

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 -

富永 浩之[†] 川崎 慎一郎[†]

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 演習の授業で、模擬試験として運用実験を行い、提出履歴の分析とアンケート評価を行った。その結果、上位群への意欲の促進、下位群への着手の支援などの効果が確認できた。

Support for Introductory C Programming Exercise with Contest Style as Competitive Learning - Log analysis of execution test series in educational practices -

Hiroyuki Tominaga[†] Shinichiro Kawasaki^{††}

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. It raises students' motivation and activity by competitive learning approach. We have also 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. They give clues and guidelines for solving the final goal. We carried out some educational practices as experiments in introductory C programming lesson. We analyzed submission logs and questionnaires. By the result, we verified the educational effect.

1. はじめに

大学の情報系学科の多くは、初年次に C 言語の入門的な科目を必修とし、演習を重視している。授業の前半で文法や例題を解説し、後半に類題を解く時間を設ける形態が多い。しかし、学生の理解度や受講態度に差があり、全ての学生を集中させて演習に取り組ませることは難しい。例えば、プログラミングの得意な学生が退屈したり、分かったつもりで後回しにしたり、苦手な学生が途中で解答を諦めたりしてしまう。そこで、プログラミング演習にも、競争型学習(Competitive Learning)を導入することが注目されている[1]。特に、教師からの出題に対して解答し、後から採点結果を知らされるテスト形式に代わって、進捗状況が即時的に公開され、学生同士の対戦を意識させるコンテスト形式が有望である。ゲーム感覚のイベントとして、勝つという目的意識や、学生間の競争意識が、学習意欲を高めることが期待される。ただし、一般的なコンテストの方法では、初心者への敷居が高く、多くの学生が達成感を得られないままとなりがちである。また、入門科目において、授業時間内に組み込みにくい。

本研究では、初級 C 授業と初心者に適応した、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[2][3][4]。大会運営サーバ tProgrEss を開発し、実際の授業で運用している[5][6]。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。

小コンテスト形式の演習の進行は、図 1 の通りである。授業内容に関連する複数の問題を与え、何問かを選択し解答する。問題文に沿ったプログラムをローカル PC で作成し、大会運営サーバにアップロードして正誤の自動判定を行う(図 2)。制限時間内に早く多く解いた方が高得点となる。進捗状況は Web 上に公開され、学生が相互に確認できる。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、全体の順位表示や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。

このような演習により、学生は自分のペースで解答を進められる。さらに、学生の提出状況を閲覧する教師監視ページも用意する。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。これにより、解答が進まない学生を早期に発見し、個別に指導する。

[†] 香川大学工学部
Faculty of Engineering, Kagawa University

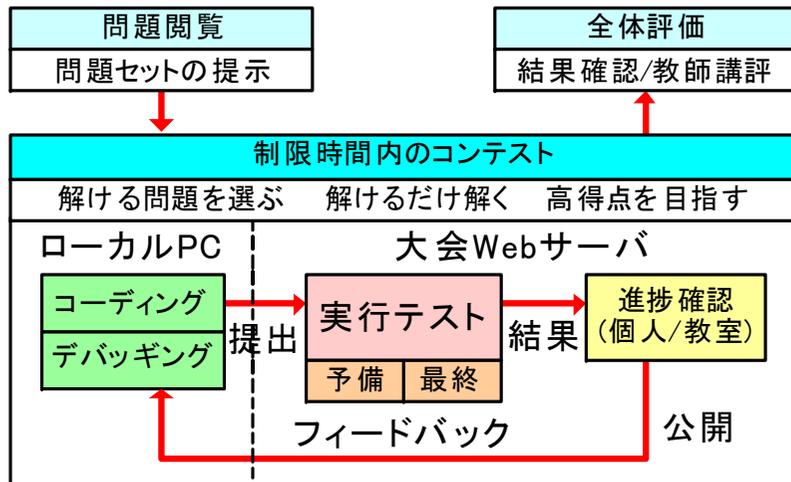


図 1 小コンテスト形式の演習の進行

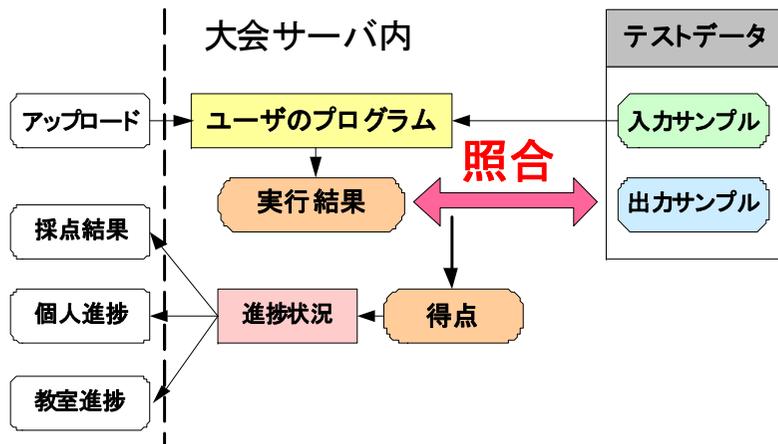


図 2 プログラムの正誤判定の手順

2. プログラムの正誤判定

本研究では、演習における判定結果の即時通知を重視するため、事前に用意された入出力サンプルによる実行テストを用いて、プログラムの正誤判定を行う。これをサーバ側に提出されたソースコードに対して行う。まず、アップロードされたソースコードを、システムがコンパイルする。生成された実行バイナリに対し、入力サンプルを与えて実行する。その実行結果と、正解となる出力サンプルとで照合を行う。出力サンプルは、問題文の仕様を満たした模範プログラムによる実行結果である。

照合に成功し、正答と判定されれば、規定のルールに従って計算された得点を与える。ただし、実行結果のみに基づいて照合を行うため、ソースコードの内部については問わない。必要があれば教師が後から自分で内容を吟味する。また、不正コピーなどを防ぐため、ソースコードの類似度を判定するモジュールを組み込む。ソースコード提出の判定結果は、図3の6段階で行う。

- (1) **不正提出** 提出したコードのファイルサイズや拡張子に不備があることを示す。誤って実行バイナリを提出した場合などである。また、プロセス制御文のような、字面で分かる危険コードを排除する。既に提出されたコードと比較し、類似度の高いコードを剽窃と判定する。
- (2) **静的エラー** コンパイルに失敗し、実行バイナリが生成されなかったことを示す。タイプミスや文法的な間違いである。
- (3) **実行時エラー** プログラムの実行がエラーとなり、異常終了したことを示す。主に、配列の範囲外参照やポインタ関連のミスである。
- (4) **実行打切** 指定時間内にプログラムの実行が終了しなかった、または、出力結果が指定サイズを超えたことを示す。無限ループが発生して暴走している、仕様のないデータを要求して入力待ちの状態になっている、と推測される。時間やサイズは、出題内容に応じて、適切な値を設ける。(3)と(4)は、判定の手順としては逆であるが、(4)の方がより正答に近い可能性があるため、判定結果としては、この順としている。
- (5) **誤答** 実行は正常に終了したが、実行結果が出力サンプルとの照合に失敗したことを示す。アルゴリズムや出力書式のミスである。
- (6) **正答** 正しく実行が終了し、実行結果が出力サンプルとの照合に成功したことを示す。

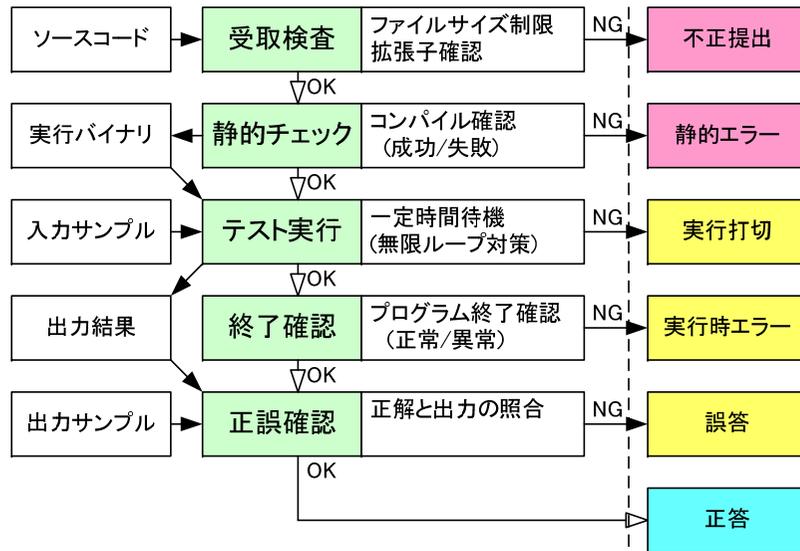


図 3 正誤判定の段階と手順

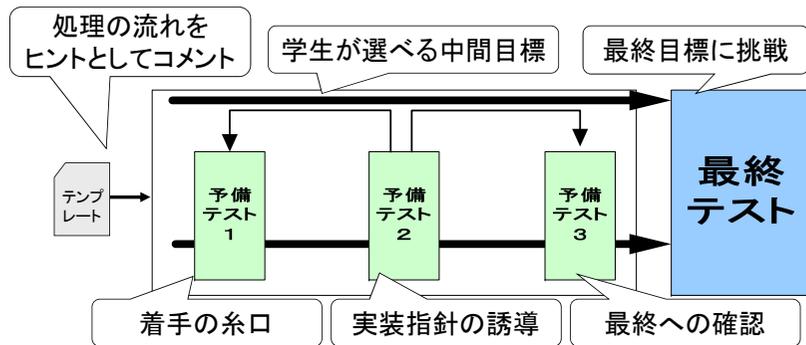


図 4 実行テスト系列の予備テストと最終テスト

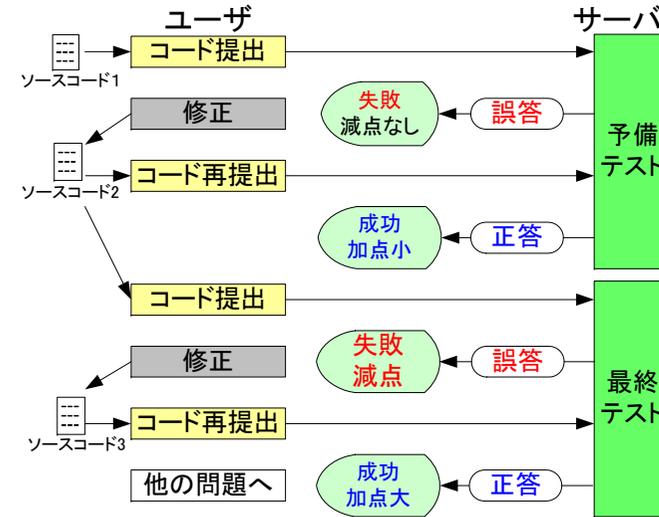


図 5 予備テストと最終テストによる判定の相違

| | | |
|--|----------|--------|
| 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

| | |
|----------------------------------|-------------------------------------|
| 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

3. 実行テスト系列

3.1 予備テストと最終テスト

初心者の演習では、必ずしも全員が問題を完答できるとは限らない。そこで、1つの問題につき、複数の予備テストおよび最終テストからなる実行テスト系列を提示する(図4)。最終テストは、題意を完全に満たす解答を要求するが、予備テストでは部分的な仕様を満たせばよい(図5)。図6は、不定個の整数値入力を受け取り、平均値を出力する問題である。この問題に対し、番兵による入力処理の確認、平均値計算の確認、出力書式の確認、などの予備テストを提示する。

予備テストでは、最終テストへ至る前の段階的な確認を行う。予備テストの正解には、部分的な得点を与える。何度でも挑戦できるよう、失敗でも減点しない。入出力のサンプルは公開されており、エラー情報も提示するため、積極的にテストに取り組み、その結果を元にデバッグを進められる。

最終テストは、十分な量の入出力サンプルを用いるが、学生には提示されず、結果のみを通知するブラックボックスとして行う。正解には大きな得点を与えるが、失敗すれば減点となる。一般的なコンテストでは、この最終テストに該当するテストが用意されている。最終テストで入出力サンプルを提示しない理由として、具体的な処理を実装せず、示したサンプルにのみ対応したようなコードを防ぐためである。また、

予備テストを利用して慎重にテストを繰り返すことを促す効果も狙っている。

学生は、自身の理解度に応じ、最終テストの正答を目指して自由に実行テストを利用できる。プログラミングが不得意な学生には、完全な解答への着手の糸口となる。中間目標として、段階的な実装を誘導する。最終テストへの正答が難しいと判断した学生も、部分点を狙って、諦めずに挑める。プログラミングが得意な学生は、一気に最終テストの正答を狙ってもよい。ただし、最終テストに減点を設けることで、最後の確認として予備テストも利用させ、慎重さを促す。

3.2 包含系と加算系の実行テスト系列

中間目標としての予備テストは、部分仕様の与え方と得点ルールで包含系と加算系の2種類を用意する。これらの実行テスト系列を学習項目や教育目的に応じて使い分け、学生のプログラム作成を適切に誘導する。包含系は、問題の仕様を緩めて、徐々に完成に近づけさせる。図6中の入出力サンプルのように、パスすべきデータの難易度を段階的に高めていく。副次的な処理を除々に追加し、完成度を高めていくような問題に適している。学生に段階的なプログラム作成を意識させることができる。本研究でのコンテストでは、包含系を中心に予備テストの作成を行う。加算系は、問題の仕様を分割して、別々の小さな問題として実装させ、最後に統合させる。機能ごとに部分的な動作を別の入出力サンプルを通して確認する。関数ごとの実装を確認するような実行テスト系列では、予備テストは単体テスト、最終テストは結合テストに相当する。例えば、番兵式入力に関するコード、加算と除算に関するコードなどを別々に作成させ、最終的に1つのコードに必要な処理を統合する。これは、処理をモジュール単位に分割しやすい問題に適している。現時点では、実際の出題には、包含系のみを用いている。

包含系の予備テストは、図7のようなガイドラインに沿って構成し、問題内容に即した処理の確認を行う。予備テスト1では、入力値をデータとして正しく扱えるかを主な確認項目とする。まずは、問題に取り組むきっかけとなる部分として、確実に解ける内容で実行テストを行う。予備テスト2では、主要な反復処理や配列操作を確認項目とする。問題の中心となる処理において、一部の機能のみを確認したり、例外を含まない簡単な例で実行したりする。予備テスト3では、最終テストに至る最後の確認として、整形出力や例外処理の確認を行う。最終テストとほぼ同様の仕様とすることで、予備テスト3を最終テスト前の確認として利用することができる。

3.3 判定結果の即時通知

問題閲覧ページは、出題解答モードと採点結果モードで表示が切り替わる。図8(a)の出題解答モードでは、問題本文および実行テスト系列の仕様と入出力サンプルが表示される。解答プログラムのソースコードをアップロードすると、図8(b)の採点結果モードに切り替わる。採点結果モードにおいて、プログラムの正誤判定の結果が、学生に即時に通知される。この段階に応じ、コンパイルメッセージ、入出力サンプルと

実行結果、提出したコード自体も併せて提示される。判定が不正提出の場合、提出コードを表示し、学生に確認させる。静的エラーの場合は、コンパイルメッセージの確認や、ローカル PC での再コンパイルを促す。実行打切の場合は、変数などの初期化の確認などを促す。実行時エラーの場合は、そこまでの実行結果を表示する。誤答の場合は、学生は出力結果を確認し、どの処理が正しく動作していないのかを推測する必要がある。正答の場合は、解答が順調に進んでいるため、次の問題やテストに誘導する。

コンテスト中は、問題閲覧ページの上部に、各問題の実行テストの進捗状況、得点と順位が速報として表示される。また、教室全体の順位表示、個人単位の提出履歴のページも用意する。図 8(c)の順位表示ページでは、共時的な進捗確認として、教室全体の進捗状況を一覧する。他の学生の解答状況や得点を確認でき、全体の中での自分の位置付けを把握して、学生間の競争意欲を促進させる。図 8(d)の提出履歴ページでは、通時的な進捗確認として、学生個人の提出履歴やテスト結果を掲載する。過去に提出したソースコードの内容を確認し、自分の弱点を把握したり、今後のコーディングの参考にする。

| コンテスト | ユーザ | 問題 | 個人 | 教室 | 速報 | テンプレート | | | | |
|---------|-------|--|---------|-----|-----|-------------------------------|-----|---|---|-------|
| W14.C01 | guest | Q04 | 問題選択に戻る | 確認 | 確認 | 100点 66位 更新 詳細 Template.Q04.c | | | | |
| | | Q02 | Q03 | Q04 | Q05 | Q06 | | | | |
| | | 00 01 02 03 00 01 02 03 00 01 02 03 00 | | | | | | | | |
| | | 0 | 0 | 0 | 100 | 0 | | | | |
| 予備01 | 25点 | 100以上のデータ列はない場合とす。合格数のみ | 51 | 62 | 78 | 35 | 100 | 0 | 3 | 65.20 |
| 予備02 | 50点 | 100以上のデータ列はない場合とする。合格数と平均 | 51 | 62 | 78 | 35 | 100 | 0 | 3 | 65.20 |
| 予備03 | 75点 | 100以上のデータ列はない場合とする。合格数、平均 | 51 | 62 | 78 | 35 | 100 | 0 | 3 | 65.20 |
| 最終 | 100点 | 問題文の通り | | | | | | | | |

図 8(a) 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01 | 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 | | |

照合結果

```

コンパイルに成功しました。
実行は正常に終了しました。
出力結果は 正答 となりました。
次のテストに進みましょう
得られた得点は40点です。
    
```

コンパイル結果

```

[コンパイルエラー表示]
    
```

実行結果

| 入力サンプル | 出力サンプル | 実行結果 |
|----------|-----------|-----------|
| 10.0 5.0 | 25.000000 | 25.000000 |

提出コード

```

// .....
// 工学部2009年度「プログラミング1」(富永) 宿題 09-01
// .....
// 2009.10.26(月) 05:12:87 川崎慎一郎
// .....
    
```

図 8(b) 問題閲覧ページの採点結果モード

| | Q01 | Q02 | Q03 | Q04 | Q05 | Q06 | Q07 | Q08 | 総得点 | 順位 |
|----------|-----|-----|-----|-----|------|-----|------|------|-----|-----|
| 予備 | 予 | 予 | 予 | 予 | 予 | 予 | 予 | 予 | | |
| 最終 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | | |
| 得点 | 100 | 60 | 100 | 60 | miss | 70 | 100 | miss | 0 | 100 |
| guest | OK | OK | OK | OK | miss | OK | miss | OK | 590 | 1 |
| s051287 | OK | OK | OK | OK | miss | OK | miss | OK | 180 | 2 |
| s1051287 | OK | OK | OK | OK | miss | OK | miss | OK | 80 | 3 |

図 8(c) 順位表示ページ

| 学籍番号 | Q01 | | | Q02 | | | Q03 | | | Q04 | | | Q05 | | | Q06 | | | Q07 | | | Q08 | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| guest | 予01 | 予02 | 予03 | 最01 | 最02 | 最03 | 予01 | 予02 | 予03 | 最01 | 最02 | 最03 | 予01 | 予02 | 予03 | 最01 | 最02 | 最03 | 予01 | 予02 | 予03 | 最01 | 最02 | 最03 | 予01 | 予02 | 予03 | 最01 | 最02 | 最03 |
| | 正 | - | - | 正 | 正 | 正 | 正 | 正 | 正 | 正 | 誤 | - | 正 | 正 | 正 | 誤 | 正 | 正 | 正 | 正 | 正 | 誤 | 誤 | 誤 | 正 | 正 | 正 | 正 | 正 | 正 |

| | | | | | |
|----------|--|------|-----|------|----|
| 予備・最終の進捗 | | 問題番号 | テスト | 判定結果 | |
| | | 全部 | 全部 | 全部 | 更新 |

| | |
|-----------|--|
| 結果の再表示ボタン | |
|-----------|--|

| 提出日時 | コンテストID | ユーザID | 問題番号 | テスト | 判定結果 | 結果確認 |
|---------------------|---------|-------|------|------|------|------|
| 2009-11-06 00:40:51 | W04_H01 | guest | Q01 | 最終 | 正答 | 結果確認 |
| 2009-11-06 00:53:30 | W04_H01 | guest | Q01 | 予備01 | 正答 | 結果確認 |
| 2009-11-06 01:05:56 | W04_H01 | guest | Q02 | 予備01 | 正答 | 結果確認 |
| 2009-11-06 01:16:44 | W04_H01 | guest | Q02 | 予備02 | 正答 | 結果確認 |
| 2009-11-06 01:17:00 | W04_H01 | guest | Q02 | 最終 | 誤答 | 結果確認 |
| 2009-11-06 01:17:58 | W04_H01 | guest | Q02 | 最終 | 誤答 | 結果確認 |

図 8(d) 提出履歴ページ

4. 教育実践と実験評価

4.1 対象授業

tProgrEss を教育実践に利用し、運用評価を行った。対象科目は、学科 1 年次後期の初級 C 演習「プログラミング I」である。予備実験として、演習課題の正誤判定に利用し、パフォーマンスの向上を確認した。その後、3 回の宿題型コンテストとして運用した。学習範囲は、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。「プログラミング 1」は、一週間で水曜と金曜に 2 コマの授業があり、それぞれ授業と演習に分けて講義を行う。表 1 に、運用実験における対象授業のシラバスを示す。この内、予備実験を第 10～11 週、運用実験を、第 12～14 週で行った(表 2)。

4.2 予備実験と暫定評価

本研究の提案である、コンテスト形式のプログラミング演習、およびコンテスト運営支援システムの有効性について、授業での実践を通じて検証を行う。その準備段階として、事前に宿題型の予備実験を実施して、システムの自動採点の機能と GUI の簡単な評価を行う。

予備実験は、表 3 の日程で行った。分岐構文や反復構文の問題 8～10 問を出題し、1～2 週間の期間を与えた。各問題の配点は、難易度によって 60 点、100 点のものに分けた。また、2～3 段階の予備テストを設けた。目標点は、400～500 点とした。

予備実験では、前章で説明したように、サーバ過負荷の問題が発現している。特に、対象授業での 80 人規模が利用した初めての実験では、進捗状況の更新負荷のため Web ページが開かず、一時的にコンテストを中断した。その後の 4 回ほどの予備実験でも、負荷や出題ミスにより、コンテストに支障をきたす結果となった。失敗を重ねる中で、システム上の問題点を列挙し、運用実験へ向けた改良を進めた。なお、システムの不備が多かったため、本格的な授業運用を始めた年明けに再提出してもらった。

4.3 宿題型コンテストとしての毎週の運用実験の実施

宿題型コンテストは、表 4 の内容で実施した。配列や関数定義の問題 8～10 問を出題し、1～2 週間の期間を与えた。各問題の配点は、難易度によって 60 点、100 点のものに分けた。また、2～3 段階の予備テストを設けた。目標点は、400～500 点とした。

表 5 に予備実験のテスト種別の提出比率と正答率を示す。予備実験では、提出比率が予備 1 と最終に集中しており、予備 2 と予備 3 は全体的に低い。そこで、模擬試験では、予備テストの位置づけと、コンテストの効率的な進め方について事前説明を行った。具体的には、いきなり完答を目指して最終に提出せず予備で出力の確認を行うこと、最終が無理そうなら予備だけ解答して次の問題に移ることなどである。

4.4 教室型コンテストとしての模擬試験の実施

教室型コンテストとして、期末の模擬試験で運用実験を行った。受験者は、その時点での受講者はほぼ全員で 81 名である。制限時間は 120 分で 6 問を出題した。範囲は、数値計算、入出力書式、打ち反復、二重反復などである(表 6)。図 9 に出題例を挙げる。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 100 点とし、予備 1・2・3 を、それぞれ 25/50/75 点とした。合格基準としての目標得点を 300 点とし、完答 3 問よりも半答 6 問を目指すように指導した。

全員のテストごとの提出比率は、予備 1 と最終が 0.3 前後で、予備 2 と予備 3 は 0.2 程度で大きな開きがある(表 7)(表 8)。解答への糸口を利用している学生が多いことが分かる。上位グループはテストが進むにつれて正答率が下がっているが、最終テストでは上がっている。予備テストの解答誘導の効果がうかがえる。また、下位の全体の提出数のうち、約半数が予備 1 に集中している。予備 1 から着手しているが、そこからより難易度の高いテストへ進めていないことが分かる。

実験結果として、答案コードの提出履歴を分析した(図 10)。図 11 は、300 点で上位群 44 名と下位群 37 名に分け、答案コードの提出状況を時系列でグラフ化している。上位群は、50 分前後と 120 分にピークがあり、予備テストを上手く活用し、最後まで粘って複数の完答に達している。下位群は、80 分のみピークがあり、完答は 1 問程度で最後は部分点を狙ったものである。また、上位群は、予備 3 の提出が多く、完答を目指した慎重な確認を行っている。下位群は、提出の約半数が予備 1 であり、予備 2 以降には進めない場合でも、問題着手には至っている。

表 1 対象授業のシラバス

| 週 | 授業内容 | 理解目標 |
|------|---|--|
| 第1週 | 入門編(1) DOS環境、C言語の概要 言語処理系、コンパイルとエラー | Cプログラミングの意義と目的を認識する コンパイルする、エラーメッセージを読む |
| 第2週 | 入門編(2) 数値データ(int, double)、変数 四則演算、入出力(scanf, printf) | 変数を正しく使う、整数と実数の四則演算 実行結果を確認する、様々なエラーを体験する |
| 第3週 | 入門編(3) 選択構文(if)、数学関数(sqrt, fabs) 比較演算、演算子と式、代入文 | 複数のif構文による多分岐・重分岐を実装する データの大小比較、二次方程式の解法 |
| 第4週 | 入門編(4) 選択構文(switch)、数学関数 論理演算、定数マクロ(#define)、結合順位 | 数学関数を用いた図形量や関数値の計算 論理演算を用いて複雑な分岐条件を構成する |
| 第5週 | 入門編(5) 反復構文(while, for)、自己代入 反復子の初期化と更新、継続条件、変数遷移 | while構文による反復処理の変数遷移を把握する while構文とfor構文を相互に書き換える |
| 第6週 | 入門編(6) 多重反復、脱出(break, continue) 型変換、注釈、書法 | 変数表で二重反復の実行過程を追跡する break文とcontinue文による反復処理の流れ |
| 第7週 | 入門編(7) 様々な反復法、反復構文(do) 入力番兵、入力確認、出力整形、予約語 | 反復変数の複雑な更新による反復処理 不定個の入力、不正入力の排除、表形式の出力 |
| 第8週 | 初級編(1) 配列の宣言と初期化、添字と要素 配列の参照と代入、データ長による配列走査 | 配列の宣言、添字による要素の参照 データ長を与えた配列処理(走査・更新・格納) |
| 第9週 | 初級編(2) 多次元配列の構造、前処理 縦横集計、行列計算、前処理 | 配列の配列として二次元配列を理解する 二次元配列を関数の引数として利用する |
| 第10週 | 初級編(3) 関数の定義と呼出、関数原型宣言 変数と宣言、仮引数と実引数、返却値 | 関数を用いたプログラミング(定義と呼出) 処理の関数化と再利用を活用する |
| 第11週 | 初級編(4) 配列と関数、配列の応用処理 終端番兵、副作用、有効範囲 | 関数の引数として配列を活用する 不定個のデータ処理を配列で実現する |
| 第12週 | 初級編(5) 再帰的処理、漸化式と計算過程 変数と宣言、有効範囲、静的変数 | 漸化式の実装として再帰的な関数を作成する 反復法と再帰法の計算過程の相違を理解する |
| 第13週 | 初級編(6) プログラムの設計と実装 配列の応用、テスト、乱数 | 仕様と宣言の与えられた関数群を実装する プログラムに必要な関数を設計する |
| 第14週 | 初級編(7) プログラムの修正と改良 実行時間、デバッグ | プログラムの実行時間を計測する 実行途中の変数を出力してデバッグする |
| 第15週 | プログラミング実技の期末試験 | 総合的な理解度の判定 |

表 2 対象授業の日程

| 授業回 | 日付 | 形式 | 学習事項 |
|------|---------------|----|--|
| 第01回 | 2009.10.02(金) | 導入 | 授業の概要、受講の注意、プログラムとアルゴリズム |
| 第02回 | 2009.10.07(水) | 演習 | C言語の概要、DOS環境、言語処理系、コンパイルと実行 |
| 第03回 | 2009.10.09(金) | 講義 | C言語の概要、DOS環境、言語処理系、コンパイルと実行 |
| 第04回 | 2009.10.14(水) | 講義 | 変数と数値データ、四則演算、代入、一時変数、入出力、コメント |
| 第05回 | 2009.10.16(金) | 演習 | 変数と数値データ、四則演算、代入、一時変数、入出力、コメント |
| 第06回 | 2009.10.21(水) | 講義 | 入出力の書式、様々なエラー、数学関数、定数定義マクロ |
| 第07回 | 2009.10.23(金) | 演習 | 入出力の書式、様々なエラー、数学関数、定数定義マクロ |
| 第08回 | 2009.10.28(水) | 講義 | if構文、分岐条件、真偽値と論理演算、比較演算、フローチャート |
| 第09回 | 2009.11.06(金) | 演習 | if構文、分岐条件、真偽値と論理演算、比較演算、フローチャート |
| 第10回 | 2009.11.11(水) | 講義 | switch構文とラベル、break文、多分岐と重分岐、ブロックとインデント |
| 第11回 | 2009.11.13(金) | 演習 | switch構文とラベル、break文、多分岐と重分岐、ブロックとインデント |
| 第12回 | 2009.11.18(水) | 講義 | while構文、継続条件、反復変数の初期化と更新、結果変数 |
| 第13回 | 2009.11.20(金) | 演習 | while構文、継続条件、反復変数の初期化と更新、結果変数 |
| 第14回 | 2009.11.25(水) | 講義 | for構文、do構文、不定反復、基本制御構造、変数の値遷移表 |
| 第15回 | 2009.11.27(金) | 演習 | for構文、do構文、不定反復、基本制御構造、変数の値遷移表 |
| 第16回 | 2009.12.02(水) | 講義 | 多重反復、構造化フローチャート、表形式の出力、デバッグ出力 |
| 第17回 | 2009.12.04(金) | 演習 | 多重反復、構造化フローチャート、表形式の出力、デバッグ出力 |
| 第18回 | 2009.12.09(水) | 講義 | 無限反復、局所脱出、break文とcontinue文、入力番兵、フラグ変数 |
| 第19回 | 2009.12.11(金) | 演習 | 無限反復、局所脱出、break文とcontinue文、入力番兵、フラグ変数 |
| 第20回 | 2009.12.16(水) | 講義 | 配列の宣言と参照、配列要素と添字、宣言長と有効長、配列の基本操作 |
| 第21回 | 2009.12.18(金) | 演習 | 配列の宣言と参照、配列要素と添字、宣言長と有効長、配列の基本操作 |
| 第22回 | 2010.01.08(金) | 講義 | 不定長配列、終端番兵、基本データ型、型変換 |
| 第23回 | 2009.01.13(水) | 演習 | 不定長配列、終端番兵、基本データ型、型変換 |
| 第24回 | 2010.01.20(水) | 講義 | 多次元配列、表形式の入出力と計算 |
| 第25回 | 2010.01.22(金) | 演習 | 多次元配列、表形式の入出力と計算 |
| 第26回 | 2010.01.27(水) | 講義 | 関数の定義と呼出、仮引数と返却値、大域変数、局所変数の有効範囲 |
| 第27回 | 2010.01.29(金) | 演習 | 関数の定義と呼出、仮引数と返却値、大域変数、局所変数の有効範囲 |
| 第28回 | 2010.02.03(水) | 講義 | 配列と関数、関数の副作用、変数の定数化、演算子の結合順位 |
| 第29回 | 2010.02.05(金) | 試験 | 模試(0) 導入編 |
| 第30回 | 2010.02.10(水) | 試験 | 本試(1) 導入編 |
| 第31回 | 2010.02.12(金) | 試験 | 本試(2) 初級編 |
| 第32回 | 2010.02.17(水) | 試験 | 追試(1) 導入編 |
| 第33回 | 2010.02.19(金) | 試験 | 追試(2) 初級編 |
| 第32回 | 2010.02.24(水) | 試験 | 追試(1) 導入編 |
| 第33回 | 2010.02.26(金) | 試験 | 追試(2) 初級編 |

表 3 予備実験の内容

| コンテスト | 出題 | 開始 | 締切 | 期間 | 出題分野 | 構成 (1問の配点) |
|---------|------|----------|----------|-----|-----------|--------------------|
| W04_L08 | 第07回 | 10.23(金) | 11.06(金) | 14日 | 数値計算、関数利用 | 基本8(100) |
| W07_L15 | 第14回 | 11.25(水) | 12.04(金) | 9日 | 反復構文 | 基本5(60) / 応用4(100) |
| W08_L17 | 第16回 | 12.02(水) | 12.11(金) | 9日 | 多重反復 | 基本5(60) / 応用5(100) |
| W09_L19 | 第18回 | 12.09(水) | 12.18(金) | 9日 | 無限反復 | 基本5(60) / 応用4(100) |

表 4 宿題型コンテストの内容

| コンテスト | 出題 | 開始 | 締切 | 期間 | 出題分野 | 構成 (1問の配点) |
|---------|------|----------|----------|----|-----------|--------------------|
| W11_H01 | 第22回 | 01.08(金) | 01.13(水) | 5日 | 一次元配列 | 基本5(60) / 応用5(100) |
| W12_H01 | 第24回 | 01.20(水) | 01.29(金) | 9日 | 多次元配列 | 基本5(60) / 応用5(100) |
| W13_H01 | 第27回 | 01.29(金) | 02.05(金) | 7日 | 関数定義、関数呼出 | 基本5(60) / 応用5(100) |

表 5 宿題型コンテストの結果

| | | W11 | | W12 | | W13 | | 平均 | |
|-----|-----|------|-------|------|-------|------|-------|-------|-------|
| | | 件数 | 比率 | 件数 | 比率 | 件数 | 比率 | 比率 | 正答率 |
| 全体 | 全体 | 1357 | | 1799 | | 1081 | | | 0.508 |
| | 予備1 | 46 | 0.034 | 255 | 0.142 | 104 | 0.096 | 0.625 | 0.091 |
| | 予備2 | 76 | 0.056 | 362 | 0.201 | 203 | 0.188 | 0.621 | 0.148 |
| | 予備3 | 96 | 0.071 | 178 | 0.099 | 122 | 0.113 | 0.262 | 0.094 |
| | 最終0 | 1139 | 0.839 | 1004 | 0.558 | 652 | 0.603 | 0.745 | 0.667 |
| 上位者 | 全体 | 790 | | 1053 | | 688 | | 0.677 | 0.523 |
| | 予備1 | 8 | 0.010 | 107 | 0.102 | 59 | 0.086 | 0.644 | 0.066 |
| | 予備2 | 39 | 0.049 | 209 | 0.198 | 126 | 0.183 | 0.738 | 0.144 |
| | 予備3 | 67 | 0.085 | 126 | 0.120 | 95 | 0.138 | 0.263 | 0.114 |
| | 最終0 | 676 | 0.856 | 611 | 0.580 | 408 | 0.593 | 0.760 | 0.676 |
| 下位者 | 全体 | 471 | | 326 | | 358 | | 0.587 | 0.498 |
| | 予備1 | 53 | 0.113 | 54 | 0.166 | 65 | 0.182 | 0.569 | 0.153 |
| | 予備2 | 29 | 0.062 | 43 | 0.132 | 61 | 0.170 | 0.393 | 0.121 |
| | 予備3 | 33 | 0.070 | 42 | 0.129 | 33 | 0.092 | 0.242 | 0.097 |
| | 最終0 | 356 | 0.756 | 187 | 0.574 | 199 | 0.556 | 0.709 | 0.628 |

表 6 教室型コンテストの出題

| 番 | テスト | 出題 |
|---|-----|---|
| 1 | 設題 | 2つの正の有理分数 a_1/b_1 と a_2/b_2 の積 a_3/b_3 と商 a_4/b_4 を既約分数として求める。 |
| | 予備1 | 積のみを出力する。結果は、約分が不要な場合(GCDが1)に限定する。 |
| | 予備2 | 積のみを出力する。共通因数が2のみの場合(GCDが2の累乗)に限定する。 |
| 2 | 予備3 | 任意の入力に対し、積のみを求める。商の出力は不要である。 |
| | 設題 | 2つの複素数 $x_1 + y_1 \cdot i$, $x_2 + y_2 \cdot i$ の和 $x_3 + y_3 \cdot i$, $x_4 + y_4 \cdot i$ と積を求める。 |
| | 予備1 | 和のみを出力する。出力書式に従う。 |
| 3 | 予備2 | $y_2 = 0.0$ の場合のみに限定し、和と積を出力する。出力書式に従う。 |
| | 予備3 | $x_2 = 0.0$ の場合のみに限定し、和と積を出力する。出力書式に従う。 |
| | 設題 | 初項 a 、公比 r 、項数 n 、上限 m を入力し、等比数列を第0項から第 $n-1$ 項まで、 n 個の各項を順に出力する。 |
| 4 | 予備1 | 上限による打ち切りが起こらない場合に限る。項数と総和は、出力しなくてよい。 |
| | 予備2 | 上限による打ち切りが起こらない場合に限る。次の行では、項数だけ出力する。 |
| | 予備3 | 上限による打ち切りが起こる場合も含む。次の行では、項数だけ出力する。 |
| 5 | 設題 | 得点として、不定個の正整数のデータ列を入力し、入力番号0で打ち切とする。 |
| | 予備1 | 100超上のデータはない場合とする。60以上の合格数のみを出力する。 |
| | 予備2 | 100超上のデータはない場合とする。合格数と平均を出力する。平均は、出力書式に従う。 |
| 6 | 予備3 | 100超上のデータはない場合とする。合格数、平均、二番値を出力する。 |
| | 設題 | 正整数 n を入力し、 $n \times n$ のサイズの2個の数字パターンを横に並べて出力する。 |
| | 予備1 | 左のパターンの左上半だけ出力する。 |
| 7 | 予備2 | 左のパターンだけ出力する。 |
| | 予備3 | 左のパターン、および右のパターンの左上半だけ出力する。 |
| | 設題 | 基数 n を指定し、 n 進数の乗法表を出力する。見出と罫線も付ける。 |
| 8 | 予備1 | 多進法の基数 n 、2数 a, b を入力し、積を n 進数で表示する。 $0 \leq a, b < n$ とする。 |
| | 予備2 | 基数 n に対し、 $n \times n$ の数表として、10進数での積を表示する。見出と罫線は不要である。 |
| | 予備3 | 見出と罫線のみを出力する。横列の罫線の個数に注意する。 |

■ 問題 29-21 有理分数の積と商 [難易度 B] [Note]

● 設題

◎ 要件

2つの正の有理分数 a_1/b_1 と a_2/b_2 の積 a_3/b_3 と商 a_4/b_4 を既約分数として求める。
例えば、入力 4 3 2 3 で、有理分数 $4/3$ と $2/3$ を表し、積 $8/9$ と商 2 から、
8 9 2 1 を出力する。ここで、整数 2 は、分母が 1 の有理分数 $2/1$ とみなす。

◎ 仕様

正整数 a_1, b_1, a_2, b_2 を、1行で空白1字を区切りとして、入力する。
分子、分母の順である。全ての入力は、10000未満とする。

正整数 a_3, b_3, a_4, b_4 を、1行で空白1字を区切りとして、出力する。
最後に改行する。結果が整数になるときは、分母を 1 とする。

◎ 助言

既約分数にするには、分母と分子の最大公約数で、分母分子を割って約分する。
素朴な方法では、分母分子の両方を割り切る共通因数を 2 から順に調べ、
割っていくことを繰り返す。
別解として、最大公約数 (GCD) をユークリッドの互除法で求めて、一気に約分する。

C言語ハンドブックから、ユークリッドの互除法で最大公約数を求める
関数 `gcd()` を使ってもよい。
有理分数の積や商は、分子と分母の2つの値を計算する必要があるため、
そのまま関数として定義することができない。関数を使わず、本体処理で計算する。

図 9(a) 出題例の本文

● 設問 最終 P29-21-0.c [100点]

設題の通りに動作する。

○ 入力
45 8 20 27
○ 出力
25 6 243 32

● 設問 予備1 P29-21-1.c [25%] []

積のみを出力する。結果は、約分が不要な場合 (GCD が 1) に限定する。商の出力は不要である。

○ 入力
2 3 4 5
○ 出力
8 15

● 設問 予備2 P29-21-2.c [50%] []

積のみを求める。共通因数が 2 のみの場合 (GCD が 2 の累乗) に限定する。商の出力は不要である。
約分は、分母分子がともに 2 で割り切れる間、割っていく。

○ 入力
16 7 3 40
○ 出力
6 35

● 設問 予備3 P29-21-3.c [75%] []

任意の入力に対し、積のみを求める。商の出力は不要である。

○ 入力
45 8 20 27
○ 出力
25 6

図 9(b) 出題例の予備テスト

表 7 模試試験の成績別の提出比率と正答率

| 成績 | テスト | 合計 | 比率 | 正答率 |
|----|-----|-----|------|------|
| 全員 | 全体 | 869 | | 0.63 |
| 全員 | 予備1 | 267 | 0.31 | 0.62 |
| 全員 | 予備2 | 162 | 0.19 | 0.61 |
| 全員 | 予備3 | 192 | 0.22 | 0.61 |
| 全員 | 最終0 | 248 | 0.29 | 0.67 |
| 上位 | 全体 | 551 | | 0.75 |
| 上位 | 予備1 | 123 | 0.22 | 0.80 |
| 上位 | 予備2 | 94 | 0.17 | 0.78 |
| 上位 | 予備3 | 141 | 0.26 | 0.67 |
| 上位 | 最終0 | 193 | 0.35 | 0.76 |
| 下位 | 全体 | 318 | | 0.42 |
| 下位 | 予備1 | 144 | 0.45 | 0.47 |
| 下位 | 予備2 | 68 | 0.21 | 0.38 |
| 下位 | 予備3 | 51 | 0.16 | 0.43 |
| 下位 | 最終0 | 55 | 0.17 | 0.35 |

表 8 教室型コンテストの学生の参加状況

| 参加 | 総提出数 | 平均提出 | 教室閲覧 | 教室平均 | 速報閲覧 | 速報平均 |
|----|------|-------|------|------|------|------|
| 81 | 869 | 10.73 | 227 | 5.63 | 167 | 2.06 |

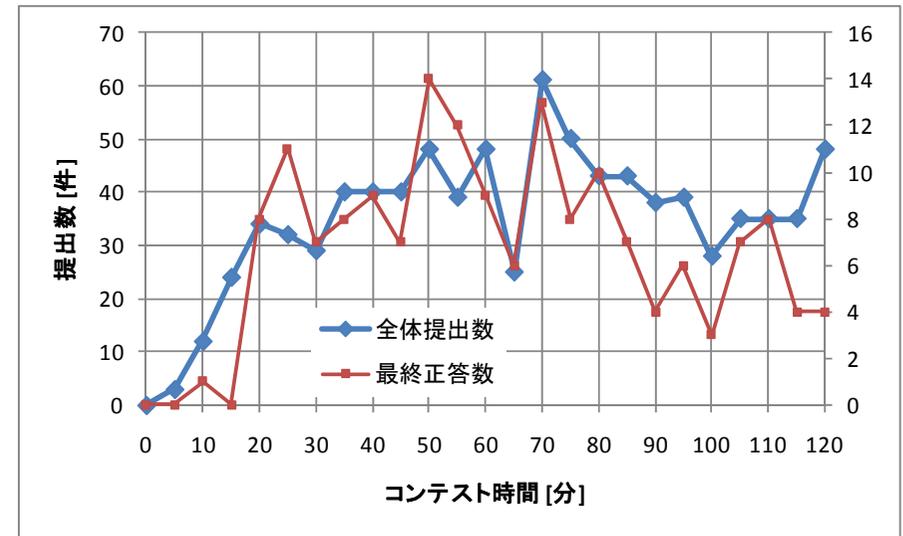


図 10 教室型コンテストの全体提出数と最終テスト正答数の推移

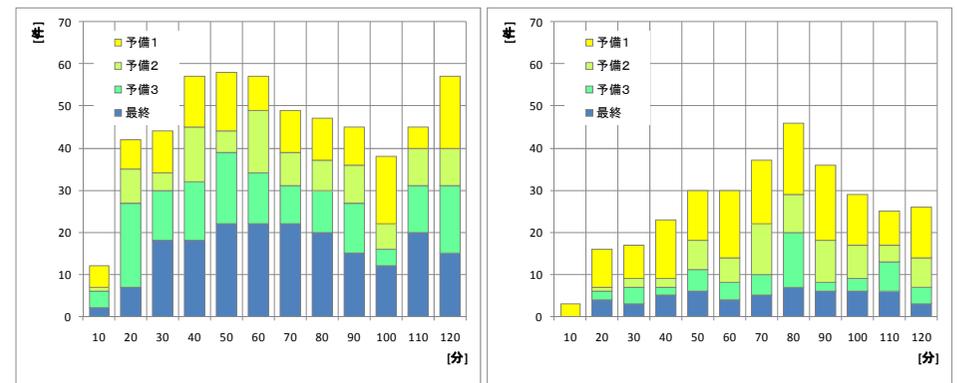


図 11 模擬試験における成績上位群と下位群の提出履歴の時系列グラフ

5. アンケート評価

受験者には無記名で、4段階の客観式アンケートを実施した。自己申告の得点によって、上位から下位まで4群に分けて集計した。81名の受験者に対し、79名から回答を得た。小コンテスト形式の演習の有効性と、システムとGUIの改良の効果を確認する2種類を用意した。また、成績別の分析を行うため、模擬試験の得点を自己申告としてアンケート項目に設けている。なお、アンケートの評価は1~4の4段階で、それぞれ「全くそうである」、「そうである」、「そうでない」、「全くそうでない」とした。

表9に、小コンテスト形式の演習に関するアンケートの結果を示す。1-2-6から、成績のやや良い人は、基準点を達成した後も他の問題へのやる気があると言える。コンテスト形式特有の競争意欲の向上効果がうかがえる。成績のよくない学生も、1-3-1の予備1の利用や、1-3-5の部分点狙いから分かるように、予備テストを上手く利用して演習に参加できている。

表10に、システムとGUIに関するアンケートの結果を示す。2-3-1に着目すると、成績に関わらず提出先を間違える学生が、少なからず存在することが分かる。問題選択ページのGUIに関して、選択している問題を明示するような改良が必要である。また、新機能への要望は全体的に高い結果が出ている。特に、2-4-3の正答でない判定時のヒントについては、全体的に要望が高い。解答が易しくなり過ぎない程度に対応する必要がある。2-5-3の応答速度に関しては、概ね良好な結果となった。パフォーマンスの改善の効果が得られているものと推測できる。

システムへの意見と要望では、自動判定の利便性を評価する意見が多かった。一方で、正答以外の判定結果のときに、エラー箇所の指摘について示唆してほしい、という意見が10件程度あった。また、成績のよくない学生には、「もっとヒントをふやしてほしい」、「間違えたときに助言してほしい」といったコメントもあった。ヒント機能の要望が多く、システムとGUIの評価にも現れていた傾向であり、実装方法と合わせて開発を検討する。

表9 小コンテスト形式の演習に関するアンケートの結果

| 番号 | 質問内容 | 000-199 | 200-299 | 300-399 | 400-600 | 全体 |
|--|---|---------|---------|---------|---------|------|
| 1-1 探索全体の印象 | | | | | | |
| 1 | 模試の得点 1.0~199点 2.200~299点 3.300~399点 4.400~600点 | 1.00 | 2.00 | 3.00 | 4.00 | 2.53 |
| 2 | 受講前からプログラミングの経験があった | 3.86 | 3.88 | 3.67 | 3.29 | 3.66 |
| 3 | 学習事項に対して、理解できた | 3.33 | 2.81 | 2.48 | 2.10 | 2.67 |
| 4 | 授業全体について、満足できた | 3.24 | 2.81 | 2.38 | 2.24 | 2.66 |
| 1-2 演習への取り組み方 | | | | | | |
| 1 | 教科書やハンドブックなどの関連事項をしっかりと読んだ | 2.43 | 2.25 | 2.24 | 2.24 | 2.29 |
| 2 | 友人や授業スタッフに積極的に質問した | 2.48 | 2.13 | 2.33 | 2.52 | 2.38 |
| 3 | 各自のPCでコンパイルに成功してから、システムに提出した | 1.62 | 1.69 | 1.52 | 1.38 | 1.54 |
| 4 | 正答でないとき、コードをよく眺めて、それなりに考えてから再提出した | 2.10 | 1.88 | 1.90 | 1.71 | 1.90 |
| 5 | 宿題の締切前までは、積極的に問題に取り組みなかった | 2.62 | 3.06 | 2.86 | 2.90 | 2.85 |
| 6 | 各回の宿題で、基準点を達成した後は、他の問題へのやる気が少なくなった | 1.95 | 2.56 | 2.57 | 3.05 | 2.53 |
| 1-3 予備テストと最終テストの利用 | | | | | | |
| 1 | 予備テスト1は、難しい問題に対する着手のきっかけになった | 1.95 | 2.44 | 2.52 | 2.52 | 2.35 |
| 2 | 限定された入力のみに対応すればよい予備テストは、中間目標として役に立った | 2.05 | 2.06 | 2.14 | 2.48 | 2.19 |
| 3 | 部分的な出力のみに対応すればよい予備テストは、中間目標として役に立った | 2.05 | 2.13 | 2.24 | 2.43 | 2.22 |
| 4 | 応用問題の予備テスト3は、最終テスト前の確認として、役に立った | 2.10 | 2.19 | 2.52 | 2.14 | 2.24 |
| 5 | 最終テストでの正答を目指さず、早いうちから予備テストで部分点を狙った | 2.38 | 2.81 | 2.90 | 3.00 | 2.77 |
| 6 | 最終テストでの正答を目指したが、途中で予備テストでの部分点に狙いを変えた | 2.86 | 2.56 | 2.86 | 2.57 | 2.72 |
| 1-4 システムを利用して、以下の事項について、学習効果があったと感じますか? | | | | | | |
| 1 | 問題文をよく読み、仕様や入出力サンプルを確認して、題意を正確に把握する | 2.14 | 2.50 | 2.24 | 1.81 | 2.15 |
| 2 | コーディング前に、日本語や図解で処理の概略を理解し、プログラムの流れを掴む | 2.33 | 2.63 | 2.48 | 2.24 | 2.41 |
| 3 | いろいろな入力サンプルで実行テストを繰り返す | 2.43 | 2.25 | 2.00 | 1.81 | 2.11 |
| 4 | ミスの可能性を意識して、それを見つけれられるような入力サンプルで実行する | 2.33 | 2.63 | 2.19 | 2.29 | 2.34 |
| 5 | 少しずつコーディングを進め、こまめにコンパイルして、早めにエラーを見つける | 2.48 | 2.50 | 2.43 | 2.48 | 2.47 |
| 6 | 中間目標を定め、段階的にコーディングを進めて、動くプログラムを作る | 2.52 | 2.69 | 2.33 | 2.48 | 2.49 |
| 1-5 システムを利用して、以下の事項について、意識の変化はありましたか? | | | | | | |
| 1 | 各自が個別に正誤判定を行えるので自分のペースで取り組めた | 2.33 | 2.50 | 2.10 | 1.76 | 2.15 |
| 2 | すぐに結果が分かるので、次の問題に取り組む意欲が増した | 2.33 | 2.56 | 2.05 | 1.90 | 2.19 |
| 3 | 自分の順位や他人の得点を見て、競争意識が生まれた | 2.71 | 2.63 | 2.43 | 2.24 | 2.49 |
| 4 | 時間減点を避けるため、素早く集中して解いた | 2.86 | 2.88 | 2.71 | 2.24 | 2.66 |
| 5 | 誤答減点を避けるため、最終テストの提出前に慎重に確認した | 2.43 | 2.44 | 2.38 | 1.76 | 2.24 |
| 6 | 部分的でも正しく動くプログラムを作成することで達成感が得られた | 2.19 | 2.25 | 2.14 | 2.14 | 2.18 |

表10 システムとGUIに関するアンケート結果

| 番号 | 質問内容 | 000-199 | 200-299 | 300-399 | 400-600 | 全体 |
|------------------------------|---------------------------------------|---------|---------|---------|---------|------|
| 2-1 システム全般 | | | | | | |
| 1 | システムの利用法を理解して、十分に使いこなせた | 2.76 | 2.31 | 2.38 | 2.00 | 2.37 |
| 2 | 問題一覧では、全ての問題の早見ができるとうい | 2.33 | 2.00 | 1.76 | 1.67 | 1.94 |
| 3 | システムの利用マニュアルがあるとよい | 2.14 | 2.06 | 1.95 | 1.86 | 2.00 |
| 2-2 問題閲覧での内容提示とフレーム表示 | | | | | | |
| 1 | フレームによる切替表示は分かりやすく便利であった | 2.24 | 2.38 | 2.19 | 2.19 | 2.24 |
| 2 | システム内でのページで、問題文を読んだ | 2.48 | 2.50 | 2.62 | 2.67 | 2.57 |
| 3 | 設題や入出力サンプルなどの表示の順序は適切だった | 2.33 | 2.38 | 2.24 | 2.24 | 2.29 |
| 2-3 実行テストの選択と答案コードの提出 | | | | | | |
| 1 | 実行テストの選択を間違えて、異なる設問に対する答案コードを出した | 2.24 | 2.31 | 2.48 | 2.24 | 2.32 |
| 2 | Webページ上で簡単なミスを修正し、答案コードをそのまま再提出できるとよい | 2.00 | 1.88 | 1.95 | 1.95 | 1.95 |
| 3 | 答案コードの提出から正誤判定が出るまで十分な応答速度だった | 2.00 | 2.44 | 1.81 | 1.76 | 1.97 |
| 2-4 実行結果と正誤判定 | | | | | | |
| 1 | 正誤判定の6段階の結果表示は分かりやすかった | 2.76 | 2.63 | 2.48 | 2.19 | 2.51 |
| 2 | コンパイルメッセージは、日本語化したり、補足してほしい | 2.14 | 1.69 | 1.86 | 1.95 | 1.92 |
| 3 | 正答でない判定のときに、エラーの原因のヒントが欲しい | 1.62 | 1.75 | 1.33 | 1.90 | 1.65 |
| 2-5 個人進捗と教室進捗 | | | | | | |
| 1 | 問題閲覧ページの上部にある進捗表示を、解答中に確認した | 2.38 | 2.69 | 2.48 | 2.38 | 2.47 |
| 2 | 個人進捗ページで、以前に提出した答案コードについて、後から確認した | 2.29 | 2.50 | 2.24 | 2.43 | 2.35 |
| 3 | 教室進捗ページで、自分の順位や他人の得点を確認した | 2.05 | 2.44 | 2.10 | 2.10 | 2.15 |

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。また、コードの剽窃を検出する機能を組み込んだ。初級 C 演習の授業で、模擬試験として運用実験を行い、提出履歴の分析とアンケート評価を行った。その結果、上位群への意欲の促進、下位群への着手の支援などの効果が確認できた。

今後の課題として、以下が挙げられる。学生側の GUI は、特に提出先を間違えやすい点が問題となっている。そこで、問題閲覧ページにおいて、現在選択している問題とテストを強調する。また、誤答の際に、提出先が異なる可能性を示唆する。

教師側の GUI の充実としては、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を強化する。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。また、問題作成のガイドラインを検討し、学習事項を網羅する問題 DB を構築する。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。

出題については、問題データの記述における人為的なミスにより、コンテスト運用に支障をきたすケースがあった。現時点では、問題の登録自体が煩雑であるため、オーサリングツールを導入し、出題の負担を軽減し、ミスを防ぐ。引き続き、2010 年度の前期授業「プログラミング 2」でも運用を予定している。学習項目は、文字列とポインタ、構造体、ファイル入出力などである。これらの分野で、本研究の手法を適用し、教育効果を検証する。また、提出状況を詳細に分析し、個人の振舞いをモデル化する。各自に適応した、演習支援を目指す。

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", 情処研報, 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", 情処研報, 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", 情処研報, 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 -", 情処研報, Vol.2010-CE-103, No.10, pp.1-8, (2010).