

# 競争型学習を取り入れた 入門的 C プログラミング演習 - 授業実践における解答コードの 提出履歴の詳細分析 -

川崎 慎一郎<sup>†</sup> 西村 智治<sup>†</sup> 富永 浩之<sup>†</sup>

大学情報系での入門的 C 授業に競争型学習を取り入れ、小コンテスト形式でのプログラミング演習を支援する大会サーバ tProgrEss を開発している。中間目標として、部分的な仕様に基づく実行テスト系列を用意し、段階的な実装を誘導する。サーバでは、提出されたソースコードの正誤を自動判定し、進捗状況を即時に通知して、競争意欲を刺激する。初級 C 授業の毎回の演習および試験で運用した結果を報告する。成績によって 4 群に分け、提出履歴の時系列的な分析を行った。

## Support for Introductory C Programming Exercise with Contest Style as Competitive Learning - Submission Log Analysis of Answer Codes in Daily Lessons and the Exam -

Shinichiro Kawasaki<sup>†</sup> Tomoharu Nishimura<sup>†</sup>  
Hiroyuki Tominaga<sup>†</sup>

To improve introductory C language lesson in computer engineering college, we have proposed programming exercises with a small contest style adjusted to beginners in classroom. We have developed tProgrEss, the contest management Web server. The server judges an uploaded student's program by an execution test with input and output data. We offer several preparation tests, which are for step-by-step sub-goals as partial specification. We carried out some educational practices as experiments in introductory C programming lesson. We accumulated submission logs in daily lessons and the exam. We divided four student groups in their scores and analyzed them by time series method. By the result, we verified the educational effect.

### 1. はじめに

大学の情報系学科では、初年次の C 言語教育を重視している。しかし、理解度が異なる多様な学生が受講しており、進捗状況や演習態度に大きな差が生じやすい。本研究では、授業中の演習を活性化させるため、初心者が気軽に参加できるコンテスト形式の演習を提案している(1)。また、Web ベースの支援システム tProgrEss を開発している(2)。tProgrEss は、問題の提示、解答プログラムのアップロード、サーバ側でのコンパイル、実行結果の正誤判定などの機能を有する。ただし、初心者がプログラムを完成するには時間がかかり、途中で諦めてしまう可能性がある。そこで、一部の仕様を満たす解答に対しても部分点を与えるようにする。また、各学生の進捗状況を即時的に公開し、競争意欲を高める(図 1)。

コンテストの実施形態として、教室型と宿題型の両者を取り入れる。教室型は、毎回の授業の最後に、総まとめ的な位置付けとして、教室内で一斉に実施する。基本的な事項で、例題の類題として、15~30 分程度で解ける問題を主に出题する。宿題型は、1~2 週間の期間内に、学外からのアクセスも許容して行う。応用的な事項の組合せで、60 分程度の問題を主に出题する。また、教室型を基にした試験型コンテストを取り入れる。教室型より長い時間で、解答に 30 分程度かかる、難易度の高い問題を出题する。制限時間を厳格に設定し、成績に関与するような授業で利用する。

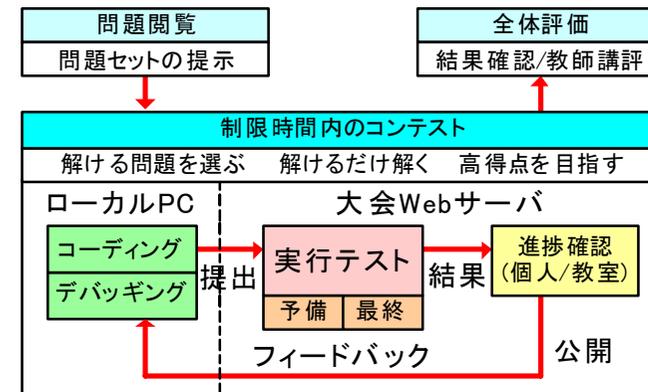


図 1 小コンテスト形式の演習の進行

<sup>†</sup> 香川大学工学部  
Faculty of Engineering, Kagawa University

## 2. 実行テスト系列による正誤判定

本研究では、コーディングの進め方の指針のため、複数の予備テストと1つの最終テストからなる実行テスト系列を採用する(図2)。中間目標としての予備テストは、仕様を緩めて徐々に完成に近付けさせるものである。予備テストは、入力制限と出力許容に分けられる。入力制限では、仕様で想定される範囲内の一部のデータでのみ正しく実行できればよく、例外処理を必要とするもの、継続条件や分岐条件で境界値となるものを避けて実行テストを行う。出力許容では、仕様で要求される出力の書式において、照合基準を緩和し、データ列の平均と最大値を求める問題で、平均だけの出力が合えばよいなどとする。予備テストは、3つ程度の中間目標として構成し、プログラムの作成指針を提示する。

プログラムの判定結果には、時間と誤答による減点を反映した得点ルールを適用する。満点は、標準時間内に誤答なしで最終テストにパスした場合の得点である。ここから、誤答回数と経過時間で減点する。ただし、コンテストの制限時間内に正答すれば、減点を打ち切って、最低でも基準点を与える。最終テストに正答できなかった場合は、予備テストの部分点のみを与え、減点を行わない。

## 3. 大会運用サーバ tProgrEss の機能

本研究では、初級C授業と初心者に応じた、小コンテスト形式のプログラミング演習を提案している。大会運営サーバ tProgrEss を開発し、実際の授業で運用している(3)(4)(5)。学生の解答コードがアップロードされると、図3の処理手順で実行テストが進む。結果は、不正提出、静的エラー、実行打切、動的エラー、誤答、正答の6段階で判定する。提出履歴は全て記録され、コンテストの実施状況や学生の振舞の分析に用いる。

正誤判定において、出力サンプルと実行結果の照合は、予備テストの仕様に応じて柔軟に行う(6)。最終テストおよび入力制限の予備テストでは、文字列としての完全一致でよい。出力許容の予備テストでは、行単位の部分一致や行内の前方一致で、部分出力に対応する。また、単語単位の一致や数値としての相等で、出力書式を許容する。入出力プロンプトやデバッグ出力を含んだプログラムに対しては、出力中のコメント記号で始まる行を無視することで対応する。

学生へのインタフェースとして、判定結果、およびコンパイル時のメッセージと実行結果を、判定確認ページで提示する。学生は、これらの情報を基に、解答の修正や、次のテストに取り組む。また、順位表示ページで、各自の判定結果は教室全体にも公開される。他人の進捗状況を見て、競争意識を高め、意欲を持続させる。教師総覧ページでは、学生の提出したソースコードも閲覧でき、自動判定では見逃される不十分

なコードや理解不足の点に注意を与える。

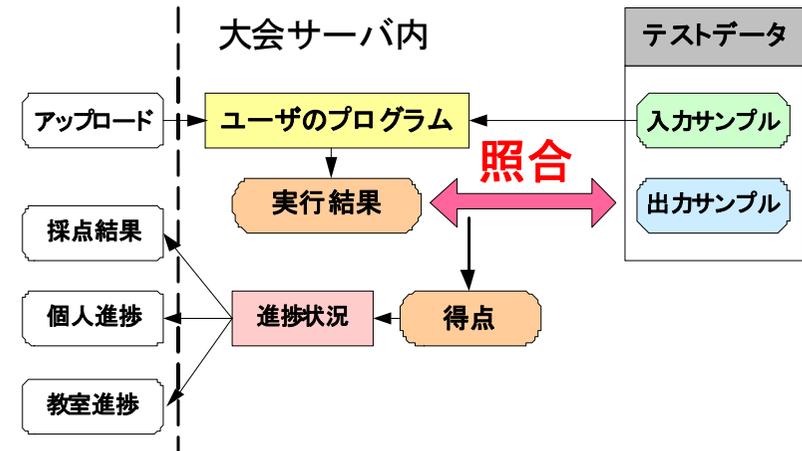


図2 解答プログラムの正誤判定

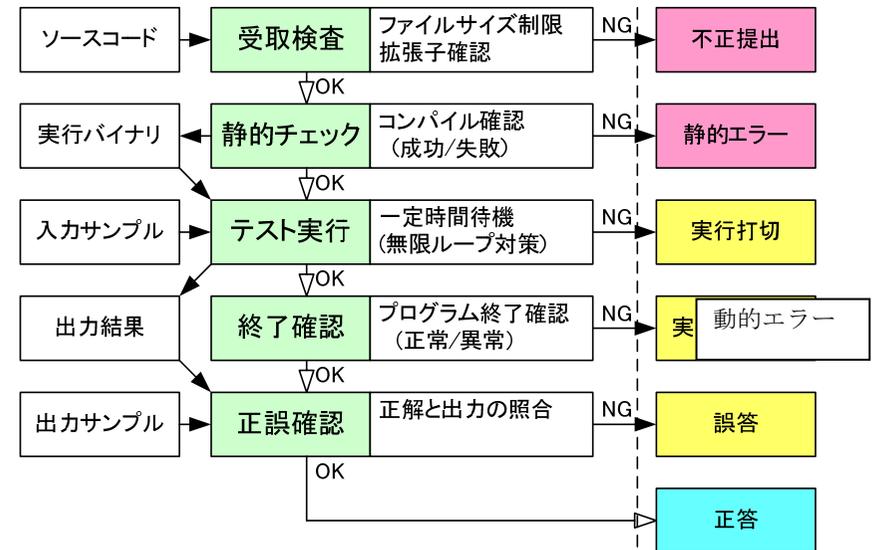


図3 実行テストの処理手順と判定結果

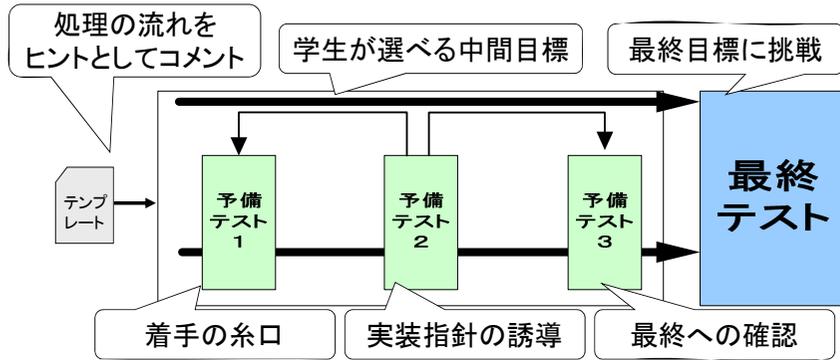


図 4 実行テスト系列の予備テストと最終テスト

不定個の整数値を入力し、平均値を出力する。 負数を番兵とし、小数点以下2桁を出力せよ。		
番兵式による入力の確認		予備テスト1
5 -1	5.000000	
平均値計算の確認		予備テスト2
2 3 4 5 -1	3.500000	
出力書式の確認		予備テスト3
1 2 3 4 5 6 -1	3.50	
十分な量の入出力サンプルを用い、 プログラムの正誤を判定する。		最終テスト

図 6 実行テスト系列の入出力サンプル

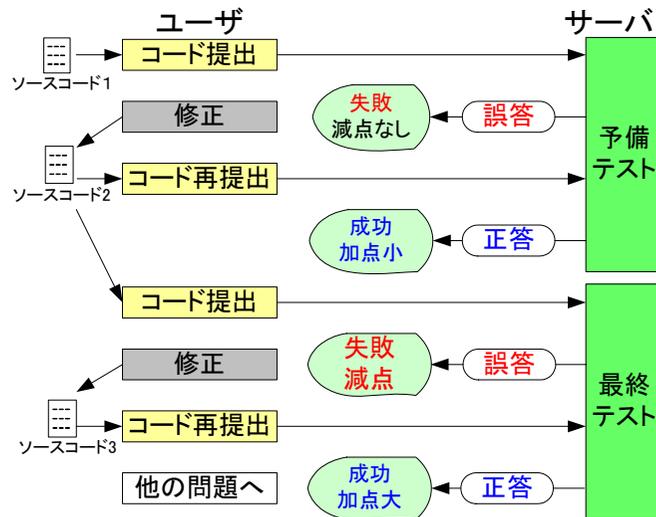


図 5 予備テストと最終テストによる判定の相違

予備テスト1: 入力値をデータとして正しく格納	
入力の順序とデータ型 不定個入力の番兵での打ち切り	不適な入力範囲の確認 入力値の計数と振分け
予備テスト2: 主要な反復処理や配列操作	
反復の構造(パス、ネスト) 反復の範囲(初期値、継続条件)	配列のシフト(追加・削除・逆順) 配列の加工(分割・併合・生成)
予備テスト3: 整形出力や例外処理	
整形出力(桁数、区切、表形式)	例外処理(レアケース) 境界値検査

図 7 実行テスト系列における確認項目

予備テスト進捗

最終テスト進捗

総合得点

ランキング

テスト情報の切換え

図 8(a) 問題閲覧ページの出題解答モード

正誤と得点表示

コンパイルエラー表示

予備のみサンプル表示

提出コード確認

図 8(b) 問題閲覧ページの採点結果モード

予備テスト進捗

最終テストの成功/失敗

総合得点

順位

図 8(c) 順位表示ページ

予備・最終の進捗

結果の再表示ボタン

図 8(d) 提出履歴ページ

表 1 対象授業のシラバス

週	授業内容	理解目標
第1週	復習編(1) 配列と制御構文	配列のプログラミング(走査・更新・格納)
	配列とデータ構造、カーソル表現	不定個のデータ処理を配列で実現する
第2週	復習編(2) 再帰的処理、漸化式と計算過程	漸化式の実装として再帰的な関数を作成する
	変数と宣言、有効範囲、静的変数、型定義	反復法と再帰法の計算過程の相違を理解する
第3週	復習編(3) 配列と関数、多次元配列の構造	配列の配列として二次元配列を理解する
	行列と整式の算法、多進数、前処理	二次元配列を関数の引数として利用する
第4週	復習編(4) プログラムの設計と実装	関数の集合体としてプログラムを設計する
	配列の応用、ビット演算、テスト、デバック	段階的にプログラムを実装して検証する
第5週	復習編(5) プログラムの設計と実装	関数の集合体としてプログラムを設計する
	乱数、実行時間、テスト、デバック	段階的にプログラムを実装して検証する
第6週	中級編(1) ポインタ、アドレス、変数の別名	ポインタ/アドレス演算を理解する
	配列とポインタ、関数引数の名前参照	関数引数の名前参照に利用する
第7週	中級編(2) 文字と文字列、空文字と文字列長	文字列処理の関数を利用する
	文字列とポインタ、文字列関数	文字列リテラルの処理(列長・出現・比較)
第8週	中級編(3) 文字種判定、文字単位の入出力	ポインタの更新による文字列処理
	文字計数、文字出現	計数・検出のプログラミング
第9週	中級編(4) 文字列処理の応用	ポインタの更新による文字列処理
	字句検出、文字列検索	照合・検索のプログラミング
第10週	中級編(5) 構造体とメンバ	抽象データ(有理分数、複素数)と計算
	宣言と定義、参照と代入、抽象データ	図形データ(ベクトル、三角形)と計算
第11週	中級編(6) 構造体の応用	構造体によるデータ表現(成績表、住所録)
	構造体と配列、構造体とポインタ	構造体の配列処理(集計・検索・整列)
第12週	中級編(7) ファイル入出力、読込と書出	行単位のファイルの読込と書出
	オープンとクローズ、上書と追加	テキスト加工、対話処理
第13週	中級編(8) ファイル入出力、読込と書出	文字単位のファイルの読込と書出
	オープンとクローズ、上書と追加	簡易DB、バイナリファイル
第14週	総合編	総合的な問題に取り組む
	中規模サイズのプログラム作成	
第15週	総合編	総合的な問題に取り組む
	中規模サイズのプログラム作成	

表 2 対象授業の日程

授業回	日付	形式	学習事項
第01回	2010.04.07(水)	講義	授業の概要、受講の注意、コンパイル時のエラーメッセージ
第02回	2010.04.08(木)	演習	授業の概要、受講の注意、コンパイル時のエラーメッセージ
第03回	2010.04.14(水)	講義	制御構造、プログラムの構成とコメント
第04回	2010.04.15(木)	演習	制御構造、プログラムの構成とコメント
第05回	2010.04.21(水)	演習	演算子と式、変数の有効範囲と有効期間
第06回	2010.04.22(木)	講義	演算子と式、変数の有効範囲と有効期間
第07回	2010.04.28(水)	演習	関数形式マクロ、関数の再帰的定義、DOS端末とリダイレクト
第08回	2010.05.06(木)	講義	関数形式マクロ、関数の再帰的定義、DOS端末とリダイレクト
第09回	2010.05.12(水)	講義	文字列、文字列処理、文字列の入出力
第10回	2010.05.13(木)	演習	文字列、文字列処理、文字列の入出力
第11回	2010.05.19(水)	講義	書式付の入出力、文字種の判定と変換
第12回	2010.05.20(木)	演習	書式付の入出力、文字種の判定と変換
第13回	2010.05.26(水)	講義	ポインタ、関数の引数とポインタ、時間関数と乱数
第14回	2010.05.27(木)	演習	ポインタ、関数の引数とポインタ、時間関数と乱数
第15回	2010.06.02(水)	講義	配列とポインタ、配列を引数とする関数、ベクトルと行列の関数
第16回	2010.06.03(木)	演習	配列とポインタ、配列を引数とする関数、ベクトルと行列の関数
第17回	2010.06.09(水)	講義	文字列とポインタ、文字列の検索と照合
第18回	2010.06.10(木)	演習	文字列とポインタ、文字列の検索と照合
第19回	2010.06.16(水)	講義	集合演算、単純整列
第20回	2010.06.17(木)	演習	集合演算、単純整列
第21回	2010.06.23(水)	講義	構造体の宣言、メンバの参照、構造体と抽象データ
第22回	2010.06.24(木)	演習	構造体の宣言、メンバの参照、構造体と抽象データ
第23回	2010.07.07(水)	講義	構造体とポインタ、複合データ、共用体、関数とポインタ
第24回	2010.07.08(木)	演習	構造体とポインタ、複合データ、共用体、関数とポインタ
第25回	2010.07.14(水)	講義	実行時引数、ファイルアクセス、ファイル入出力
第26回	2010.07.15(木)	演習	実行時引数、ファイルアクセス、ファイル入出力
第27回	2010.07.21(水)	講義	ファイルの基本処理、簡易データベース
第28回	2010.07.22(木)	演習	ファイルの基本処理、簡易データベース

## 4. 毎週の授業における教室型と宿題型の運用実践

### 4.1 授業概要

tProgrEss を教育実践に利用し、運用評価を行った。対象科目は、情報系学科 2 年次前期の初級 C 演習「プログラミング II」である(表 1)。本授業は、学科 1 年次後期の「プログラミング I」で、C 言語の基礎文法を習得した学生が履修する。なお、「プログラミング I」においても運用実験を行っているため、学生は本システムの利用経験がある(7)(8)。

受講者は学科 2 年生 64 名である。学生のほとんどは情報系だが、一部に電子系も含まれている。学習範囲は、文字列、ポインタ、構造体、ファイル入出力である。「プログラミング II」では、一週間で水曜と木曜に授業があり、それぞれ講義(2 コマ)と演習(1 コマ)に分けて授業を行う(表 2)。講義では教室型、演習では宿題型の授業実践を実施した。また、期末試験において、試験型の授業実践を実施した。

### 4.2 教室型コンテストの実施と分析

原則として、講義では、1 コマ目で文法説明や例題解説の授業が行われ、2 コマ目から演習を実施する。教室型コンテストの出題は、2 コマ目から行う。問題数は 6 問で、例題の類題レベルの易しい問題を 4 問、応用的な難しい問題を 2 問設ける。予備テストは、易しい問題には 1 個、難しい問題には 2 個用意する。基本的に、90 分の授業時間を解答時間とする。TA3 名による補助も行う。また、授業後の提出も受理する。当日または翌日中の提出を締切とする。授業の進行状況によって、締切は前後する。

予定通りに行われたコンテストの 1 つを取り上げ、提出履歴の分析を行った(表 3)。最終成績別の学生の振舞いの特徴について分析するため、期末試験の結果から、成績の上位 34 名と下位 30 名の集団に分けた。ほぼ同数として分析を行う。なお、集団の詳細は 5 章で述べる。それぞれの提出数と実行テスト系列別の提出比率、正答率を表 5 に示す。また、開始から 10 分ごとの実行テスト系列別の提出履歴および正答率のグラフを図 9 に示す。

上位は、予備 1 の提出比率が 0.40 で、下位の 0.31 より 0.09 高い。また、60 分の時点で第 1 の提出ピークを迎えている。予備 1 を上手く利用し、早期の解答着手と部分点の取得に繋げている。一方、下位については、最終の提出比率が 0.56 で、上位の 0.49 より 0.07 高い。また、第 1 の提出ピークが 110 分ごろで、上位より 50 分も遅い。実行テスト系列を十分に活用できず、他の問題への着手が遅れている。さらに、最終の正答率が 0.83 で、上位の 0.89 より 0.06 低い。誤答減点にも影響が出ている。予備テストの積極的な利用を促す必要がある。

### 4.3 宿題型コンテストの実施と分析

原則として、演習では、授業時間の全てを宿題型コンテストの問題解答に充てる。問題数は 8 問で、宿題型の基本問題を 4 問、より応用的で難しい問題を 4 問設ける。解答時間は、前者が 30 分、後者が 1 時間程度のものを用意する。授業後の 90 分程度は TA2 名が残り、学生の問題解答の補助を行う。予備テストは、基本問題には 1 個、応用問題には 2 個用意する。授業の進行状況によって、締切は前後する。

予定通りに行われたコンテストの 1 つを取り上げ、提出履歴の分析を行った(表 4)。教室型と同様に、期末試験の成績の上位 34 名と下位 30 名の集団に分ける。それぞれの提出数と実行テスト系列別の提出比率、正答率を表 6 に示す。また、開始から 6 時間ごとの実行テスト系列別の提出履歴および正答率のグラフを図 10 に示す。

上位は、初日の提出が 30 件ほどあり、下位の 5 件より 6 倍も多い。授業日以外についても、コンスタントに一日当たり 50 件程度の提出がある。また、土日の提出を合わせると 110 件以上あり、上位の全提出 588 件の 2 割を占める。予備 2 の提出比率は、下位に比べて 0.10 も高い。取り掛かりが早いいため、応用問題への着手に到達している。予備 2 の利用により、慎重な解答を行っている。一方、下位については、初日の提出が少なく、授業日以外の提出もほとんど 30 件以下である。特に、金曜から日曜は、ほとんど提出できていない。しかし、締切直前は、上位と同程度の提出がある。正答率は、どの予備テストについても、上位よりやや低いが大差はない。予備 2 の正答率については、下位は 0.83 で上位 0.69 に対して 0.14 も高い。これは、下位の予備 2 の提出件数が低いことから、自信のある解答のみを提出しているためと考えられる。取り掛かりが遅く、締切直前に提出が集中している。早期の提出を促す仕組みが必要である。

表 3 教室型の実施概要

授業	第 17 回 講義
形式	教室型
名称	W09_C01
開始	2010.06.09(水) 16:20:00 (5 コマ)
期間	同日中 (16:20-19:20 の 2 時間程度)
出題	文字列の入力と字句解析
問数	6 問 (易 4+難 2)
予備	易 1 個、難 2 個
配点	易 100(50/100)、難 150(45/105/150)
合格	400 点以上

表 4 宿題型の実施概要

授業	第 20 回 演習
形式	宿題型
名称	W10_H01
開始	2010.06.23(水) 15:30:00(4 コマ)
期間	1 週間
出題	集合演算と単純整列算法
問数	8 問 (易 4+難 4)
予備	易 1 個、難 2 個
配点	易 100(50/100)、難 150(45/105/150)
合格	400 点以上

表5 教室型の分析結果

			合計	比率	正答率
全員	全体	提出	649		0.71
全員	予備1	提出	231	0.36	0.58
全員	予備2	提出	82	0.13	0.49
全員	最終	提出	336	0.52	0.86
全員	全体	正答	464		
全員	予備1	正答	134	0.29	
全員	予備2	正答	40	0.09	
全員	最終	正答	290	0.63	
上位	全体	提出	356		0.71
上位	予備1	提出	141	0.40	0.52
上位	予備2	提出	42	0.12	0.55
上位	最終	提出	173	0.49	0.89
上位	全体	正答	251		
上位	予備1	正答	74	0.29	
上位	予備2	正答	23	0.09	
上位	最終	正答	154	0.61	
下位	全体	提出	293		0.73
下位	予備1	提出	90	0.31	0.67
下位	予備2	提出	40	0.14	0.43
下位	最終	提出	163	0.56	0.83
下位	全体	正答	213		
下位	予備1	正答	60	0.28	
下位	予備2	正答	17	0.08	
下位	最終	正答	136	0.64	

表6 宿題型の分析結果

			合計	比率	正答率
全員	全体	提出	803		0.75
全員	予備1	提出	348	0.43	0.59
全員	予備2	提出	163	0.20	0.72
全員	最終	提出	292	0.36	0.97
全員	全体	正答	604		
全員	予備1	正答	204	0.34	
全員	予備2	正答	117	0.19	
全員	最終	正答	283	0.47	
上位	全体	提出	588		0.75
上位	予備1	提出	247	0.42	0.59
上位	予備2	提出	134	0.23	0.69
上位	最終	提出	207	0.35	0.98
上位	全体	正答	441		
上位	予備1	正答	146	0.33	
上位	予備2	正答	93	0.21	
上位	最終	正答	202	0.46	
下位	全体	提出	215		0.76
下位	予備1	提出	101	0.47	0.57
下位	予備2	提出	29	0.13	0.83
下位	最終	提出	85	0.40	0.95
下位	全体	正答	163		
下位	予備1	正答	58	0.36	
下位	予備2	正答	24	0.15	
下位	最終	正答	81	0.50	

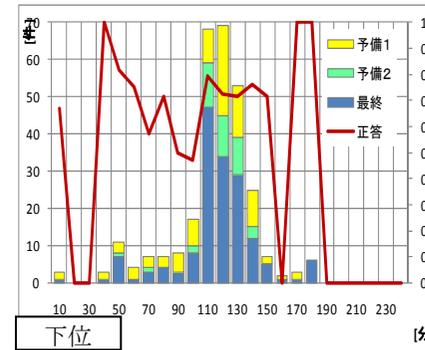
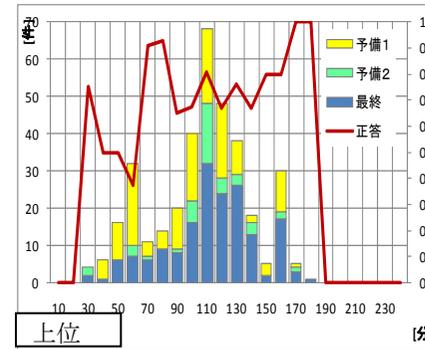
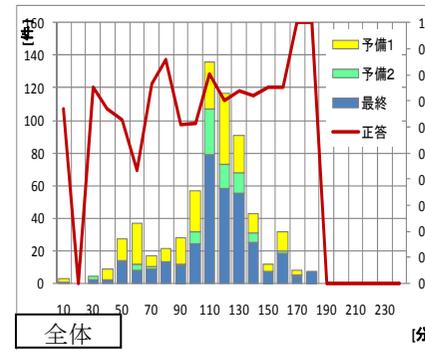


図9 教室型の時系列の提出グラフ

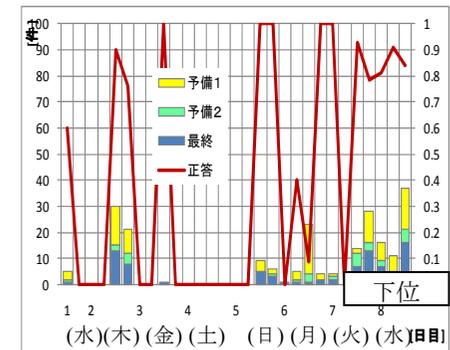
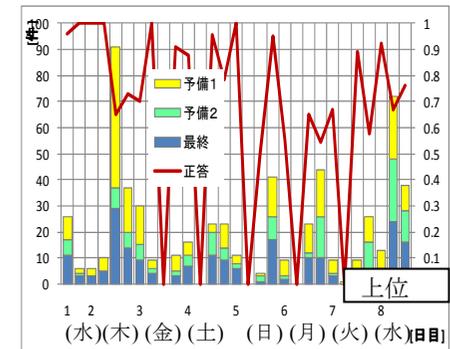
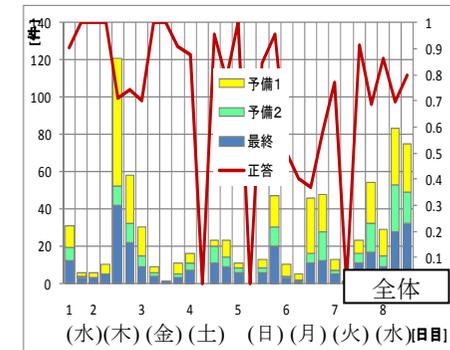


図10 宿題型の時系列の提出グラフ

## 5. 期末試験における試験型の授業実践

### 5.1 試験型コンテストの実施概要

期末試験において、試験型コンテストの運用評価を行った(表 7)。受験者は、その時点で受講者全員で 64 名である。制限時間は 185 分で 8 問を出題した。出題範囲は、文字列のポインタ処理、構造体と整列算法である。各問は、解答の所要時間が 30 分程度である。配点は、最終テストを 150 点とし、予備 1・2 を、それぞれ 45/105 点とした。合格基準としての目標得点を 400 点とし、完答 3 問よりも半答 7 問を目指すように指導した。なお、8 問目は、この時点で難易度の高い文字列集合の整列問題とした。

### 5.2 試験型コンテストの結果分析

コンテストの結果として、得点分布を図 11、参加状況を表 8 に示す。教室型や宿題型と異なり、厳密な制限時間内での総合得点にはばらつきがある。そこで、試験型では参加者 64 名を表 10 の第 1~4 群の集団に分ける。教室型と宿題型の分析では、第 1~2 群を上位、第 3~4 群を下位とした。なお、第 4 群は、400 点未満の不合格者の集団である。集団ごとの、実行テスト系列の提出割合と正答比率を表 11 に示す。また、時系列の提出件数と正答率の推移グラフを図 12 に示す。

第 1~3 群について、全体の提出数は、約 340 件で同程度であった。第 1 群は、予備 2 と最終を中心に、コンスタントな提出を行っている。終盤は、部分点狙いである。第 2 群は、予備 1 から最終までをバランスよく提出している。第 3 群は、予備 1・2 を中心に提出している。能力に応じた実行テスト系列の利用は、こちらの想定通りと言える。一方、4 群については、提出数が他 3 群に比べて半分以下であった。提出の内訳は、予備 1 と 2 を合わせると 9 割を占めている。なお、第 4 群のほとんどは、電子系の学生であった。教育資源の集中が必要である。

表 7 期末試験の実施概要

授業	2010 年度 後期「プログラミングⅡ」
対象	2 年生 64 名
形式	試験型
期間	185 分
名称	A03 E01
問数	8 問 (各予備×2)
配点	各 150 点 (45/105/150)
学習	文字列処理、ポインタ、構造体
題材	整式の累乗、単純選択法、

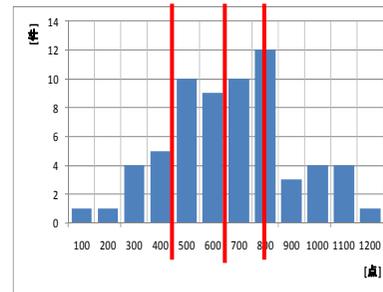


図 11 期末試験の得点分布

表 8 試験型コンテストの学生の参加状況

参加人数	総提出数	平均提出	教室閲覧	教室平均
64	1141	17.83	737	11.52

表 9 期末試験の出題

No	テスト	出題
1		2個の有理分数 F1, F2 を帯分数として入力し、除算の商 F3 および二乗和 F4 を出力する。
	予備 1	入力において、各整数は非負とする。出力は、除算の商 F3 の行のみでよい。
	予備 2	出力は、除算の商 F3 の行のみでよい。
2		整数を係数とする整式 E1 の m 乗 E0 を求める。
	予備 1	入力は、1次式とする。
	予備 2	入力は、m=2 とする。
3		単純選択法で、辞書式順序での昇順の整列を行う。
	予備 1	n=4 で、各単語は1文字のみとする。比較回数と交換回数は、出力しない。
	予備 2	n=4 とする。
4		不定個の文字列の入力を集合 set[] に格納する。
	予備 1	全て英小文字とし、要素の重複はない。
	予備 2	全て英小文字とする。
5		木構造を入力し、指定ノードのパスを出力する。
	予備 1	木構造として、枝分かれがなく、入力順で一本道になっている。
	予備 2	ノード数 n=3 の場合に限定する。
6		入力プロンプト "Regist ?" の後、受講レコード数を指定する。
	予備 1	受講と担当が1件ずつのみとする。
	予備 2	受講と担当がそれぞれ2件以下とする。
7		複素数を係数とする整式 E に複素数 Z を代入した値を求める。
	予備 1	虚部が 0 とする。
	予備 2	2次式までとする。
8		単語集合を構造体に単純挿入法で整理しながら挿入する。
	予備 1	各行1件のデータしかない。
	予備 2	1行目は整列済で、他の2行は1件のデータしかない。

表 10 成績による学生集団の分類

集団	順位	得点	人数
第1群	最上位	750-1200	17名
第2群	中上位	600-749	17名
第3群	中下位	400-599	19名
第4群	最下位	000-399	11名

表 11 試験型の分析結果

集合			合計	割合	正答率
全員	全体	提出	1141		0.59
	予備1	提出	440	0.39	0.39
	予備2	提出	410	0.36	0.62
	最終	提出	291	0.26	0.87
最上位	全体	提出	341		0.68
	予備1	提出	87	0.26	0.36
	予備2	提出	134	0.39	0.68
	最終	提出	120	0.35	0.93
中上位	全体	提出	340		0.65
	予備1	提出	123	0.36	0.50
	予備2	提出	123	0.36	0.65
	最終	提出	94	0.28	0.84
中下位	全体	提出	312		0.54
	予備1	提出	149	0.48	0.40
	予備2	提出	106	0.34	0.57
	最終	提出	57	0.18	0.89
最下位	全体	提出	133		0.41
	予備1	提出	81	0.61	0.26
	予備2	提出	47	0.35	0.47
	最終	提出	20	0.15	0.60

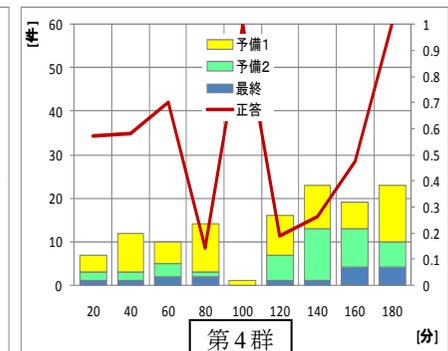
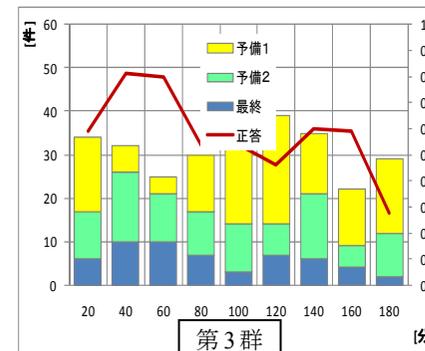
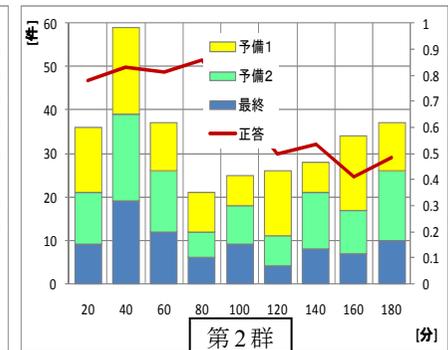
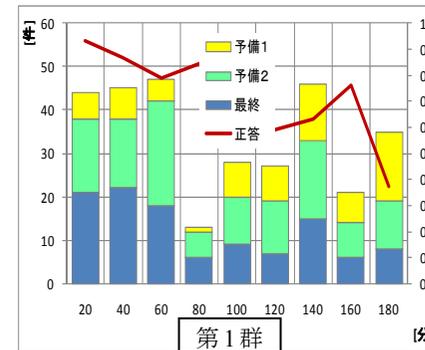
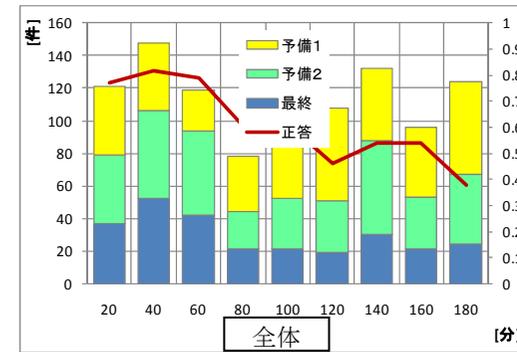


図 12 試験型の時系列の提出グラフ

## 6. おわりに

大学情報系学科の入門的 C 授業において、初心者向けの小コンテスト形式でのプログラミング演習を提案し、大会運営の支援サーバ tProgrEss を開発した。tProgrEss は、サーバ側に提出された解答コードを自動採点し、コンテストの進捗状況を即時に表示する。中間目標としての実行テスト系列による正誤判定に対応するため、実行結果と出力サンプルとの柔軟な照合機能を実現した。初級 C 演習の授業で、毎回の講義と演習で教室型と宿題型、期末試験で試験型の授業実践を行った。期末試験での成績を基に学生を 2~4 群に分け、それぞれの提出履歴を分析した。

教室型では、上位が実行テスト系列を上手く利用できていることが分かった。一方、下位は最終の提出が多く、段階的なプログラミングへの意識と慎重な確認が不十分であった。プログラミング中に行き詰まる点は、コンテスト中でも教師側で発見できるように、システムでサポートする必要がある。宿題型では、着手状況に明確な違いが見受けられた。上位は、休日も利用して積極的に提出している。コードの正誤を即時確認できる、自宅からでも提出できるなど、システム利用のメリットが結果に表れている。下位は、初日から解答提出が少なく、着手が極端に遅いことが分かった。提出状況がよくない学生へ、提出催促を自動通知するなどの対応が必要である。

今後の課題としては、教師側の GUI の機能不足が挙げられる。全体への通知や個人への助言など、即時的な指導への支援の強化が必要である。現状の tProgrEss では、各学生の問題への着手状況は、指導側で即時的に把握することが難しい。例えば、100 名以上の学生の中から、提出が滞っている者を見つけることは困難である。そこで、教員の指導を補助するモニタリング機能を追加する。例えば、提出が滞っている学生を自動通知する機能が考えられる。ポップアップや色分けなどで、教員による監視を支援する。進捗状況の検出には、システムログを利用する。ユーザの行為を時系列で分析して、演習態度の注意やヒントの提示を行う。実行結果の正誤判定だけでなく、教師の目視によるコード確認も行い、各学生への個別指導を支援する機能も導入する。これらを踏まえ、運用実験を行い、教育効果を検証する。

## 参考文献

- 1) 村井万寿夫, 学習意欲を高めるための手立てについて, JSET 研究報告集, JET03-4, pp.31-36, (2003).
- 2) 倉田英和, 富永浩之, 林敏浩, 垂水浩幸, 山崎敏範, 実行テストを用いたコンテスト形式の入門的 C プログラミング演習の大会運営サーバの開発, 情報処理学会 研究報告, Vol.2006, No.108, pp.9-16, (2006).
- 3) 倉田英和, 富永浩之, 林敏浩, 垂水浩幸, 実行テストによるプログラム判定を用いた初級 C プログラミング演習支援と授業実践, 情報処理学会 研究報告, Vol.2007, No.101, pp.11-18, (2007).

4) 富永浩之, 倉田英和, コンテスト形式による初級 C プログラミングの演習支援, 情報処理学会 研究報告, Vol.2008, No.42, pp. 49-56, (2008).

5) 川崎慎一郎, 富永浩之, 競争型学習を取り入れた入門的 C プログラミング演習 - 演習支援サーバ tProgrEss の出題解答と採点結果のページ表示の改良 -, 信学技報, Vol.109, No.335, pp.187-192, (2009).

6) 川崎慎一郎, 富永浩之, 競争型学習を取り入れた入門的 C プログラミング演習 - 実行テスト系列による部分採点のための柔軟な照合機能 -, 情報処理学会 研究報告, Vol.2010-CE-103, No.10, pp.1-8, (2010).

7) 川崎慎一郎, 富永浩之, 実行テスト系列と自動判定を取り入れた入門的 C プログラミング演習の運用実験, 教育システム情報学会 全国大会, 第 35 回, pp.165-166, (2010).

8) 富永浩之, 川崎慎一郎, 競争型学習を取り入れた入門的 C プログラミング演習 - 運用実験での実行テスト系列の利用状況 -, 情報処理学会 研究報告, Vol.2010, No.4, pp.1-12, (2010).