

小コンテスト形式の初級 C 演習における教師支援 — 解答プログラムの提出状況と得点推移による モニタリング機能 —

西村智治[†] 青木辰徳[†] 富永浩之[†]

実行テスト系列を取り入れた小コンテスト形式の初級 C 演習を提案し、大会運営サーバ tProgrEss を開発している。サーバでは、提出されたソースコードの正誤を自動判定する。中間目標で部分点を与える実行テスト系列を用意し、時間調整点を含む得点ルールを設定する。進捗状況を即時通知し、順位公開して、競争意欲を刺激する。システムの教師支援としては、モニタリング機能とチュータリング機能を提供する。本論では、モニタリング機能の概略と基本機能を述べる。運用実践の結果を分析し、成績別の特徴や典型的な振舞いパターンについて議論する。また、今後の分析に必要な、積算得点と提出履歴を組み合わせた進捗推移グラフのリアルタイム表示を実装した。

Teacher Support for Introductory C Programming Exercises Based on Contest Style Using Execution Test Series - Viewing Functions for Combined Information about Point Integration and Submission Logs as a Monitoring Subsystem -

TOMOHARU NISHIMURA[†] TATSUNORI AOKI[†] HIROYUKI TOMINAGA[†]

We have proposed a small contest style exercise in classroom and homework. We have developed tProgrEss, the contest management Web server. It judges an uploaded program by an execution test with input and output data. To adjust a contest for beginners, we offer several preparation tests, which are for step-by-step sub-goals as partial specification. We set the rule of partial points and plus or minus points based on time. The system shows the result of judgment and exhibits the ranking page immediately. The system also contains monitoring and tutoring subsystems as teacher support. This paper describes the outline of the monitoring subsystem. We analyze results of an experiment in educational practice. We discuss some features of classified groups by scores and behavior patterns of students. We implemented real-time viewing functions of progress transition graph combined with point integration and submission logs.

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[1]。本研究室では、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小コンテスト形式の演習を提案している[2][3]。1 回のコンテストで数問のプログラミング問題を提示する(図 1)。ローカル PC で解答コードを作成し、大会 Web サーバ tProgrEss にアップロードする。サーバ側でコンパイルした実行バイナリに、入力サンプルを与えて実行し、実行結果と出力サンプルを照合して、正誤を判定する(図 2)。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、実行テスト系列を導入する[4]。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。本論では、運用実践の結果を踏まえ、教師側への支援機能を紹介する。

2. 演習形式と大会支援サーバ tProgrEss の概要

2.1 コンテスト形式の演習の実施形態

コンテストの実施形態として、大きく分けて、教室型/自習型/試験型/宿題型/課題型を用意する。これらの型は、異なる機能を提供するのではなく、コンテストの要項や問題編成の指針を示すテンプレートである(図 3)。

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。基本的な文法事項で、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。教室型の変種として、自習型と試験型を用意する。自習型は、オープンな出題とし、受験者が解答を開始した時点から一定の間を試験期間とする。いつでもできる自習用として利用する。試験型は、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。制限時間を厳格にし、成績に関与するような授業で利用する。

宿題型は、指定された開始時期から 1~2 週間の期間で、学外からのアクセスも許容して行う。情報数学、数値計算、統計処理などを題材に、応用的な事項の組合せで、60~90 分程度の問題を主に出题する。課題型は、その変種として、

[†]香川大学

[†]Kagawa University

オープンな出題とし、受験者が解答を開始した時点から一定の間を試験期間とする。長期間の休暇の課題で利用する。

2.2 実行テスト系列による正誤判定

コーディングの進め方を誘導したり、プログラムの作成指針を示すため、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4)。予備テストは、題意の仕様を緩めたもので、3つ程度までの中間目標として構成し、徐々に完成に近付けさせるものである。予備テストは、入力限定と出力許容に分けられる[5]。入力限定では、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い。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、照合基準を緩和したものである。例えば、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(図5)。初級Cの制御構造の範囲では、実行テスト系列による確認項目は、図6のようになる。

2.3 誤答減点と時間調整点による得点ルール

プログラムの判定結果には、実行テストの種別、および時間調整点と誤答減点を導入した得点ルールを適用する。時間調整点は、提出時間によって、通常の配点に加減点を付与する(図7)。提出時期を、事前、早期、通常、延長、事後、無効の6つに区分する。事前期間の提出は、コンテストの開始前なので無効とし、得点を0にする。早期期間の提出は、積極性を評価して加点する。通常期間の提出には、加点も減点もしない。延長期間の提出は、締切の延長という扱いであるため、減点する。事後提出は、最終締切後で、追試的な措置としての提出である。本来の得点の2分の1以下を付与する。無効提出は、授業終了後にシステムの正誤判定の機能のみを使用する際の提出である。得点は0にする。早期と延長における加減点は、提出時期で傾斜させる。これらの時間による得点調整を設けることで、学生の早期解答を促し、演習の活性化を狙う。

一方、誤答減点は、最終テストの誤答に減点のペナルティを与えるものである。最終テストでは、例外的な入力を含む、網羅的な入出力サンプルを用意している。誤答減点により、完答への慎重な確認を行わせる。最終テストに正答できなかった場合でも、予備テストに正答していれば、部分点のみを与え、減点は行わない。なお、時間や誤答による減点は、ある程度で打ち切り、最低でも基準点を与える。これらの得点ルールにより、素早く解答させるコンテスト、じっくり時間をかけて取り組ませるコンテストなど、コンテストの性格を調整することができる。

2.4 大会支援サーバ tProgrEss の機能

上記のような演習を実施するため、大会支援サーバ tProgrEss を開発し、実際の授業で運用している[6]。学生の解答コードがアップロードされると、コンパイルおよび入出力サンプルによる実行テストが行われる。その結果は、

正答、誤答、実行打切、実行時エラー、静的エラー、不正提出の6段階で判定する(図8)。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いの分析に用いる。

学生へのインタフェースとして、問題文、実行テスト、簡易進捗、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、問題閲覧ページの出題解答モード(図9)と、採点結果モードで提示する(図10)。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。教師側ページでは、学生の提出状況や問題の着手状況を、時系列グラフで表示する[7]。また、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。

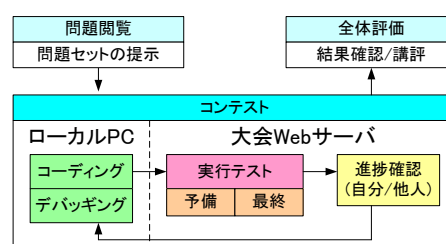


図1 コンテスト形式の演習の進行

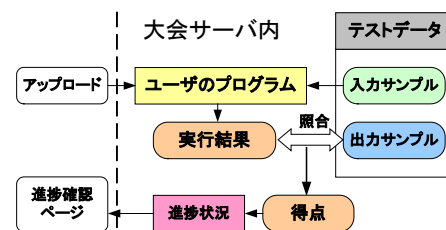


図2 解答コードの判定手順

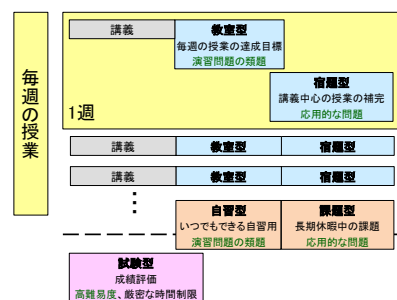
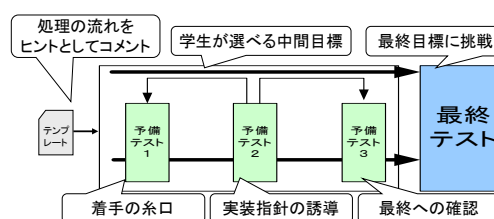


図3 演習形態に応じたコンテスト型



	予備テスト	最終テスト
入出力サンプル	部分的な仕様 データを明示	完全な仕様 データを隠蔽
正答の場合	解ける所まで取り組める部分点	解答時間を差し引いた得点
誤答の場合	減点なしで何度でも提出可能	回数による減点で慎重さを促す

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	5 -1	5.000000
平均値計算の確認	2 3 4 5 -1	3.500000
出力書式の確認	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。		
予備テスト1	予備テスト2	予備テスト3
最終テスト		

図5 実行テスト系列の出題例

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	不適な入力範囲の確認
不定個入力の番兵での打ち切り	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)	配列のシフト(追加・削除・逆順)
反復の範囲(初期値、継続条件)	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6 実行テスト系列における確認項目

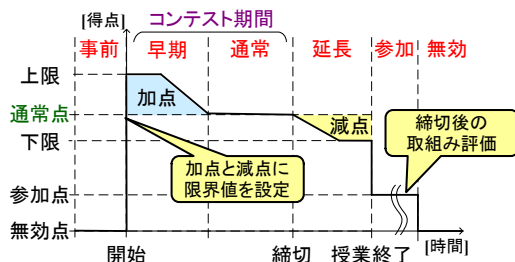


図7 提出時期による時間調整点

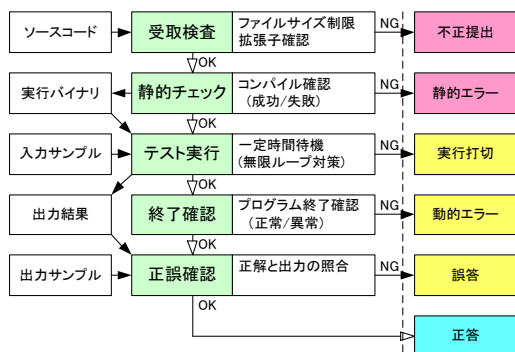


図8 実行テストの処理手順と判定結果

予備テスト進捗											
問題	1	2	3	1	2	3	1	2	3	1	2
予備	○	○	○	○	○	○	○	○	○	○	○
最終	100	60	100	60	70	100	0	100	0	100	1
最終テスト進捗											
問題	1	2	3	1	2	3	1	2	3	1	2
予備	○	○	○	○	○	○	○	○	○	○	○
最終	100	60	100	60	70	100	0	100	0	100	1
総合得点											
総得点	590										
順位	1										

図9 問題閲覧ページの出題解答モード

テスト	区分	得点	概要	入力	出力
予備	入力限定	40 %	計算結果が整数値になる場合のみに対応させる。	10.0 5.0	25.000000
最終	100 %	問題文の通り			
照会結果					
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10 問題閲覧ページの採点結果モード

3. 教師支援としてのモニタリング機能

3.1 教師支援のモニタリング機能とチュータリング機能

本システムは、2009 年度後期から本格的に運用している。当初のシステムでは、教師側への直接的な支援が不十分であり、GUI も貧弱な点があった。そこで、現在、教師側のサブシステムを大幅に改良し、モニタリング機能とチュータリング機能を充実させることを目指している[8]。

以前のシステムでは、得点状況や提出履歴など、素データに近い情報を迅速に表示していたが、必要な情報のみを集約する機能に欠けていた。モニタリング機能は、各学生の問題への着手状況を一覧するだけでなく、教師側が注意を払うべき部分を発見しやすくするものである。チュータリング機能は、モニタリングされた情報に基づき、学生側のミスや躓きを判断して、教師側が個別に指導することを支援したり、半自動化するものである。

現時点では、モニタリング機能の実装を進めている。例えば、提出が滞っている学生を自動通知する。ポップアップや色分けなどで、教師側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、コメント付加などで、教師側の目視によるコード確認も支援する。

3.2 モニタリング機能の概要

モニタリング機能は、図 11 のように整理される。監視対象として、学生個人、教室全体、出題設問に大別する。それぞれに対し、学生単位または解答単位での集計や集約を行い、その情報を連携して、必要なモニタリング情報を提供する(図 12)。具体的には、学生全体の進捗状況を一目で把握できるようにし、適切な助言や指導が行えるようにする得点順位ページ、提出された解答コードを閲覧し、学生へ適切な指導が行えるようにする提出履歴ページ、各学生の提出状況を把握し、提出していない学生や問題に詰まっている学生を発見する全学生と学生別の提出状況ページ、各問題の着手人数を把握し、問題の難易度の適切さを判断する学生単位の着手集計ページ、各問題の提出数を把握し、どの問題が解かれているか問題が適切か判断する解答単位

の着手集計ページが挙げられる(図 13).

3.3 モニタリング機能の一覧表示ページの機能

得点順位ページは、参加学生の総得点による順位表示を行う。各問題の実行テストごとの正誤判定の結果と得点も表示する(図 13a)。コンテスト中に、学生全体の進捗状況を把握するために用いる。現時点では、学生側のページとほぼ同じである。ただし、学生や答案に対して、他の教師側ページへのリンクを張っている。

提出履歴ページは、提出された解答コードを時系列で表示する。6 段階の判定結果を示し、解答コードと実行結果へのリンクを張る(図 13b)。コンテスト中に、学生が提出した解答コードを閲覧するために用いる。各項目の指定を行うことで、絞り込み表示が行える。

3.4 モニタリング機能の詳細分析ページの機能

全学生の提出状況ページでは、全学生の実行テストごとの提出回数を一覧する(図 13c)。学生を選ぶと、学生別の提出状況ページに移動する(図 13d)。いつどの実行テストに取り組んだかが時系列で表示される。コンテストの最中と事後に、特定の学生の振舞いを把握するために用いる。

学生単位の着手集計ページでは、実行テストごとの着手人数と着手率、および正答者数と正答者率を表示する(図 13e)。下限未満の正答率の箇所に色付けを行う。コンテスト中に、未着手のままの問題を発見するために用いる。このような問題に対して、ヒントなどを全体に通知する。

解答単位の着手集計ページでは、実行テストごとの提出件数と提出比率、正答件数と正答率を表示する(図 13f)。下限未満の正答率の箇所に色付けを行う。提出件数が多く、正答率が低い実行テストは、適切でない出題である可能性がある。必要があれば、訂正や補足、採点時の配慮を全体に通知する。また、問題ごとに、各実行テストの時間軸に対する提出件数のグラフを表示する。

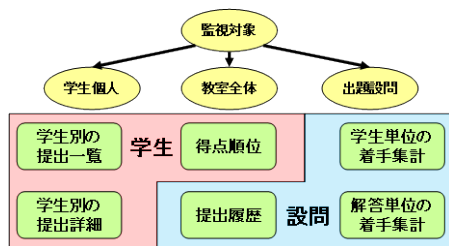


図 11 モニタリング機能の概要

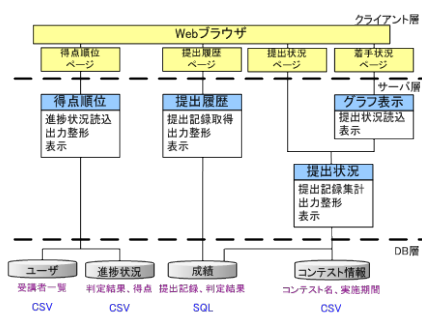


図 12 モニタリング機能のシステム構成

Lesson : RiseProgl 2010 Contest : W13_H01 更新

	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08	順位	得点
	予1	予2	予3	予1	予2	予3	予1	予2	予3	
s10c263	最終	最終	最終	最終	最終	最終	最終	最終	最終	1
	180	165	180	180	180	220	220	220	220	1545
s08c207	最終	最終	最終	最終	最終	最終	最終	最終	最終	2
	180	180	175	170	170	215	220	220	220	1530
s10c231	最終	最終	最終	最終	最終	最終	最終	最終	最終	3
	180	160	180	180	170	220	220	220	220	1530
s08c234	最終	最終	最終	最終	最終	最終	最終	最終	最終	4
	170									1490

(a) 得点順位ページ

コンテスト 問題 テスト 結果 学籍番号 表示数

全部 全部 全部 全部 最新100 更新

提出	日時	時差	コンテスト	ユーザ	問題	テスト	判定	解答	実行	割当
36881	2011-01-27 20:51:35	---	W13_H01	s10t232	Q06	最終	誤答	解答	結果	提出
36880	2011-01-27 20:49:57	00:01:38	W13_H01	s10t226	Q01	予備02	正答	解答	結果	提出
36879	2011-01-27 20:48:11	00:01:46	W13_H01	s10t232	Q06	最終	誤答	解答	結果	提出
36878	2011-01-27 20:47:13	00:00:58	W13_H01	s10t232	Q06	予備03	正答	解答	結果	提出
36877	2011-01-2	---	---	---	---	---	---	解答	結果	提出
36876	2011-01-2	---	---	---	---	---	---	解答	結果	提出

(b) 提出履歴ページ

	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08	得点
	予1予2予3	予1予2予3	予1予2予3	予1予2予3	予1予2予3	予1予2予3	予1予2予3	予1予2予3	最終提出
s06c201	1	---	---	---	---	---	---	---	170
	01:27 16:41	---	---	---	---	---	---	---	01:27 16:41
s06c244	29 7	3 1	1 1	1 1	2 1	1 1	1 1	1 1	1435
	01:31 19:37	01:27 17:17	01:27 04:26	01:27 17:25	01:27 03:55	01:27 03:40	01:27 03:26	01:27 03:11	01:31 19:37
s07c201	2 1	1	---	---	---	---	---	---	170
	01:27 00:16	---	---	---	---	---	---	---	01:27 00:16
s07c245	---	---	---	---	---	---	---	---	0
	---	---	---	---	---	---	---	---	---

(c) 全学生の提出状況ページ

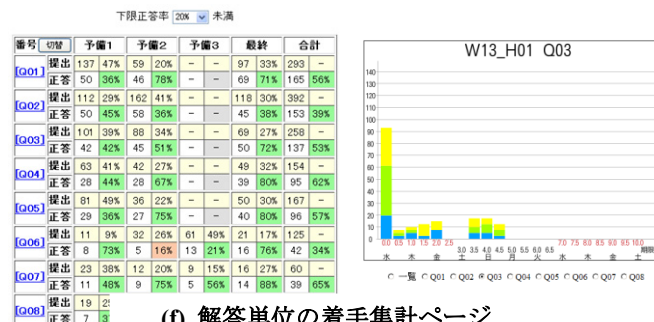
経過時間	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08
経過時間	予1予2予3	予1予2予3	予1予2予3	予1予2予3	予1予2予3	予1予2予3	予1予2予3	予1予2予3
0 12	1	---	1	1	1	1	1	1
1	1	---	1	1	1	1	1	1
2	---	---	---	---	---	---	---	---
3	---	---	---	---	---	---	---	---
4	---	---	---	---	---	---	---	---
5	---	---	---	---	---	---	---	---
6	---	---	---	---	---	---	---	---
7	---	---	---	---	---	---	---	---
8	---	---	---	---	---	---	---	---
9	---	---	---	---	---	---	---	---
10	---	---	---	---	---	---	---	---
11	---	---	---	---	---	---	---	---
12-24	---	---	---	---	---	---	---	---
24-36	---	---	---	---	---	---	---	---

(d) 学生別の提出状況ページ

受講者人数 93人
下限正答率 70% 未満

番号	予備1	予備2	予備3	最終
Q01	29 31%	23 25%	---	36 39%
Q02	24 83%	22 96%	---	34 94%
Q03	19 20%	22 24%	---	23 25%
Q04	18 95%	17 77%	---	17 74%
Q05	15 16%	16 17%	---	17 18%
Q06	8 8%	8 9%	---	11 12%
Q07	8 10%	8 9%	---	10 11%
Q08	7 78%	7 88%	---	9 90%
Q09	4 4%	3 3%	6 6%	6 6%
Q10	4 100%	3 100%	5 83%	6 100%
Q11	3 3%	3 3%	2 2%	5 5%
Q12	3 100%	3 100%	2 100%	5 100%
Q13	3 3%	3 3%	2 2%	5 5%

(e) 学生単位の着手集計ページ



(f) 解答単位の着手集計ページ

図 13 教師側機能の GUI

4. 運用実験の結果分析

4.1 2010 年度後期「プログラミング I」での運用実験

システムの評価として、情報系学科 1 年次後期の初級 C 演習で、運用実践した。コンテストは、76 名程度の受講者に対し、宿題型として実施した。出題範囲は、変数とデータ型、演算と式、選択構文と反復構文、配列、関数である。コンテスト期間を 1 週間とし、8 問(基本 5、応用 3)を出題した。基本問題は、配点 150 点、予備テスト 2 個(50/100 点)で、所要時間 60 分程度である。応用問題は、配点 200 点、予備テスト 3 個(50/100/150 点)で、所要時間 90 分程度である。合格基準を 750 点とし、完答 5 問よりも半答 8 問を目指すように指導した。時間調整点については、図 7 による説明だけでなく、表 1 を明記して周知した。

4.2 提出状況グラフと積算得点グラフ

コンテスト中の学生の振舞いについて、時系列での提出状況と個人ごとの積算得点に着目し分析を行う。提出状況のグラフは、横軸をコンテスト開始からの時間とし、適当な時間幅で分ける(図 14)。縦軸をその区間ごとの提出件数の棒グラフとし、予備と最終の内訳も示す。折線は、予備および最終の正答率である。積算得点グラフは、横軸をコンテスト開始からの時間とし、個人ごとの積算得点を折線で示す(図 15)。

4.3 全体での提出履歴の時系列分析

実験結果として、解答コードの提出履歴を時系列グラフで示した。初日の提出が 1500 件中の 350 件で、全体の 1/5 以上を占めている。早期提出による加点を狙って、すぐに解答に着手している。早期と通常で、期間の終了前 12 時間に、提出の山がある。時間調整点の導入により、以前より、前倒しで解答に取り組む傾向が出ている[9]。通常期間の土～日は正答率が 0.7 以上で、各締切前や遅延提出の 0.4 程度より高い。この期間は、確実な正答を目指している。一方、早期と通常の期間の終了前は、正答率が 0.4 程度で、各期間中の 0.5～0.8 より低い。締切を意識し、慌てて正誤を確認している傾向がみられる。遅延期間の締切前の正答率は 0.6 で、それ以前の締切前の 0.4 程度より高い。短期間での連続した提出などから、他人の解答コードを丸移した可能性がある。実際に、そのようなケースも見つかった。また、何度も誤答を繰り返している問題に対して、該当する解答コードを集中して確認し、頻出する間違いを指摘できた。例えば、変数の初期化をせずに参照するなど、コンテスト中にヒントを追加した場合もある。

4.4 全体での積算得点の時系列分析

演習中の学生の振舞いを分析するため、全学生の得点変化を積算得点グラフで示した。配点での満点が 1350 点、時間調整点を含めた満点が 1590 点である。最高点は、1545 点で、最低点は、0 点だった。最終的に着手した学生は、72 名で全体の 95%であった。初日から着手して得点を得て

いる学生は 33 名(43%)で、加点期間終了までの着手では 47 名(61%)であった。減点期間が始まるまでには、59 名で 78%の学生が着手している。また、早期、通常、延長期間の締切直前に得点を伸ばしている学生も確認できた。これらから、提出状況グラフでも読み取れた、時間調整点による前倒しで解答に取り組む傾向と、締切を意識した取り組みが確認できた。

4.5 成績別での提出履歴の時系列分析

さらに、解答コードの提出履歴を表 2 の成績別での 4 群に分けた(図 16)。第 1 群(15 名)は、提出数が最も多く約 530 件である。全体の提出割合は、加点期間 70%、通常期間 25%、減点期間 5%だった。時間調整点による加点への意識が高い。実行テストの利用では、予備テストをバランスよく使い、最終で誤答しないようにしている。第 2 群(21 名)は、提出数が第 1 群に次いで多く約 450 件である。全体の提出割合は、加点期間 60%、通常期間 20%、減点期間 10%だった。時間調整点の加点と減点を意識している。実行テストの利用では、最終の提出数が多く、誤答率も高くなっている。最終への慎重な解答を促す必要がある。第 3 群(21 名)は、提出数が第 2 群より少なく約 380 件である。全体の提出割合は、加点期間 40%、通常期間 30%、減点期間 30%だった。加点期間がやや高いが、時間調整点への意識はあまり高くない。実行テストの利用では、予備 1、2 をメインに提出しており、最終の正答率も高くなっている。これは、提出件数が上位群と比べ少ないことから、難しい問題を避け、確実に解ける予備テストのみを解いている。問題で行き詰っている学生への助言が必要である。第 4 群(19 名)は、提出数が最も少なく約 160 件である。問題には着手しているが、解決への糸口が掴めていない。

4.6 成績別での積算得点の時系列分析

積算得点についても成績別の 4 群に分けて分析を行った。積算得点の推移から、演習の取り組み方についていくつかのパターンが見られた。第 1 群では、3 つのパターンが確認できた。1 つ目は、加点期間の間に全ての問題を解き、早期に得点を大きく伸ばすものである。2 つ目は、加点期間の間に着手し継続的に問題を解き、徐々に得点を伸ばすものである。3 つ目は、減点開始直前に一度に問題を解き得点を伸ばすものである。第 2 群では、第 1 群と同様に継続的に問題を解いていく取り組み方が見られるが、目標得点を超えるとその時点で解答をやめるパターンが多く確認できる。これは、今回のコンテストが自由参加型のものであったためと考えられる。第 3 群では、2 つのパターンが確認できた。1 つ目は、上位群と同様に継続的に得点を伸ばすものであるが、得点の伸びが細かく目標点には到達できていない。これは、提出履歴の分析結果にもあるが、確実に解ける予備テストのみを解いているためだと考えられる。2 つ目は、加点終了、減点開始、提出締切の直前に得点を伸ばすものである。時間調整点の導入以前は、提出締

切の直前に提出が集中していたため、時間調整点による加点と減点により、提出行為が前倒しされたと考えられる。第4群では、得点の変化が見られず、提出締切直前に得点を伸ばすパターンが確認できた。得点の変化は少ないが、提出履歴では提出が行われているため、誤答を提出していると考えられる。予備テストも正答できていないため、問題の解決への糸口が掴めていないと考えられる。

5. 振舞いパターンと進捗推移グラフ

5.1 振舞いパターンと活性度

コンテスト中の学生の振舞いは、理解度と積極性の観点により4つのパターンに分類できると考えられる(図17)。パターン1は、授業内容を理解し、コンテストへの参加も積極的な学生である。そのため、開始時から積極的に提出を行い時間調整点での加点を獲得でき、理解度も高いため多くの問題を正答する。パターン2は、授業を理解しているが、コンテストへの参加が消極的な学生である。開始時はあまり提出せず、締切直前に提出するので時間調整点で減点されるが、理解度は高いため多くの問題を正答し、締切直前で得点が伸びる。パターン3は、授業の理解は不足しているが、コンテストには積極的な学生である。開始時から提出を行い、時間調整点での加点をある程度得られるが、簡単な問題しか正答できない。パターン4は、授業の理解が不足し、コンテストにも消極的な学生である。全般的にあまり提出を行わず締切直前になって取り組むが、簡単な問題しか正答できない。

振舞いパターンの分析には、積算得点グラフが適している。しかし、得点を軸としているため総得点の異なるコンテスト間で扱うのが難しい。そこで、全てのコンテストで共通して扱えるように、積算得点グラフの面積の大きさを指標とした、活性度を定義した[10]。共通して扱うため、各学生の積算得点グラフの面積をグラフの総面積(経過時間×コンテストの設問の総得点)で割り、割合で表す。成績に関わるため学生の振舞いが顕著になる試験型のコンテストについて、時系列での4群ごとの活性度の変化をグラフにした(図18)。グラフから、想定していた学生の振舞いパターンを確認することができた[11][12]。

5.2 学生個人の進捗推移グラフ

得点推移では、大まかな進捗状況は確認できるが、個人の振舞いを細かく把握することができない。例えば、得点変化がない学生が、問題がわからず提出できない、さぼって提出していない、提出しているが誤答となっている、などの詳細な状況の把握が難しい。そこで、得点推移グラフに学生の解答の提出時刻、問題番号、実行テスト種類、正誤判定の結果、などの提出ログ情報を合わせて表示する、学生個人の進捗推移グラフを提案し実装した。実装に当たり、既存の学生個人の提出状況ページに、進捗推移グラフ

を組込むことにした。提出状況ページでは、これまで、解答の提出ログを時系列の表形式で表示していた(図13d)。しかし、表形式では縦に長くなり、提出ログ情報の確認が手間だった。得点推移グラフに提出ログ情報を合わせることで、いつ、どのような提出が行われたかを即時的に確認できるようにする。

改良後の学生別の提出状況ページでは、指定されたコンテストと学生の進捗推移グラフと提出履歴の表示を行う(図19)。進捗推移グラフは、時系列で積算得点の推移の折線グラフと、解答コードの提出履歴の散布図を合成して表示する。グラフは、全問題の提出の合計と、問題別に集計したものを表示する。散布図のマーカーの形と色により、提出された実行テストの種類と正誤判定の結果を表す。マーカーは、予備テスト系列で三角形、最終テストで菱形、正答で青、誤答で赤としている。マーカーにカーソルを合わせると、詳細な提出ログ情報も参照できる。目安として、コンテストの合格点と加点期間の終了時刻に補助線を入れている。問題の取り組みの順序や、現在どの問題に着手しているか、最後の提出から時間が経っていないか、などを素早く確認できる。コンテストの最中と事後に、特定の学生の振舞いを把握するために用いる。

表1 運用実験の宿題型コンテストの得点ルール

得点	水	木	金	土	日	月	火	水	締切後
難易度B	180	170	160	150	150	150	145	140	15
難易度C	230	220	210	200	200	200	195	190	20

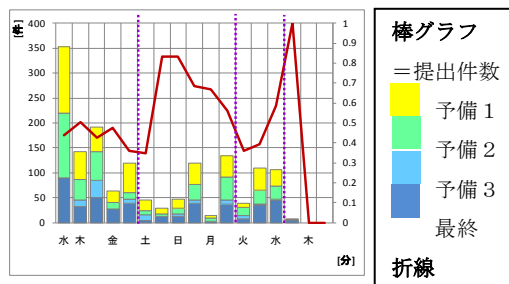


図14 答案コードの提出履歴の時系列分析

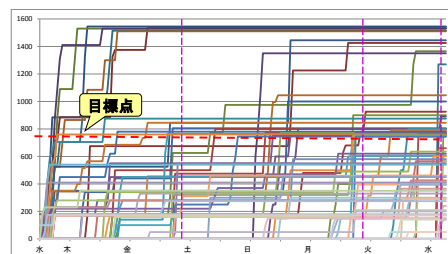


図15 個人ごとの積算得点グラフ

表2 成績による4段階の学生の分類

集団	順位	得点	人数
第1群	最上位	900-1600	15
第2群	中上位	750- 899	21
第3群	中下位	350- 749	21
第4群	最下位	000- 349	19

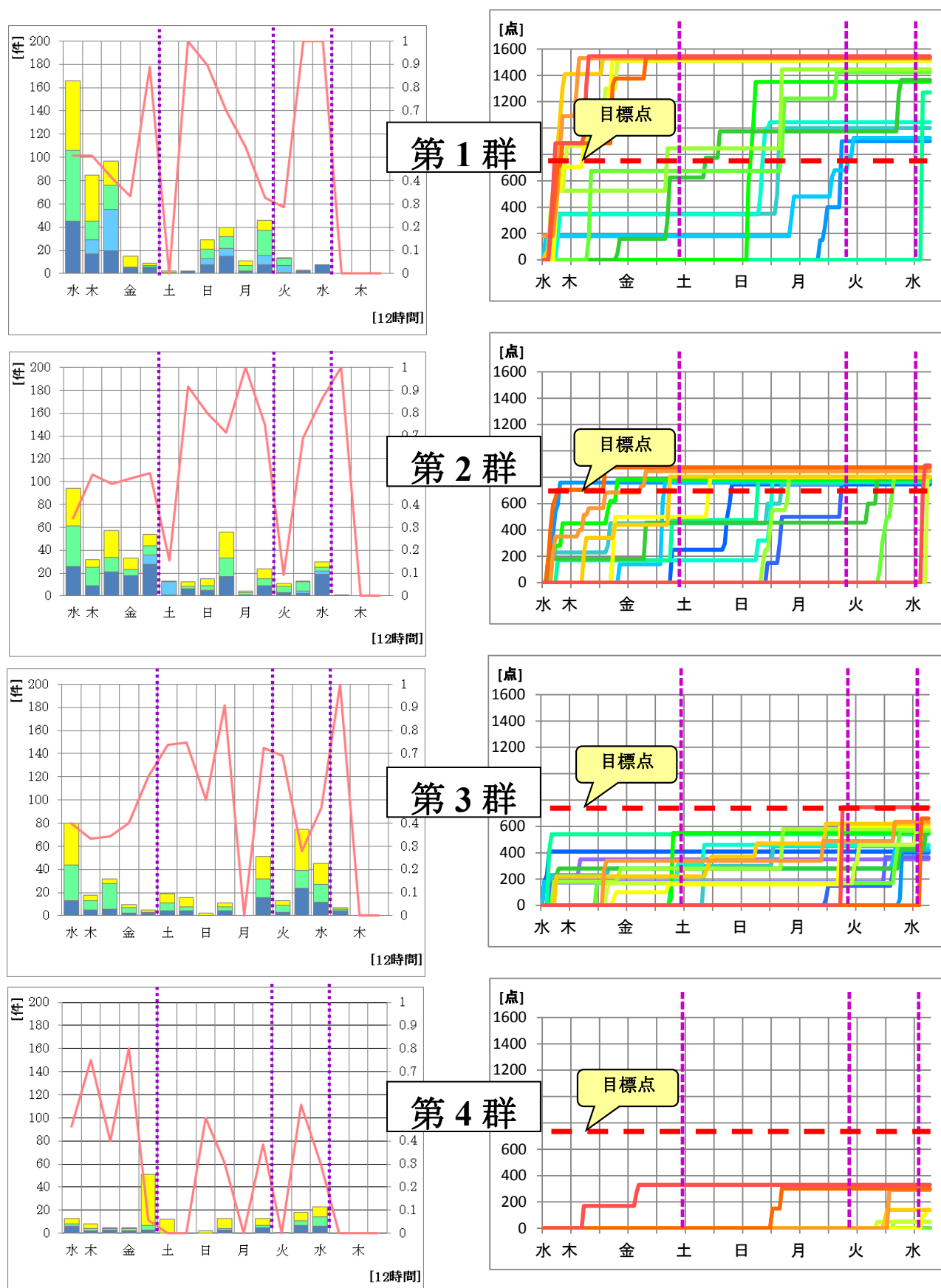


図 16 時系列による提出状況グラフと積算得点グラフ

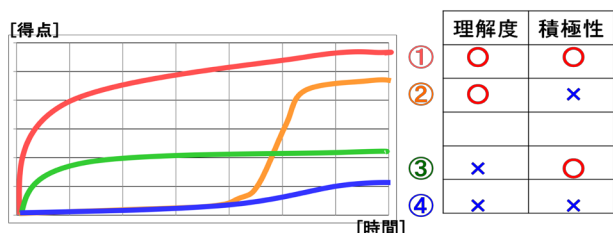


図 17 学生の 4 つの振舞パターン

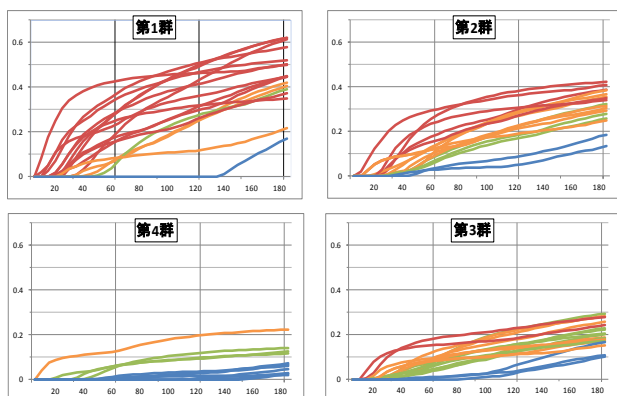


図 18 時系列での 4 群ごとの活性化度グラフ

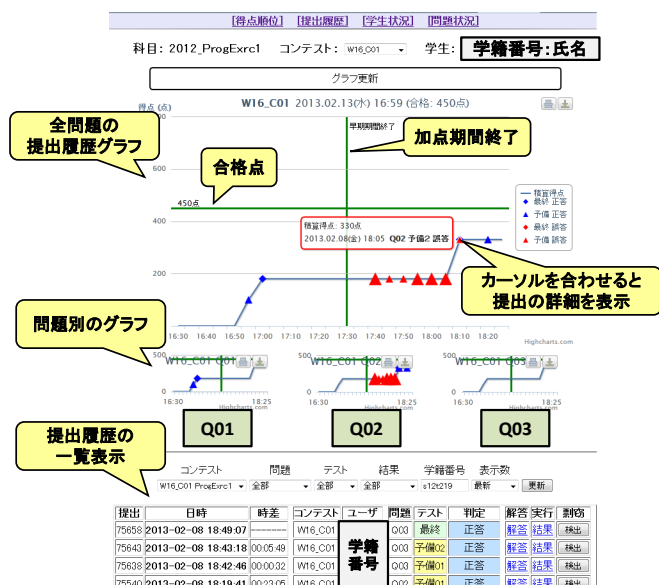


図 19 改良後の学生別の提出状況ページ

6. おわりに

大学情報系の入門 C 授業において、初心者向け的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を開発している。実行テスト系列と時間調整点を含む得点ルールを採用し、演習への積極的な参加を促す。教育実践として、宿題型のコンテストを実施した。実践結果の分析から、実行テスト系列による部分解答の頻繁な提出、時間調整点による提出行為の前倒しがみられ、演習の

活性化という教育効果が確認できた。また、教師側への間接的な効果としては、必要に応じて解答の目視確認を詳細に行ったり、補助学生と連携して発見した躓きに迅速に対応するなど、時間的余裕を振り分けることができた。即時的に学生の振舞いをグラフ化する機能として、進捗推移グラフを提案し、実装した。今後は、進捗推移グラフを実際の授業実践で活用し、演習中の学生の振舞いを詳細に分析する。

参考文献

- 1) 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", 情処研報, Vol.2006, No.108, pp.9-16, (2006).
- 2) 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", 情処研報, Vol.2007, No.101, pp.11-18, (2007).
- 3) 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", 情処研報, Vol.2008, No.42, pp. 49-56, (2008).
- 4) 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 -", 情処研報, Vol.2010-CE-104, No.4, pp.1-12, (2010).
- 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 - 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 -", 情処研報, Vol.2010-CE-103, No.10, pp.1-8, (2010).
- 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 - 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).
- 7) 西村智治, 川崎慎一郎, 富永浩之, "競争型学習を取り入れた初級 C プログラミング演習における教師支援 - モニタリング機能としての解答状況と提出状況の視覚化 -", 信学技報, Vol.110, No.453, pp.163-168, (2011).
- 8) 西村智治, 川崎慎一郎, 富永浩之, "小コンテスト形式の初級 C 演習における教師支援 - 解答状況の時系列表示によるモニタリング機能の試用実験 -", 情処研報, Vol.2011-CE-110, pp.1-8, (2011).
- 9) 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 - 授業実践における解答コードの提出履歴の詳細分析 -", 情処研報, Vol.2010-CE-107, No.10, pp.1-10, (2010).
- 10) 川崎慎一郎, 西村智治, 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 - 解答状況による学生の振舞いパターンの分析 -", 信学技報, Vol.111, No.141, pp.53-58, (2011).
- 11) 西村智治, 蔵本孝司, 富永浩之, "競争型学習を取り入れた初級 C プログラミング演習におけるモニタリング機能の開発", JSiSE 全国大会, Vol.37, pp.180-181, (2012).
- 12) 富永浩之, 西村智治, "実行テスト系列を取り入れた小コンテスト形式の初級 C 演習 - 学生の得点状況の時系列分析による活性化の推定 -", 情処研報, Vol.2012-CE-116, No.15, pp.1-8, (2012).