 2$

答 :

1.5

6

1

0

1

プログラミングドリルの項目

- 式評価
- 変数への代入
- 変数の初期化
- 数学関数
- 整数型と実数型
- 変数宣言
- 型の表現範囲
- 文字型
- printf 関数
- 配列
- 宣言時の初期化
- while ループ
- 二重ループ
- 無限ループ
- if
- else
- 剰余
- 算術演算子
- for ループ
- フローチャート
- ポインタ

約 20 項目 250 問

プログラミングドリルを用いた プログラミング実習

12:50

先週の復習小テスト

ドリル

14:20

14:35

ドリル

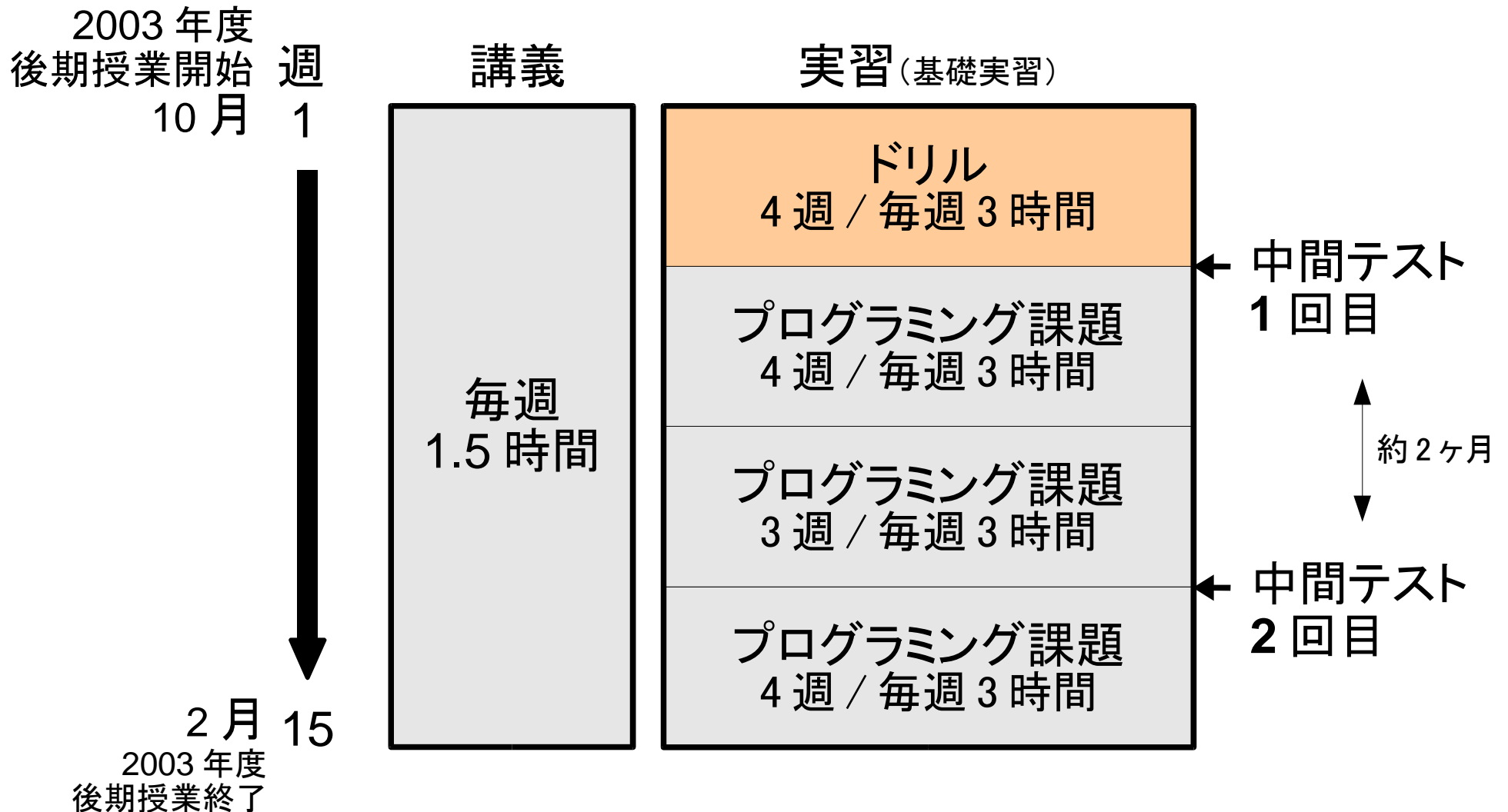
16:05

本日の復習小テスト

- 実習：週 3 時間（3,4 限）
- 毎週約 50 問
- 実習の最初と最後に小テスト



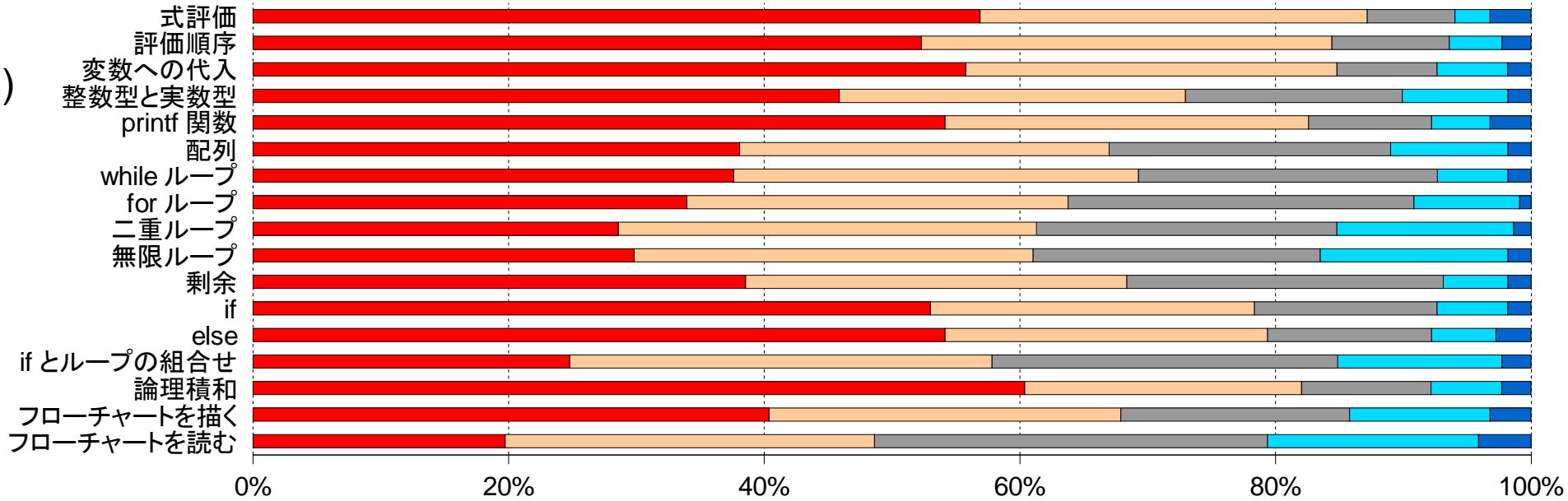
プログラミングドリルを用いた プログラミング実習(2003年度)



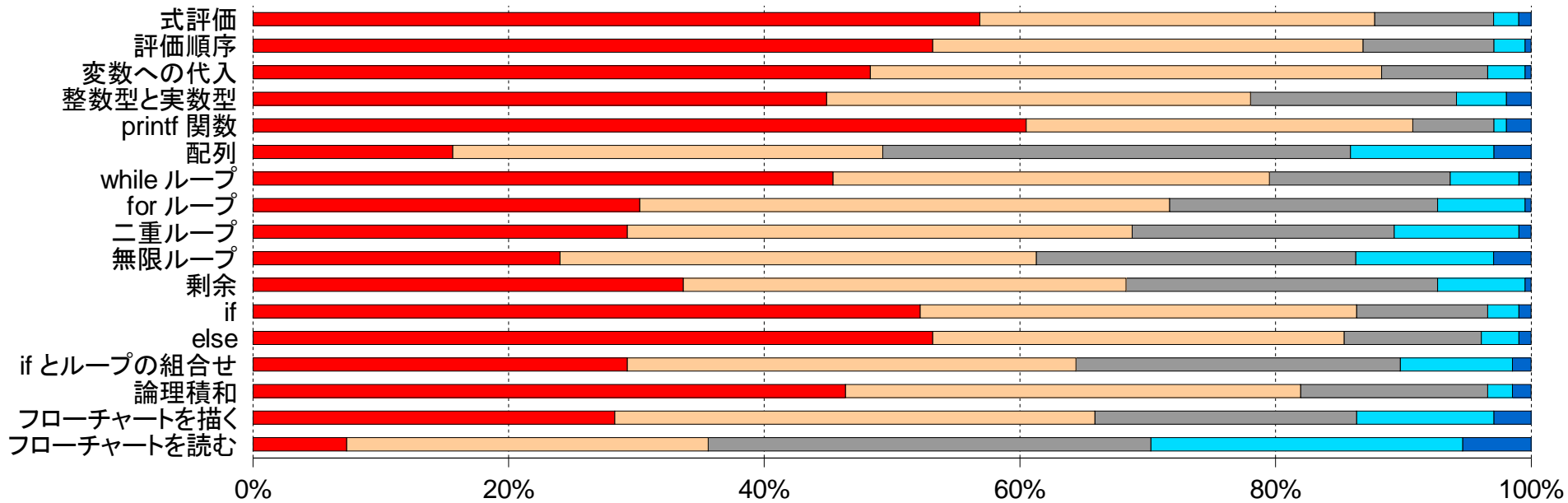
理解度アンケート



1 回目 (2003/11)



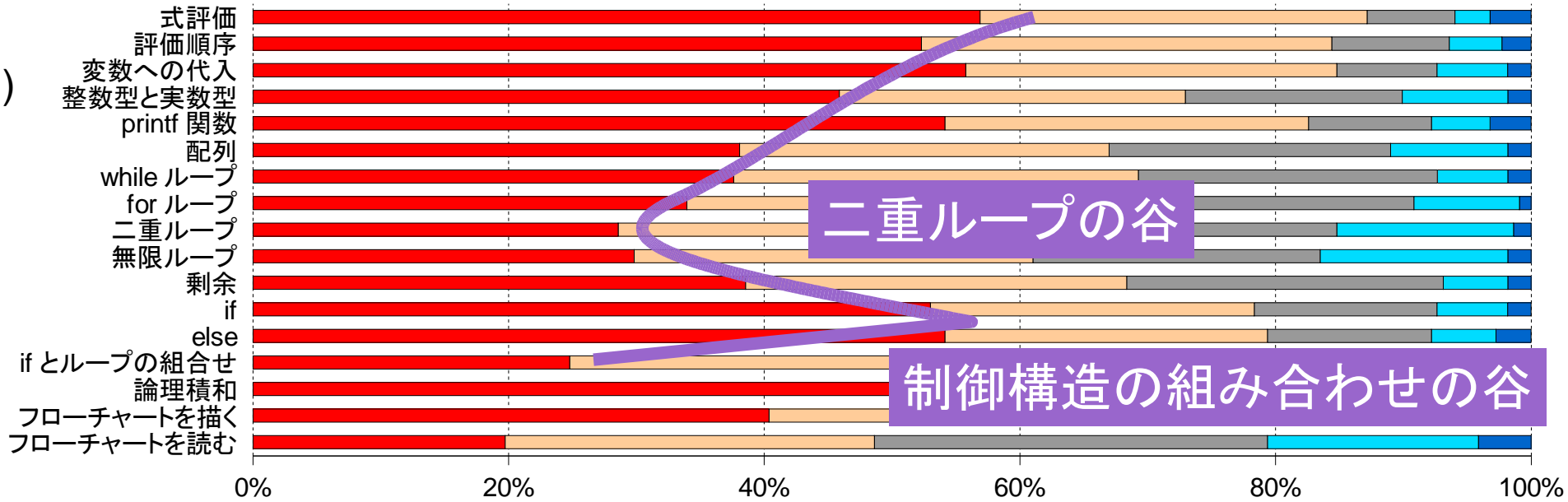
2 回目 (2004/1)



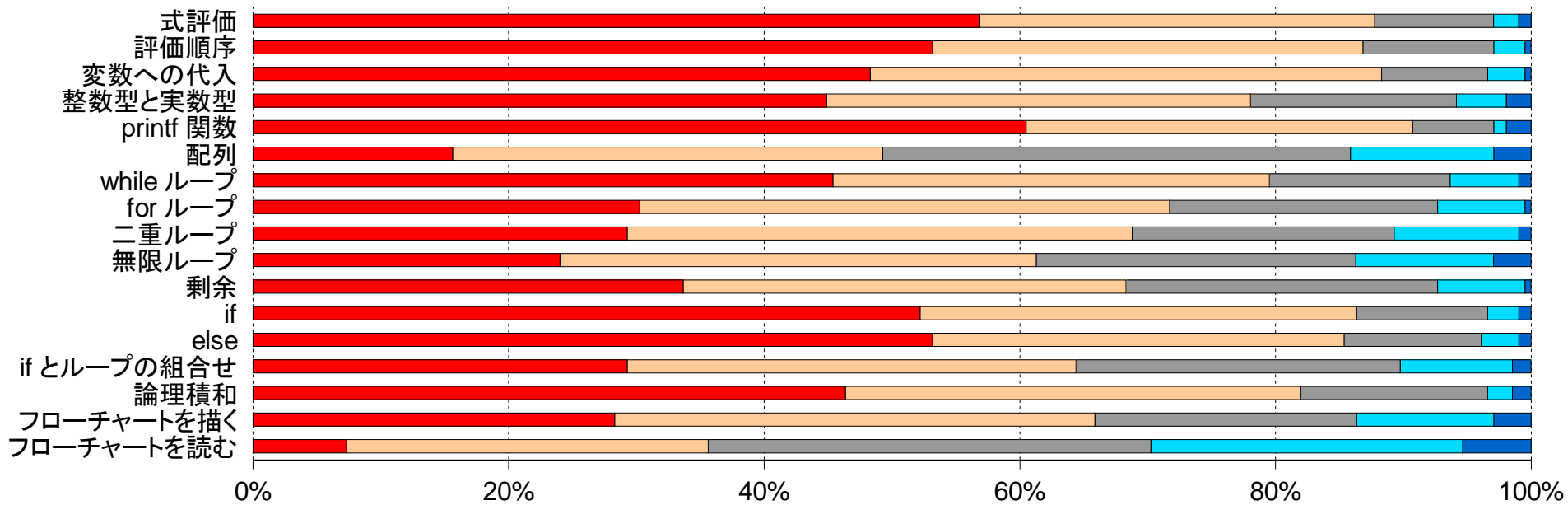
理解度アンケート



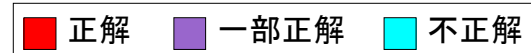
1 回目
(2003/11)



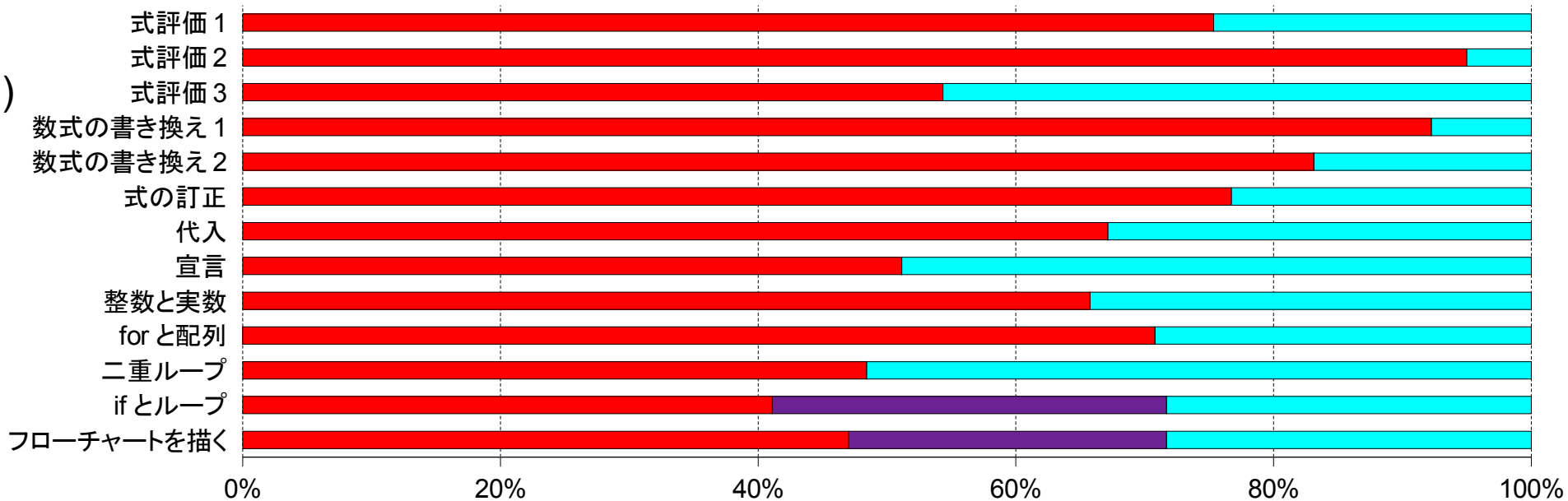
2 回目
(2004/1)



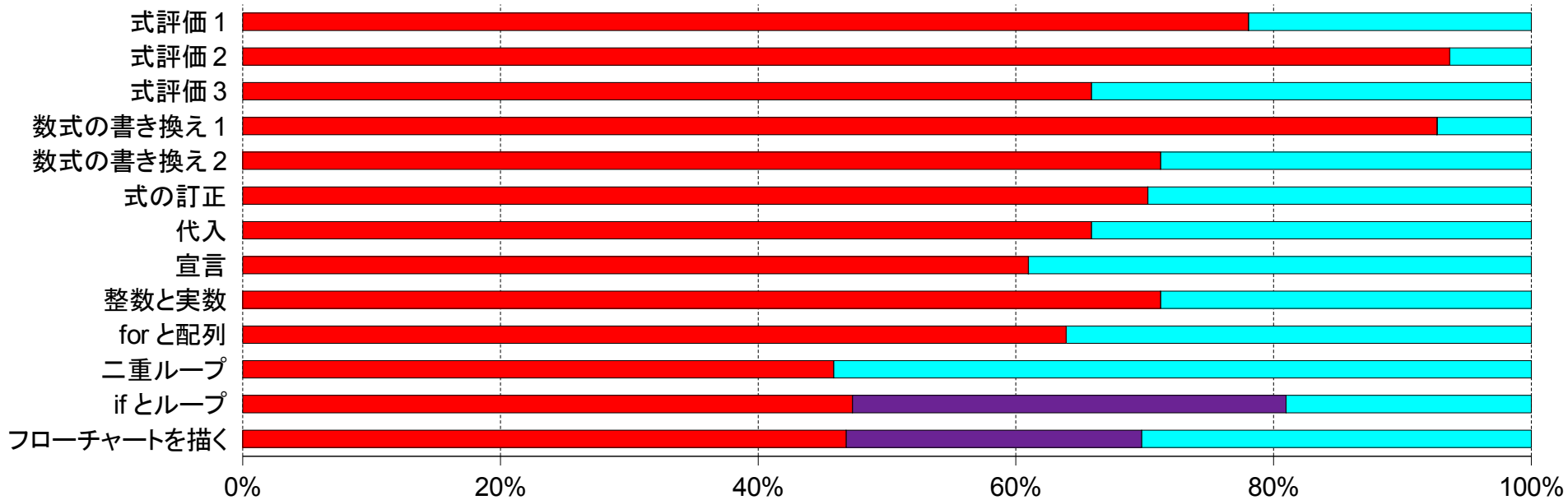
中間テスト結果



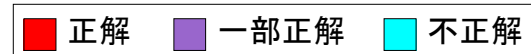
1 回目 (2003/11)



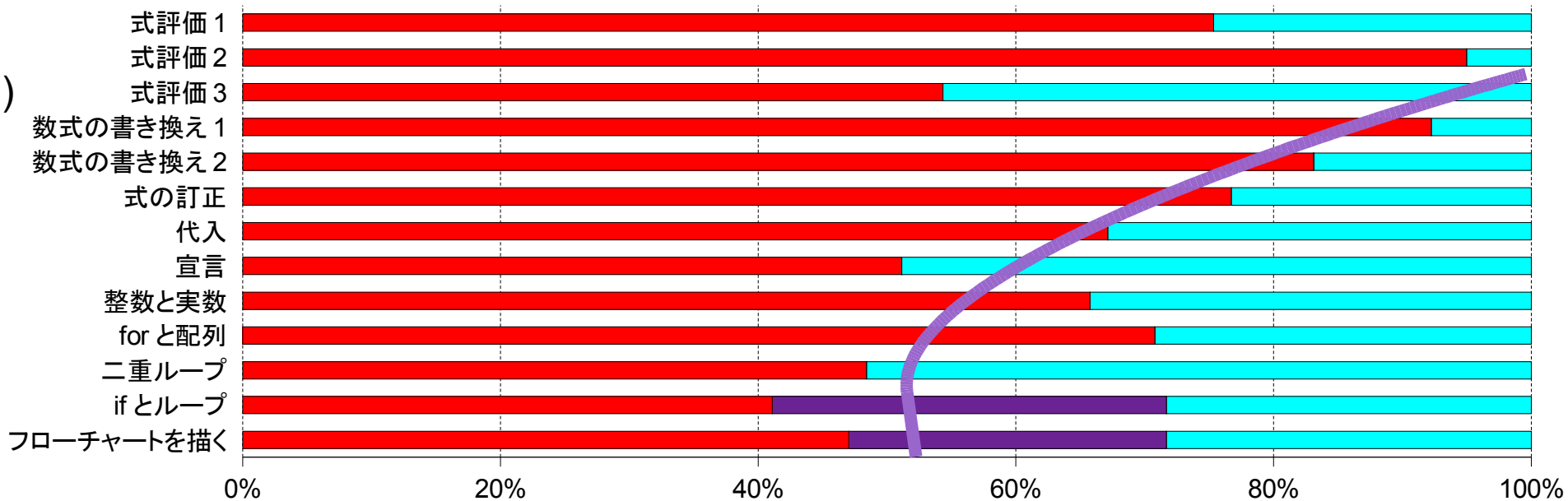
2 回目 (2004/1)



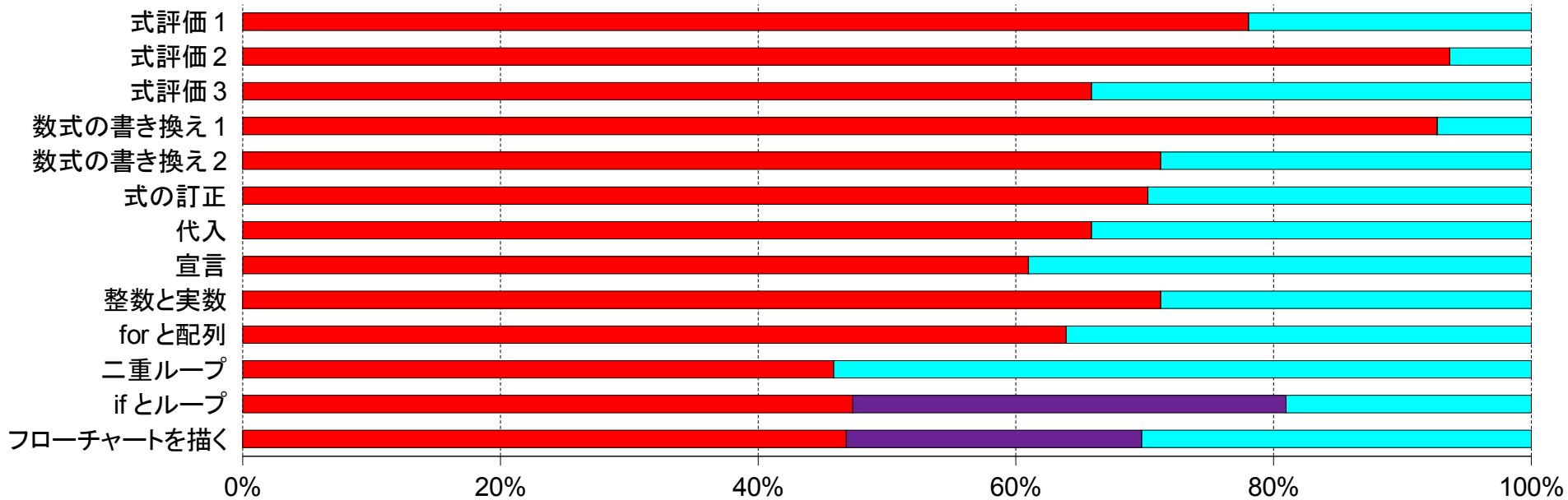
中間テスト結果



1回目 (2003/11)



2回目 (2004/1)



中間テストと成績の相関

講義、実習の成績と2回の中間テスト結果の相関

	講義	実習	1回目	2回目
講義	1.00	0.59	0.48	0.54
実習	0.59	1.00	0.60	0.66
1回目	0.48	0.60	1.00	0.69
2回目	0.54	0.66	0.69	1.00

問題：

- 相関は 0.5 ~ 0.6
- 二回のテストの間隔が2ヶ月と短い
- 長期の定着度をみるための追跡調査が必要

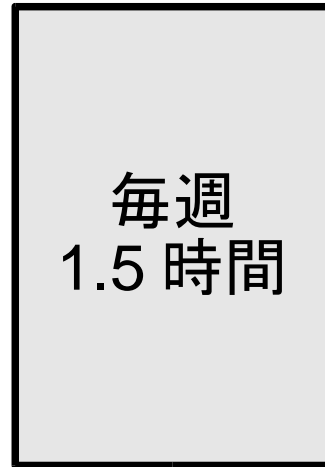
プログラミングドリルを用いた プログラミング実習(2004-5年度)

2004年度
後期授業開始
10月

週
1

講義

実習(基礎実習)



← 中間テスト
1回目
(2004/11)

2月
2005年度
後期授業終了

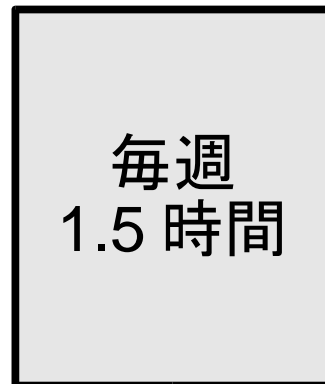
15

2005年度
前期授業開始
4月

1

講義

実習(実習1)



約半年

7月
2005年度
前期授業終了

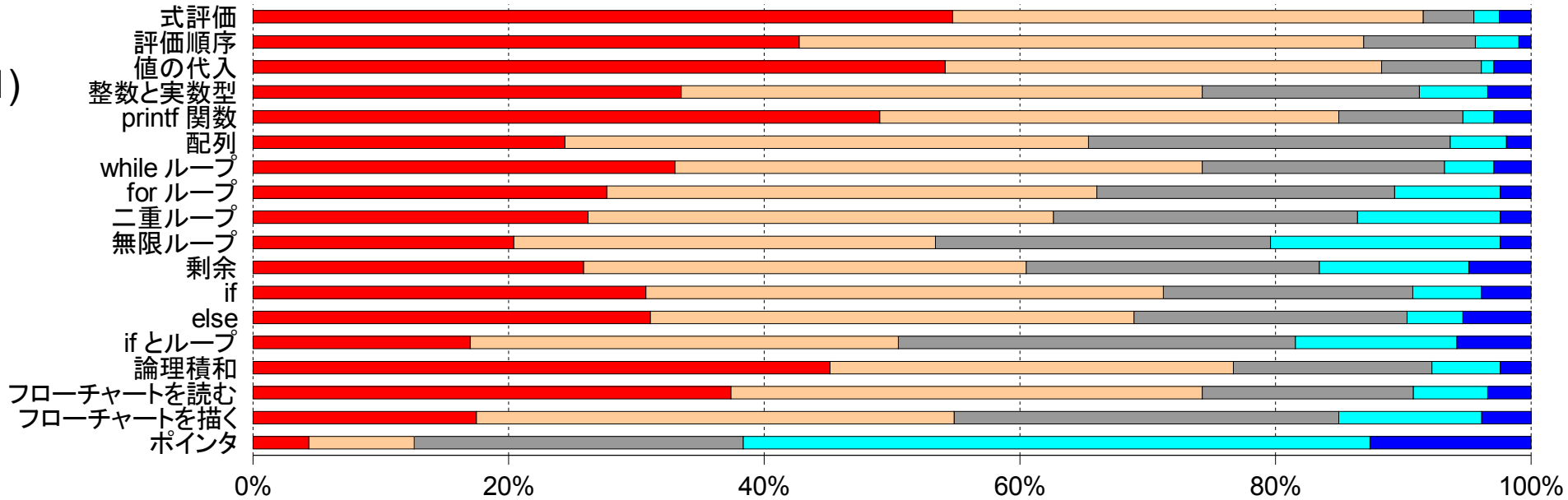
15

← テスト
2回目
(2005/6)

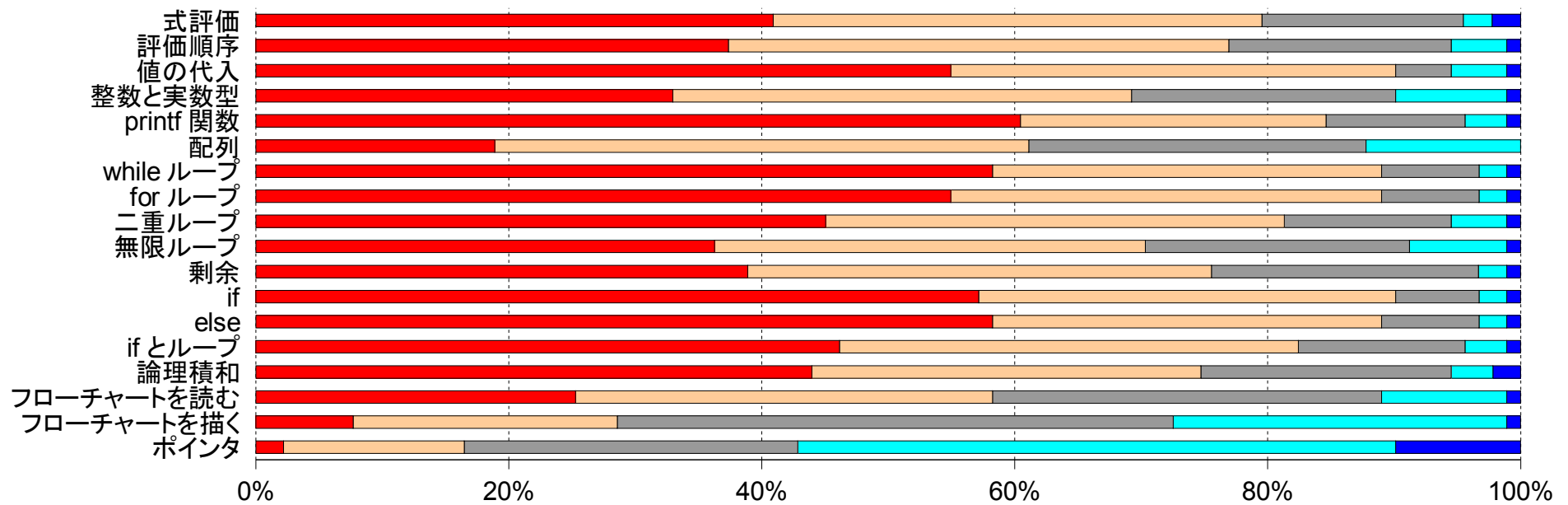
理解度アンケート



1 回目 (2004/11)



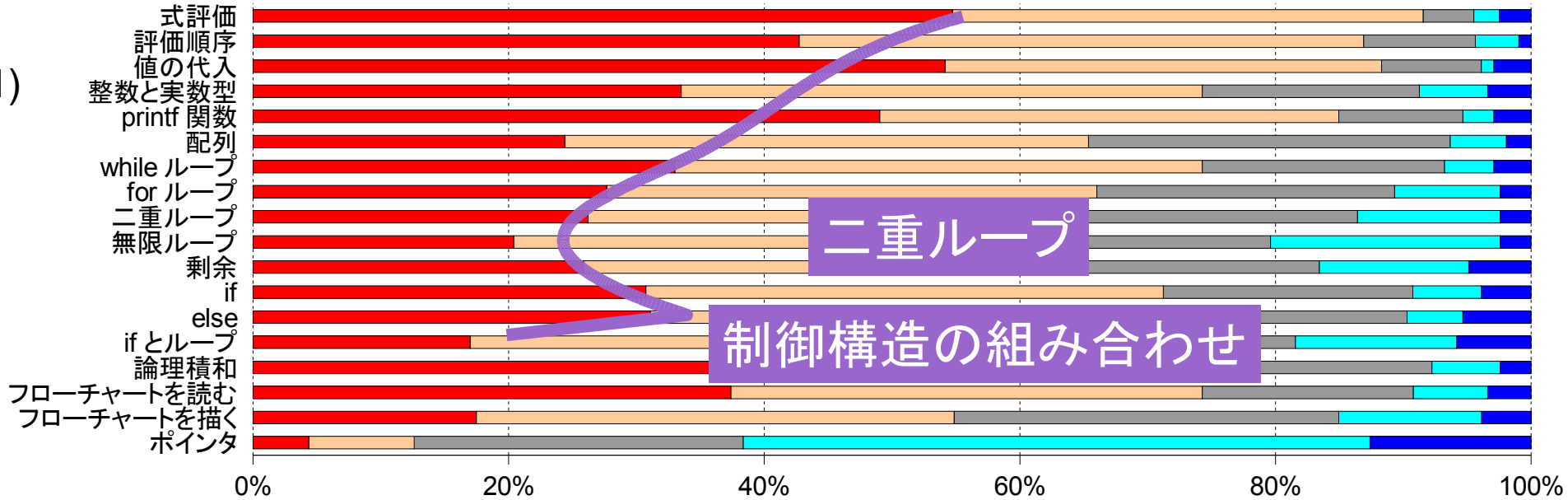
2 回目 (2005/6)



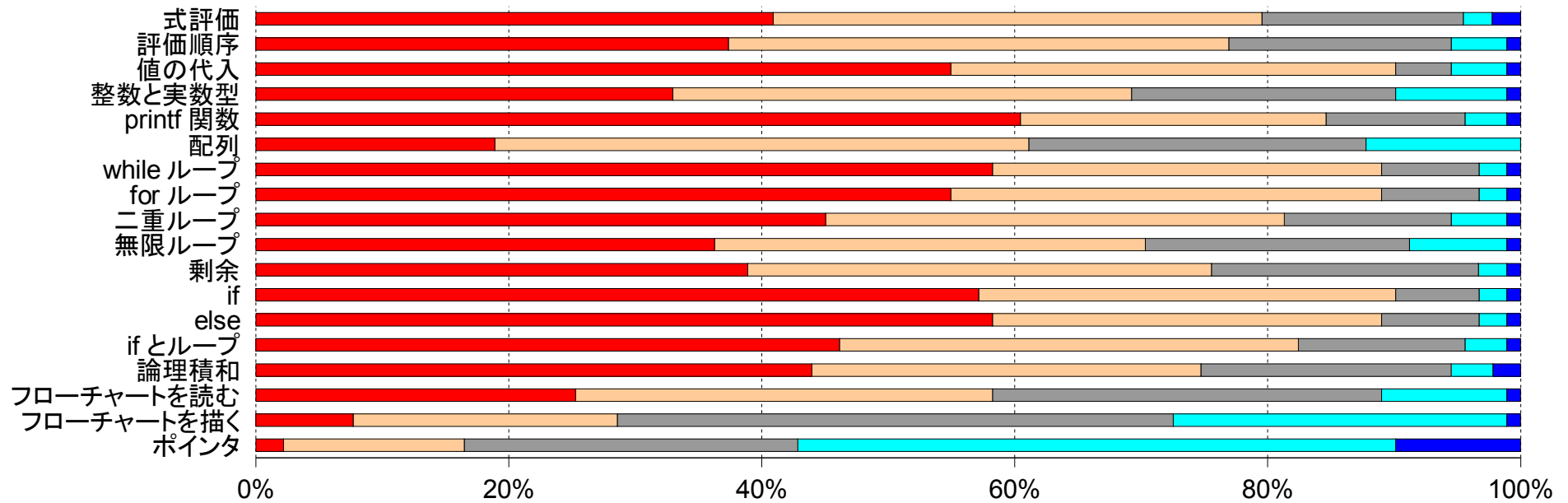
理解度アンケート



1 回目 (2004/11)



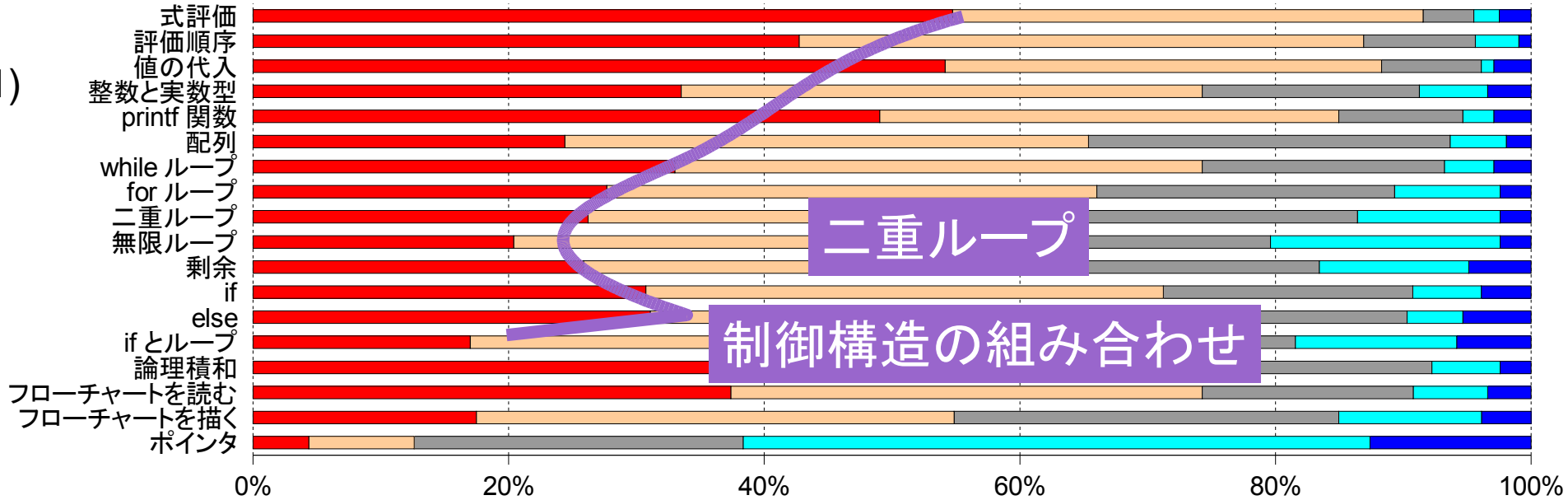
2 回目 (2005/6)



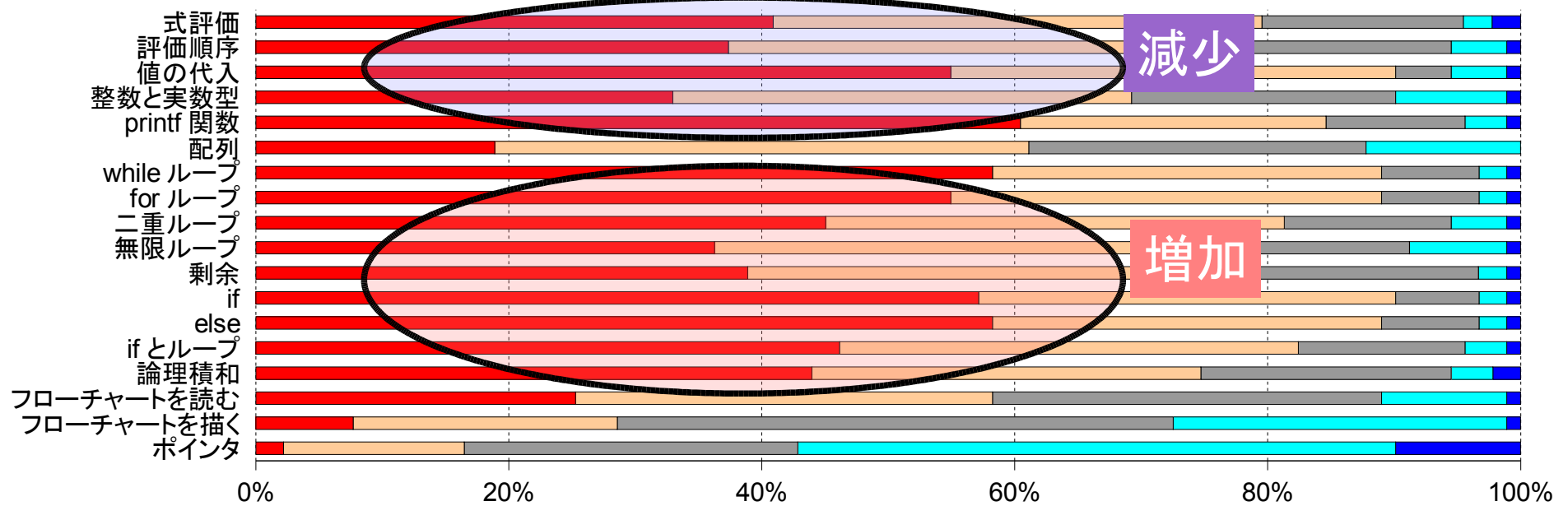
理解度アンケート



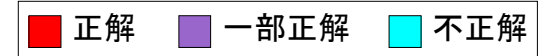
1 回目
(2004/11)



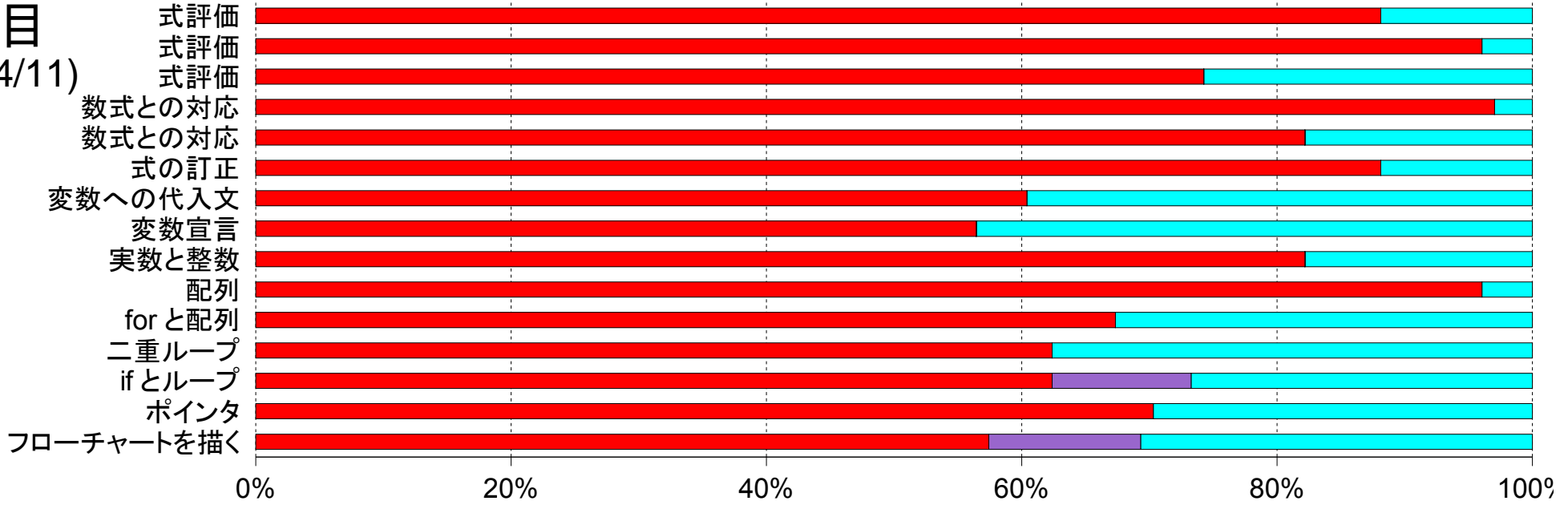
2 回目
(2005/6)



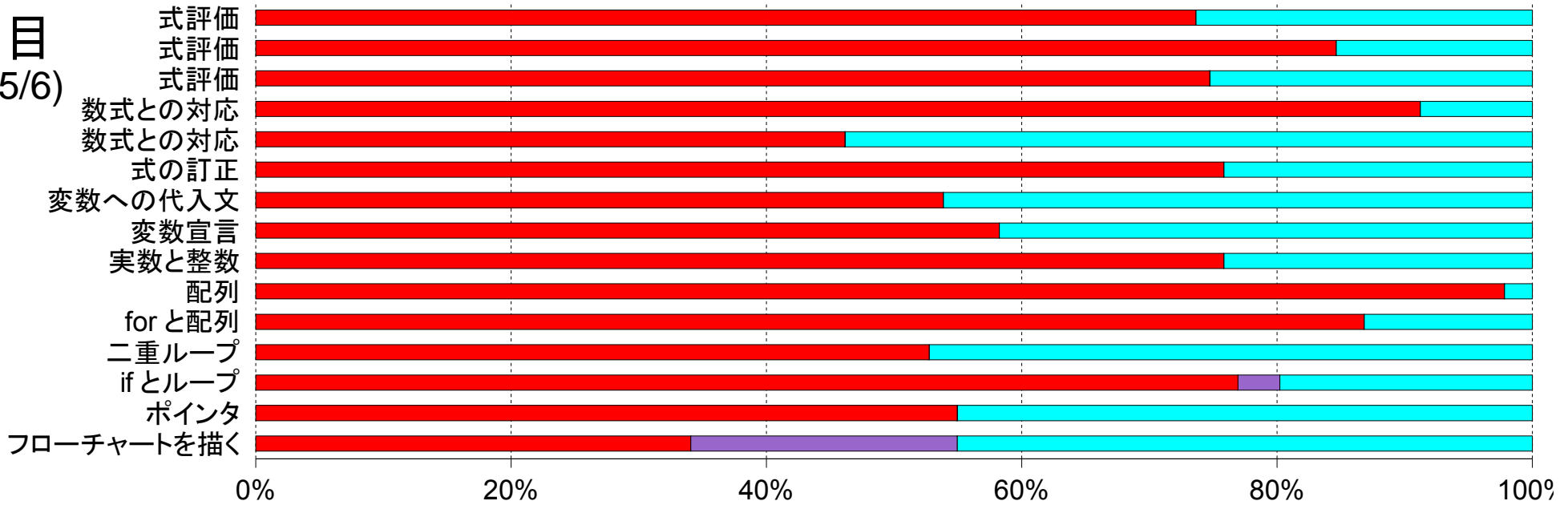
テスト結果



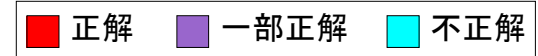
1回目 (2004/11)



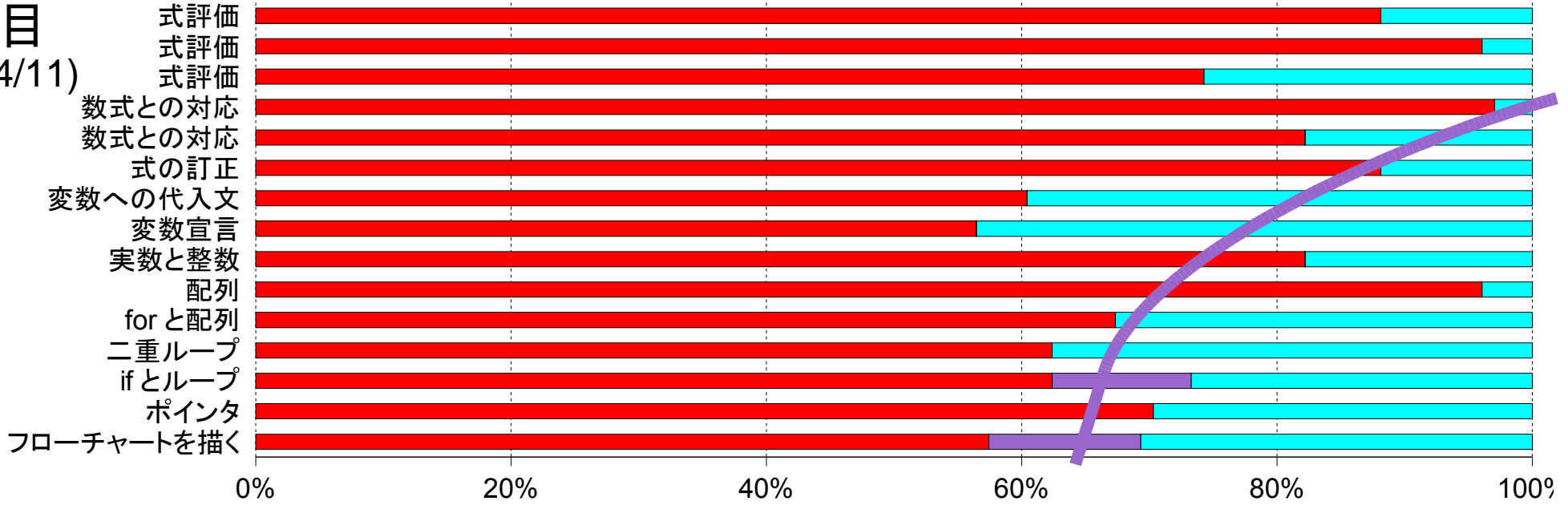
2回目 (2005/6)



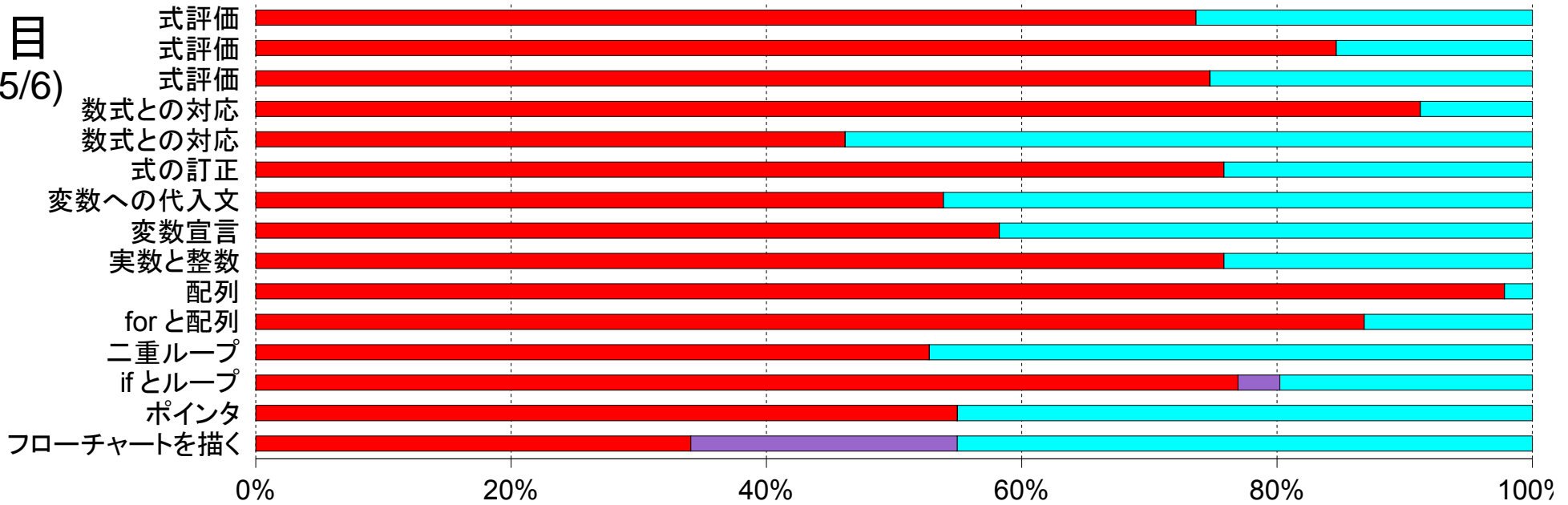
テスト結果



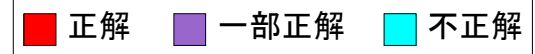
1回目 (2004/11)



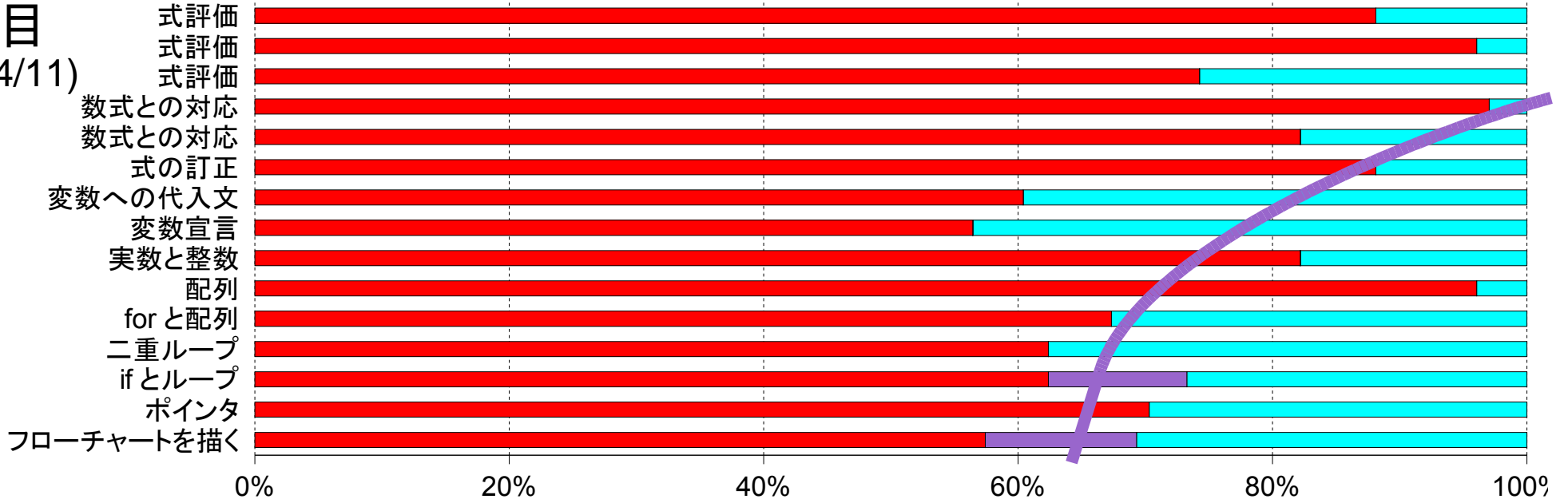
2回目 (2005/6)



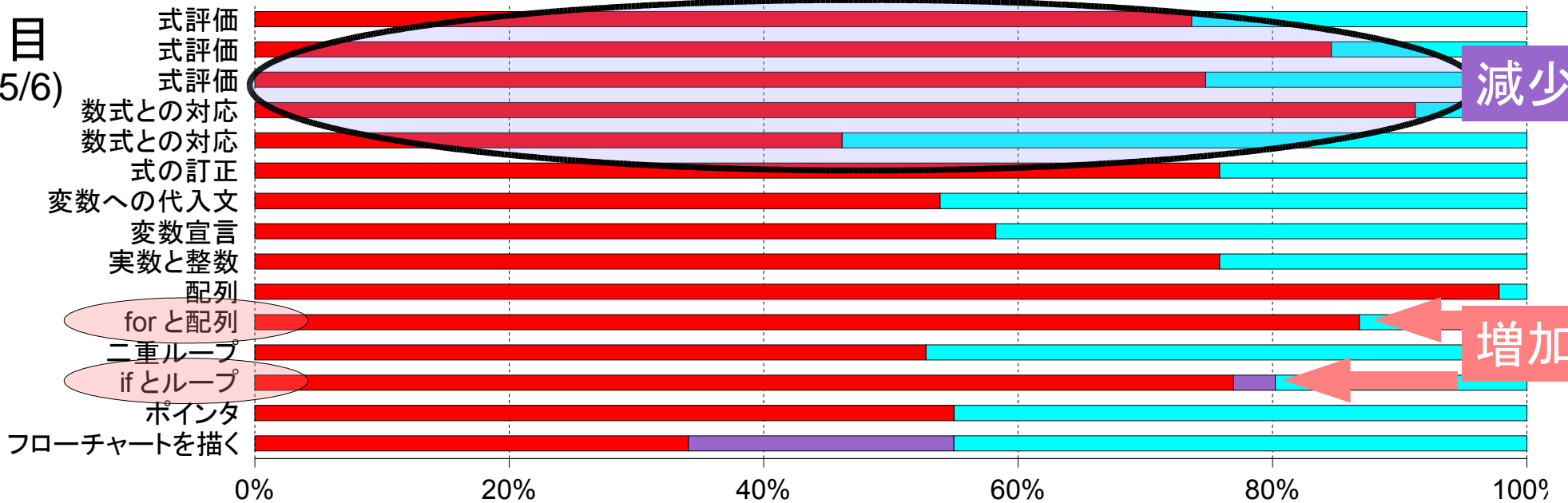
テスト結果



1回目
(2004/11)



2回目
(2005/6)



減少

増加

アンケートとテスト結果について

- 制御構造の組み合わせ
 - 理解度と成績が**上昇**
 - 実際のプログラミング課題を通して制御構造に対する理解が深まった結果
- 式評価・評価順序
 - 理解度と成績が**減少**
 - 実際にプログラム課題を行ってはいるが、基礎的な式評価に苦手意識が広がった
 - 処理の分岐条件や終了条件をしっかりと把握せず、漫然とプログラムを組み立てている？

考察：理解度アンケートについて



- **実際にプログラムを組む課題**
 - ループや分岐などの制御構造に慣れることに適している
 - 式評価など基礎的な理解は深まらない
- **提案する教材**
 - 基礎的な理解を身につける
 - (計算機を使わず自分の頭を使って)考える訓練を継続的に行うことに適している



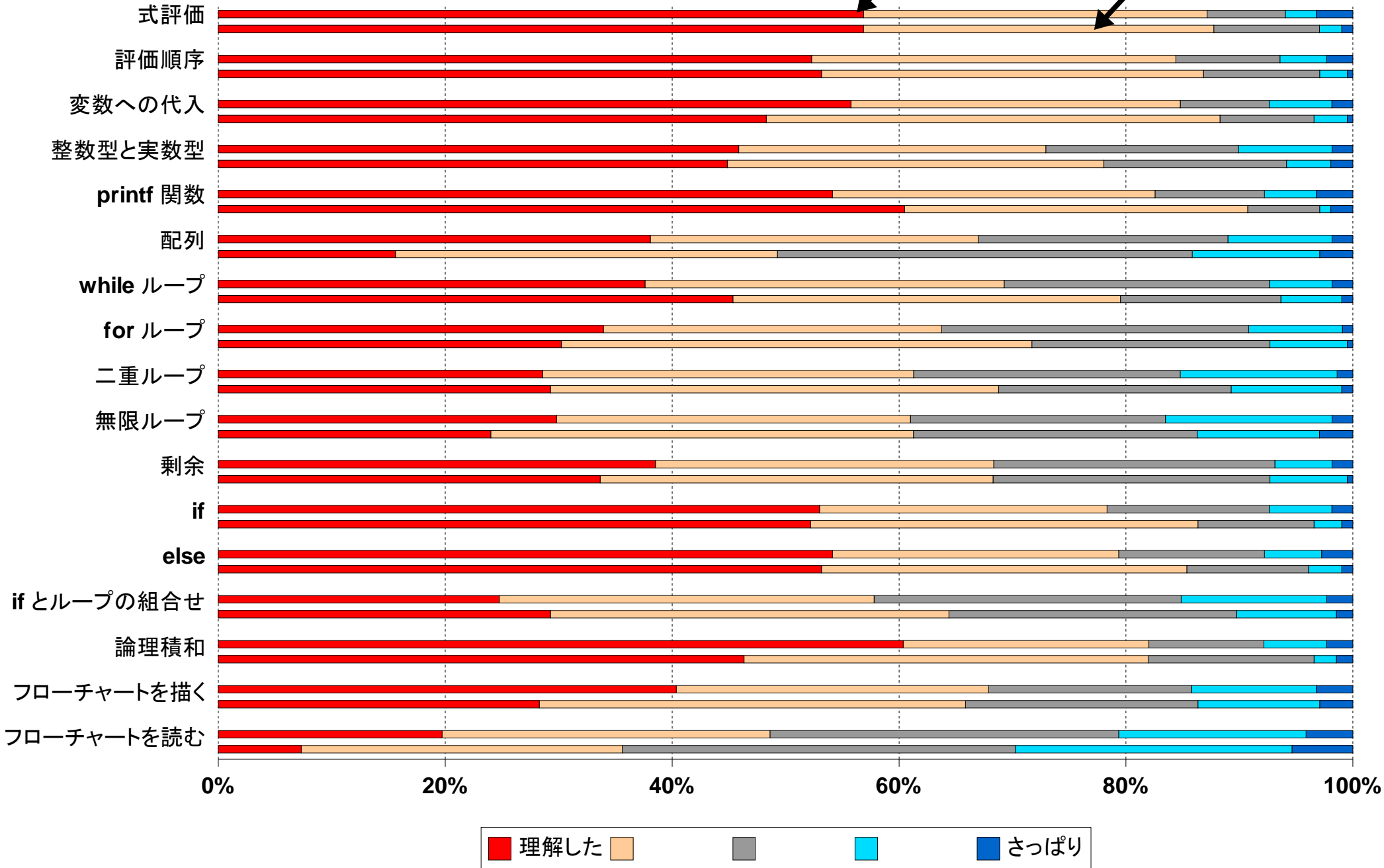
まとめ

- プログラミング学習のための新しい教材の提案と実際のプログラミング実習で使用了結果を報告した
- 提案した教材は、従来型のプログラミング課題で身につけにくい基礎的な部分の理解を補うものである
- ドリルがどの程度学生のプログラミング能力に反映されているのかの評価は今後の検討課題である

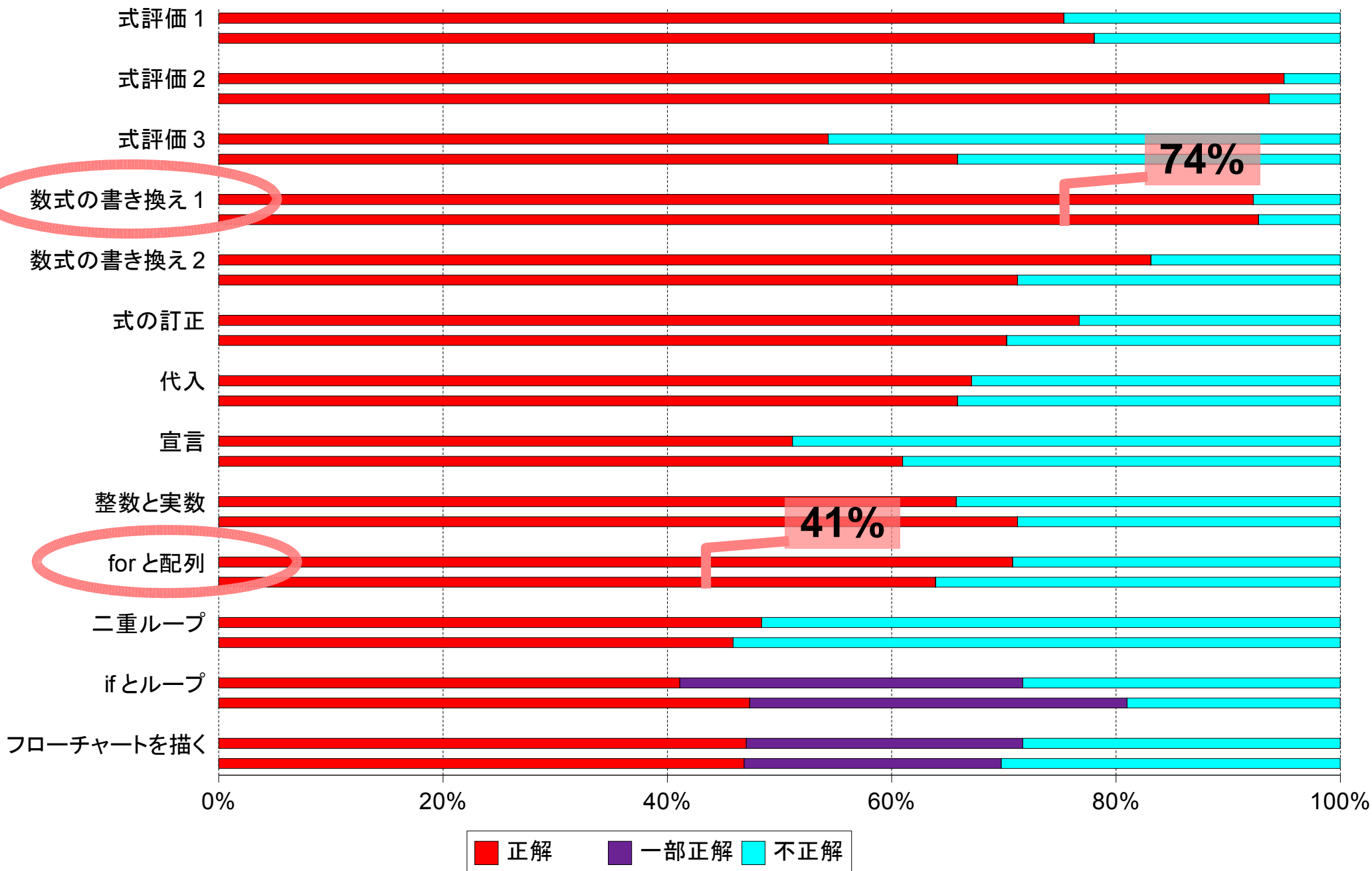
理解度アンケート

1回目(上)

2回目(下)



中間テスト結果



プログラミング言語

Pascal / FORTRAN / C / C++

FORTRAN / C / C++

Perl / Ruby / PHP

Visual C++
Visual Studio .NET

手続き型言語

非手続き型言語

Lisp / Prolog

教育

科学

インターネット

商用
ソフトウェア



有名なサンプルプログラム

C 言語 :

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    printf("Hello World!\n");
    return 0;
}
```

問題 :

- プログラムを入力し、実行する必要がある
- 多数の記号と暗黙の規則がある
- 他のプログラミング言語と共通する部分がない

有名なサンプルプログラム : プログラムの入力

C 言語 :

```
1:  #include <stdio.h>
2:  int main(int argc, char *argv[]) {
3:      printf("Hello World!\n");
4:      return 0;
5:  }
```

行番号 :

- プログラムを見やすくするためのもの
- 実際のプログラムに書いてはならない
- しかし初心者にとってはプログラムの一部に見える

有名なサンプルプログラム : 入力したプログラム

C 言語:

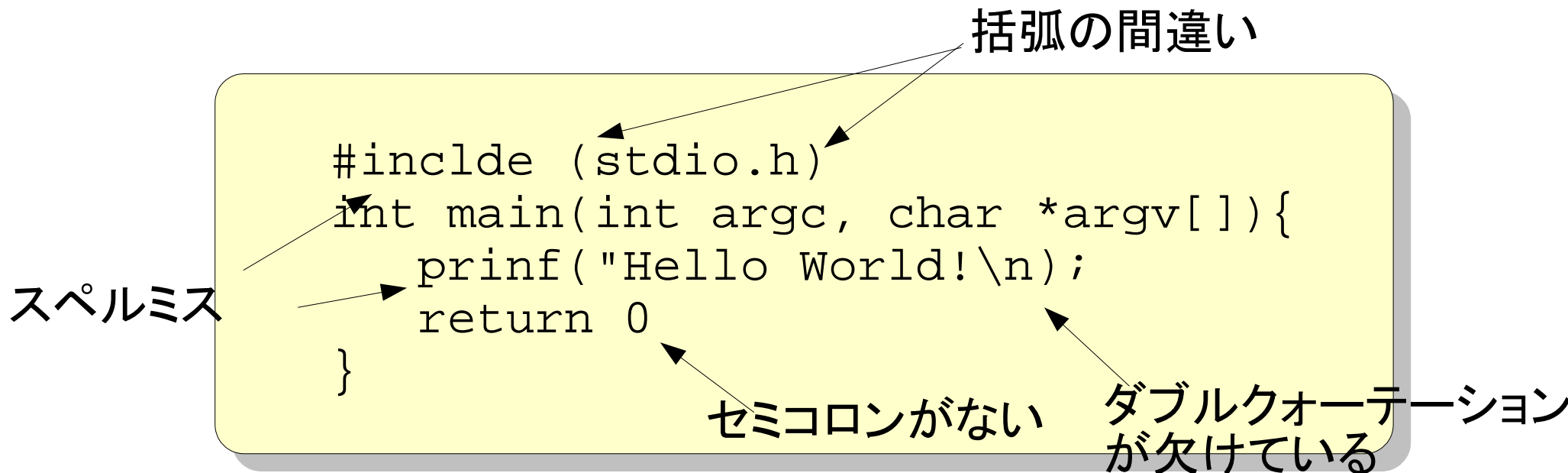
```
1: #include <stdio.h>
2: int main(int argc, char *argv[]){
3:     printf("Hello World!\n");
4:     return 0;
5: }
```

```
#inclde (stdio.h)
int main(int argc, char *argv[]){
    prinlf("Hello World!\n);
    return 0
}
```

有名なサンプルプログラム : 構文ミス

C 言語:

```
1: #include <stdio.h>
2: int main(int argc, char *argv[]){
3:     printf("Hello World!\n");
4:     return 0;
5: }
```



有名なサンプルプログラム :コンパイルエラー

```
test.c: In function `main':  
test.c:1: undefined or invalid # directive  
test.c:1: `#include' expects "FILENAME" or <FILENAME>  
test.c:3: possible real start of unterminated constant  
test.c:3: unterminated string or character constant  
test.c:5: parse error before `}'
```

括弧の間違い

```
#inclde (stdio.h)  
int main(int argc, char *argv[]){  
    printf("Hello World!\n);  
    return 0  
}
```

スペルミス

セミコロンがない

ダブルクォーテーション
が欠けている

有名なサンプルプログラム : 他の言語との互換性

C 言語:

```
#include <stdio.h>
int main(int argc, char *argv[]){
    printf("Hello World!\n");
    return 0;
}
```

Pascal:

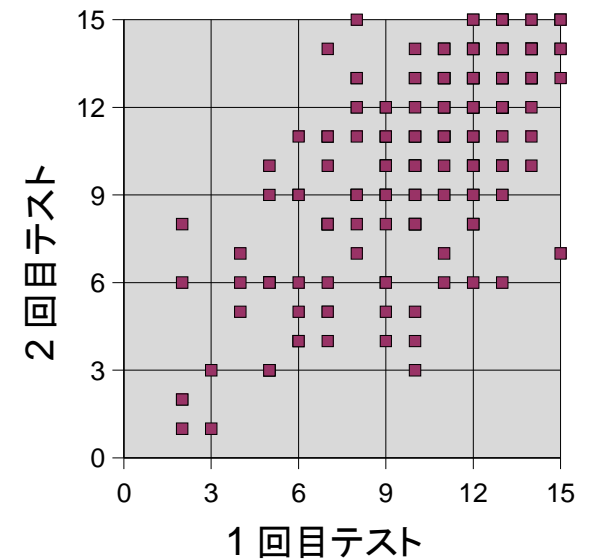
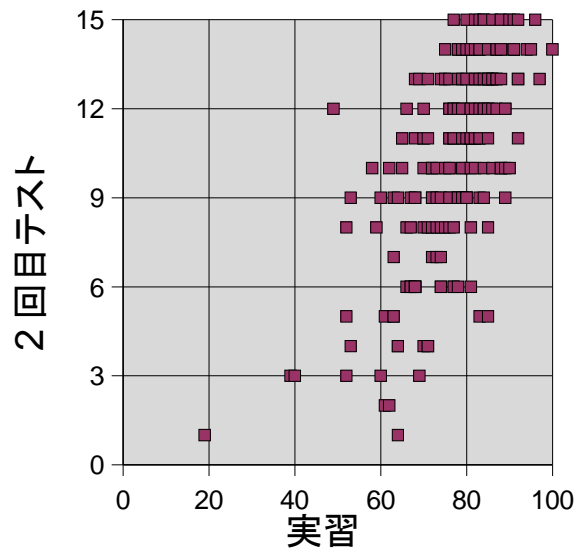
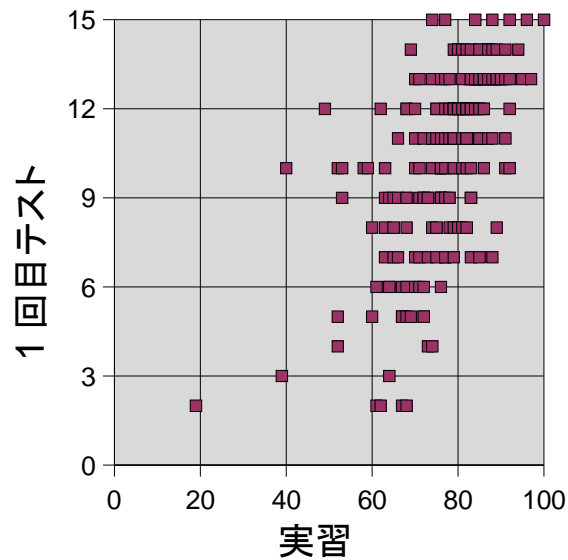
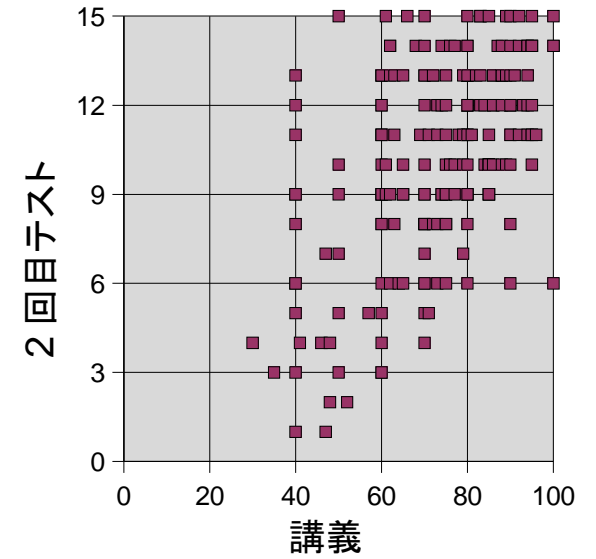
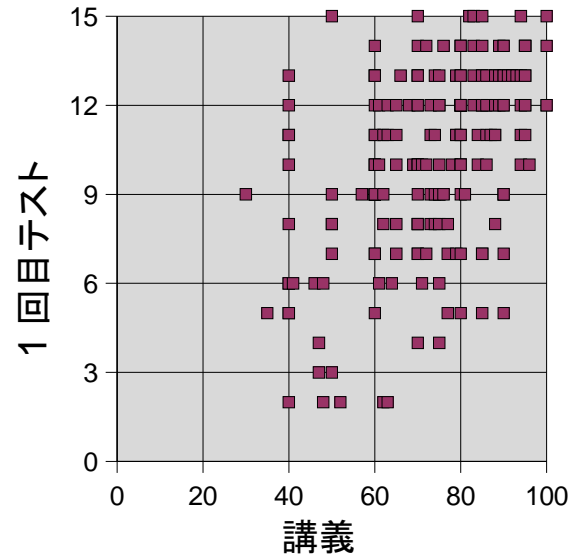
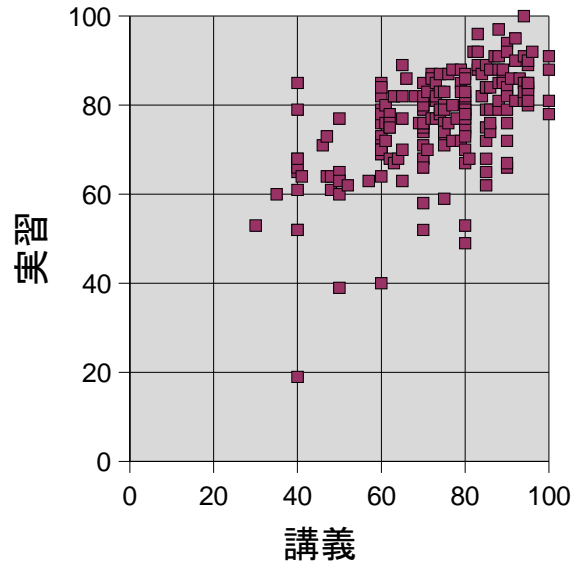
```
program hello(input, output);
begin
    writeln('Hello World!');
end.
```

互換性 :

- それぞれの言語が独自の規則を持つ

基礎ではない！

相関散布図(2003年度)



中間テストと成績の相関

講義、実習の成績と2回の中間テスト結果の相関

	講義	実習	1回目	2回目
講義	1.00	0.59	0.48	0.54
実習	0.59	1.00	0.60	0.66
1回目	0.48	0.60	1.00	0.69
2回目	0.54	0.66	0.69	1.00

成績：

- 講義と実習は別教員が成績を付ける
- 講義：ペーパーテスト
- 実習：二回の中間テスト（各5%）と三つのプログラミング課題のレポート提出（各30%）