

初学者を対象とした自習中心のプログラミング教育の教材開発と評価

岡本 雅子[†] 村上 正行^{††} 喜多 一^{†††} 吉川 直人^{††††}

本研究では、企業における初級プログラムの育成実践に注目し、参与観察を通じてその分析から、C言語を例としてプログラミング初学者を対象とした自学自習用の教材テキストを開発した。さらに、開発した教材の評価を目的として、京都大学の2009年度の全学共通科目「情報科学演習」の一部としてこのテキストを用いて授業実践を行った。

Development and Evaluation of Computer Programming Course Materials for Self-Learning Novices

Masako Okamoto[†] Masayuki Murakami^{††} Hajime Kita^{†††}
and Naoto Yoshikawa^{††††}

Teaching of computer programming may often stay just introduction of the grammar of programming languages such as C, C++ or Java, and students may fail in acquiring real programming ability. One of the solutions to this inefficiency is to introduce problem-solving with computer programming into programming courses. In such course design, however, classes were used more for introduction and discussion of the problem and students' progress, and students have to study programming language itself by themselves in self-learning style. This paper discusses difficult points encountered by novices in learning of computer programming, and based on the findings we propose authoring guidelines for materials of computer programming for novices. Self-learning material of C language was developed along the proposed guidelines and it is evaluated by using some of them in computer literacy courses.

1. はじめに

プログラミング教育に関する研究は、これまで、大学院、大学、専門学校、高等学校、企業などの場を対象として、多くの実践が行われており、教授法や支援環境に関する研究が数多くある[1,2]。しかしながら、従来のテキストや授業では、特定のプログラミング言語の文法や言語特性に重点を置いたものが多く[3,4]、Robinsら[5]は、「典型的なプログラミングの入門編のテキストでは、詳細なプログラミング言語の知識の獲得に内容を割いている」と結論付けている。

しかしながら、プログラミングは、多様な知識・能力を要求する知的活動であり、文法の修得に重点を置いた教育では、学習者が自立してプログラミングする能力を獲得できるのかという点について疑問が残る。情報技術全般について高等教育段階での教育を検討した米国 National Research Council の報告でも、情報技術に関する知識獲得のみならず、技術変化に対応する自己学習力など様々な能力の養成の必要性を説いている[6]。

プログラミング教育をより効果的なものとする教育方法として、課題達成型の学習が注目されており、先の報告書でも推奨されている。Palumbo[7]も、問題の明確化、要求分析、問題解決に関連した能力に重点を置いたテキストや教授法が重要で、プログラミング言語の学習から問題解決能力の養成へと転換するべきであると指摘している。プログラミング教育に課題達成型の学習を取り入れた場合、授業時間は課題の紹介や進捗の確認などに割く必要があり、多くの場合、プログラミング言語自体の学習は、学習者による自習にゆだねることになる。そこで、学習者が、プログラミング言語を自習することを想定した教育支援を検討することが大変重要になってくる。本研究では、自習中心の研修でプログラマを育成してきた企業における実践についての分析結果から、初学者が指導者による一定の指導は受けられることは想定しつつ、自習を中心にプログラミングを習得できるような教材を開発した。さらに、京都大学における2009年度全学共通科目「情報科学演習」での利用により実践評価を行った。同授業では教材の開発の趣旨から、教員による講義などは必要最小限にとどめ授業内でも自習を中心に授業運営を行った。

[†]京都大学大学院情報学研究科

Graduate School of Informatics, Kyoto University

^{††}京都外国語大学 マルチメディア教育研究センター

Research Center for Multi-Media Education, Kyoto University of Foreign Studies

^{†††}京都大学学術情報メディアセンター

Academic Center for Computing and Media Studies, Kyoto University

^{††††}株式会社キヤミー

CAMMY Co., Ltd

2. C言語自習教材の開発

本章では、プログラミング初学者の躓きを明らかにし、その躓きを考慮したC言語テキストの教材開発について述べる。

2.1 初学者の躓き箇所の分析

本研究では、プログラミング初学者の躓き箇所を明らかにするため、実習を中心としたカリキュラムで初学者にC言語を学ばせているソフトウェア会社において調査を実施した。同社の研修では、初級のC言語の教科書の例題をすべて入力し、実行すること（以下、「写経型学習」と呼ぶ）を求めた上で、その後は課題を与えてプログラムを構成させる手法を取っている。調査として、第一筆者による同社での研修への参与観察、研修修了者を対象としたインタビュー調査を実施した。インタビュー方法は、研修の過程に沿って、行った内容などを逐次確認しながら、その過程を振り返ってもらい、各段階で研修者が感じた困難さを抽出する方法を取った。インタビュー対象者は2名であり、振り返りには、2時間程度のインタビュー5回を要した。実施期間の制約からインタビュー対象者は限られていたが、インタビューで得られた状況とこれまでの経験との整合性などに関しては、研修担当者等、同社の社員に併せてインタビューを行い確認した。

調査結果から、研修担当者の躓きは4つに大別される。

(1) プログラミングに関するコミュニケーション上の問題

プログラミング言語に関わる用語や記号などの音読ができないために、研修者が担当者とのコミュニケーションを円滑に行えないという問題が生じていた。C言語には英語を語源とする短縮形や記号が多用されていることも一因となっていた。

(2) プログラムを構成する基本概念の理解上の躓き

手続き型言語全般に共通する基本的な概念の把握に困難を感じていた。例えば、a)「変数」は、「文字(列)」であるという認識をし、数値などの値を「代入」して用いることが理解できないと考えてしまう、b)ループなど、処理を繰り返すという概念が理解できない、c)if文などで条件を組み合わせて使用できない、などである。

(3) プログラミングに必要な思考法の課題

コンパイルや実行時のエラー箇所を自分で発見できずデバッグができないことが示された。

(4) C言語特有の概念の理解上の躓き

ポインタなどC言語特有の概念の把握に困難を感じていた。

上述の知見(2)については[8]、(3)については[9,10]など初学者を対象とするプログラミング教育に関して、しばしば指摘されているが、同社の研修担当者へのインタビューでも過去の研修者に広く観察されることを確認した。担当者へのインタビューでは、

このほか、プログラミングの基本である逐次実行なども初学者には難しい場合がある。とりわけ初学者が困難さを感じるものとして、for文などの繰り返し動作が挙げられた。また、条件などを組み合わせることについても、困難さが指摘されたが、for文の入れ子の利用も初学者が典型的に困難さを抱えるものである[11]。

一方、(1)については、担当者とのコミュニケーションという側面で初めて顕在化されるため、これまで必ずしも十分に意識されてこなかった点である。

(4)については、C言語の問題として、しばしば指摘されることである[12]が、実務面では、C#やJavaなどポインタを明示的には使わない言語への移行が進んでおり、C言語の位置づけをこれらの言語への導入過程として捉えれば、必ずしも深く教育内容に盛り込む必要はないとも考えられる。

2.2 教材開発

上述の教育における学習者の躓きや学習段階の調査結果を踏まえ、本研究では、新たにC言語のテキスト教材を以下の方針の基に開発した(図1参照)。なお、前節でも述べたように、C#やJavaなどの移行する前段階としてC言語を学習すると位置づけ、ポインタの利用などは最小限に扱う方針を取った(表1参照)。以下の方針は、従来のプログラミング言語の教科書と比較して、プログラミング言語の文法面を網羅的に紹介することよりもむしろ、学習者の学習プロセスと実践的なプログラムやプログラミングのプロセスにより配慮したものとなっている。

(1) 学習の目的、学習の仕方など、学ぶ内容とともに「学び方」の明示

(2) 音読への配慮

コミュニケーションの円滑化目的に、プログラミングで使用する記号には読み方を、予約語や良く使われる表現には、読み方および適切な日本語訳を表記した。また、英語を語源とする短縮形が多用されていることから、語源とその読み方も表記した。学習者が読めない記号や単語があった場合、一覧表の中からそれらの読み方を探せるようになっている。さらに、英語を語源とする短縮形や記号を多用し表現をコンパクトにするC言語の特徴に配慮して、プログラムを日本語に訳して読ませることを導入し、「日本語からプログラムへ」「プログラムから日本語へ」の翻訳例を提示した。

(3) プログラミングの基本概念の理解への支援

本テキストでは、写経型学習での自習を前提にサンプルプログラムについては実行可能な完全なソースコードと、その実行結果を必ず示す形で用意した。

そして、サンプルプログラムを対象に、その逐次的解説とプログラム構成要素の紹介も併せて行った。プログラミング言語の文法の詳細は他の参考書を参照させるものとし、実践的に利用するポイントに焦点を当てて解説した。また、プログラムの動作の理解を助けるために図解なども重視した。

(4) 実利用に配慮した例題の選択

本テキストは、プログラミング言語中心であるという点で従来のテキストと同様で

あるが、言語使用の紹介よりもむしろ実際のコーディングでよく使用される「使い方」に配慮した内容選択を行っている。つまり、本テキストに示す例題は、プログラミング言語の文法の習得とともに、それを実際に使用する場面に配慮して選択した。また、プログラミングに慣れるため、バリエーションの練習として、サンプルプログラムの内容を少しずつ変更させたものを多数用意した。

(5) プログラミングのプロセス面での学習の支援

市販のテキストでは、あまり説明されない落とし穴や作法など、プログラミングという活動のプロセス面を意識し、「コラム」として紹介した。また、デバッグでの躓きを軽減するために、補足資料として、いくつかの典型的なエラーについては、エラーメッセージとともに対処方法も紹介した。

(6) 実習での使用への配慮

本テキストは、それを見ながらコンピュータに向かって実習することに配慮し、見開き 2 ページで完結する内容とし、1 つのサンプルプログラムに複数の学習内容を詰め込み過ぎないように配慮した。

目次	例題数
学習の仕方を理解する	-
プログラミングに使用する文字と用語	-
C 言語プログラムの開発手順	-
プログラムを作成してみよう	8
画面に表示する方法	5
データを入力する方法	2
値を保存する方法	8
処理の流れとフローチャートを理解する	-
条件によって処理を分ける方法	6
変数を一括管理する方法と繰り返し処理	11
配列の便利な使い方	2
for 文ドリル	74
繰り返し処理(while 文)	6
ポインタ	4
関数	7
複数のデータ型を 1 つにまとめる方法(構造体)	5

3. 京都大学での実践

開発したテキスト教材の評価を目的に、2009 年度京都大学全学共通科目としてコンピュータリテラシ全般を扱う「情報科学演習」の一部として、同教材を利用した。本授業は、前期および後期に週一コマの科目として開講されており、それぞれ同様の授業を実施している。本実践では、5 回の授業をプログラミング学習にあてている。本授業でプログラミングを教授する目的として「プログラムの論理構造を理解してもらう」ことに主眼を置いており、本教材を利用して、繰り返し練習することにより、その論理構造を理解してもらうことを目指している。この論理構造を身につけることは、今後、プログラミングを学習する上で重要であることはもちろんのこと、一般的な物事を考える上でも有用であると考えており、文科系の学生がコンピュータを介して、このような内容を学習することにも大きな意義があると考えている。上述の趣旨を踏まえ、この科目では本教材の使用前からプログラムの動作の分かりやすい C 言語を選択している。学習用に開発された言語の採用なども考えられるが、授業開始時に「どのようなプログラミング言語を学びたいか」と問うたところ「聞いたことのある C 言語を学習してみたい」との要望も多く、C 言語の採用は学習の動機づけにもなると考

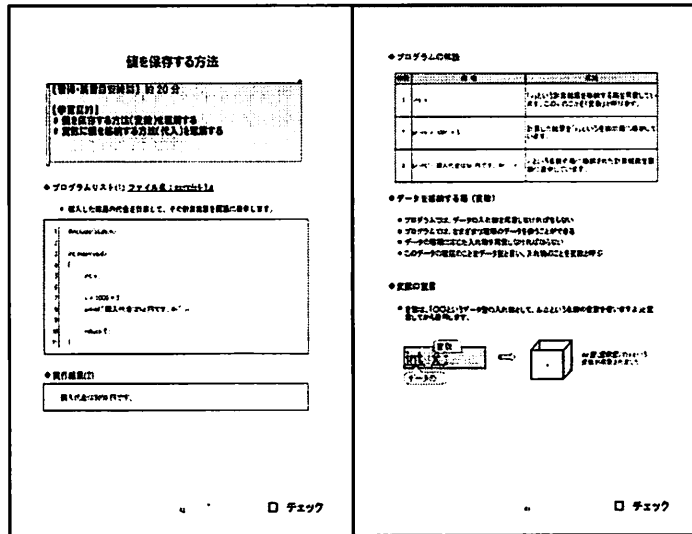


図 1 開発したテキストのサンプル

えている。

プログラミングの授業は、プログラミングの基本的な概念の習得を目的として計 5 回分実施した。学習する内容は時間が限られていることから、変数と制御構造を中心とし、関数や構造体は取り上げなかった。教材の設計意図を踏まえ、同科目では、講義と写経型学習による自習とを取り入れた授業形態を採用し、講義での解説はポイントを絞って行った。また、受講生は、自立的に学習できる教材に従って、受講生間で細かな同期を取ることもなく、各々の学習進度に応じて学習を進めた。学生が躓いた場合は、教員や TA (1 名) がサポートに入り、直接指導を実施した。

なお、同科目は、総合人間学部向けの授業であったが、前期後期ともに 6 割以上が経済、理学、農学など他学部の学生であり、受講生は前期 34 名、後期は 22 名であった。演習には、京都大学の教育用コンピュータ端末を用いた。前期は、Linux の動作する遠隔サーバにログインして gcc を使う形で、後期には、Windows 上で MinGW を用いて実習した。

3.1 受講生の前提知識

本実践では、プログラミングの授業を開始するにあたり、受講生のプログラミング経験を把握するため、事前アンケートを実施した。アンケートではプログラミングの経験のほか、既習のプログラミング言語や概念、実装したプログラムの規模などを回答させたが、前期・後期ともに受講生は、プログラミング経験のないもの（以下、未経験者）が大多数を占めた。調査結果を表 2 に示す。また、プログラミングの授業に対する意欲を知る目的で、「プログラミングは必要か」を問うたところ、前期(図 2)・後期(図 3)ともに約 7 割の学生が必要を感じていた。

表 2 受講生のプログラミング経験

プログラミング経験	前期	後期
あり	1 名	3 名
なし	31 名	15 名
無回答	2 名	4 名

3.1 2009 年度の実践結果

学習者の自習を中心としたプログラミングの授業と教材、学習内容の難易度に関する評価をするため、授業の最終日にアンケートを実施した。その集計結果を図 4 に示す。図 4 は、授業と教材への調査結果である。

3.1.1 2009 年度前期の結果

2009 年度の前期においては、プログラミングの授業の満足度に関しては、約 9 割もの学生が肯定的な評価を行っている。このほか、テキストのレイアウトや文書表現に

ついても高い評価を得た。しかしながら、前期における授業外の取り組みに関しては、否定的な評価を示す学生が多かった。その要因として、Linux 環境で授業を実施していたため、自宅に Windows 環境しかない学生にとっては、自宅での学習が困難であったことが考えられる。一方、自由記述から、サンプルや課題に関しては、量が多いと感じた学生が多く、授業時間内にすべて取り組むことが困難であったのではないかと推察される。

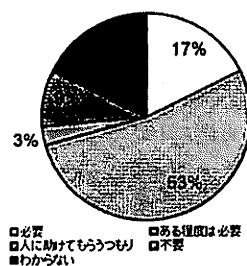


図 2 プログラミングの必要性(前期)

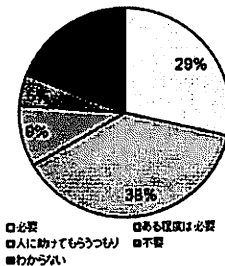


図 3 プログラミングの必要性(後期)



図 4 最終アンケート結果

次に、図5は、学習内容の難易度に関する調査結果である。サンプルプログラムを入力し、コンパイル、実行することについては「易しかった」「やや易しかった」とする回答が多く、難易度は低かったようである。一方、プログラミングの概念として、変数や代入、繰り返し処理、配列に関しては、「難しかった」「やや難しかった」と回答する学生が多かった。しかし、学習者の授業ごとの学習状況から、変数についてはプログラムで実装することができており、この項目に関しては、授業の初期段階に学習したということもあり、エラー箇所を発見や修正に労力を割かなければならず、「難しかった」という印象を強く持ったのかあるいは精神的な負担が大きかったのが要因として考えられる。また、繰り返し処理や配列に関しては、授業時間に制限があるため、自身の学習ができなかったため、「理解できなかった」と回答していると思われる。

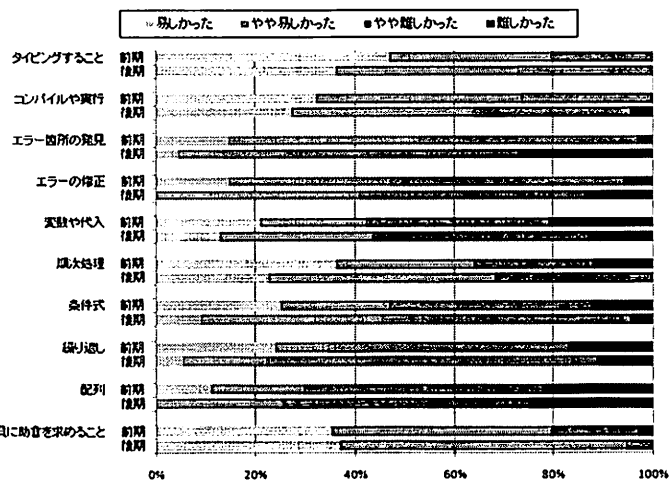


図5 学習内容の難易度

また、「ポイントのみの講義」と「自習」で構成するプログラミングの授業に関しては、「自分のペースで学習できて良かった」「置き箇所を直接開けて良かった」など肯定的な意見が見えられた。講師へ質問することへの難易度は、8割以上の学生が容易であったと回答しており、比較的、質問することへの敷居が低かったと推察される。

また、多くのサンプルプログラムからプログラミングを学ぶ学習法(写経型学習)とそのため構成した例題に関しては、「繰り返しの演習により、プログラミングを体で身につけることができた」「実際に自分で打ってみることで理解できたのでサンプルを写す学習は効果的だと思います」「サンプルと見比べてエラーが発見できる」など肯定的な意見が数多く見受けられた。

最後に、受講生の理解度を測るため、「理解度アンケート」という形式でテスト(成績評価には反映しない)を実施した(付録 理解度アンケートの設問を参照)。同アンケートは、1)文法エラーの間違い探し、2)変数・printf・scanfに関する問題、3)if文に関する問題、4)for文と配列に関する問題から構成される(補足資料参照)。その結果を図6に示す。回答は35点満点で計算し、平均22.7点であった。また、設問ごとの平均点を表3、設問ごとの得点分布を表4~7に示す。4の)for文と配列に関する問題の平均点が3.03と低く、回答した学生は全体の2割しかいなかった。さらに、2)の変数などを問うた問題においては、変数の型、printfやscanfの型の指定などができていない学生が多数見受けられた。1)の文法エラーと3)のif文に関しては、全体的に理解度が高かった。

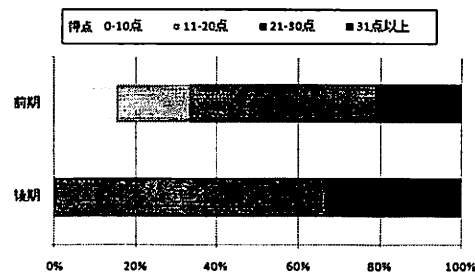


図6 理解度アンケート(テスト)の得点分布

表3 設問ごとの平均点

問題番号	前期	後期
1 (5点満点)	4.84	4.67
2 (10点満点)	8.06	10.00
3 (10点満点)	7.03	10.00
4 (10点満点)	3.03	4.33

表 4 設問 1 の得点分布

得点分布	前期	後期
0	0%	0%
1	0%	0%
2	0%	0%
3	0%	0%
4	15%	33%
5	85%	67%

表 5 設問 2 の得点分布

得点分布	前期	後期
0	9%	0%
1	0%	0%
2	0%	0%
3	0%	0%
4	0%	0%
5	9%	0%
6	0%	0%
7	9%	0%
8	12%	0%
9	9%	0%
10	52%	100%

表 6 設問 3 の得点分布

得点分布	前期	後期
0	21%	0%
1	0%	0%
2	6%	0%
3	0%	0%
4	0%	0%
5	3%	0%

6	0%	0%
7	3%	0%
8	6%	0%
9	0%	0%
10	61%	100%

表 7 設問 4 の得点分布

得点分布	前期	後期
0	64%	40%
1	0%	0%
2	0%	0%
3	0%	0%
4	0%	0%
5	12%	33%
6	0%	0%
7	0%	0%
8	0%	0%
9	0%	0%
10	24%	27%

3.1.2 2009 年度後期の結果

前期の結果を踏まえて、2009 年度の後期の授業では、次のような改善を行った。1) プログラミングの開発環境を自宅などでの自習に配慮して Linux から Windows 環境に変更した、2) 使用するテキストの量を、例題を減らす形(for 文ドリルを任意とした)で前期の 2/3 程度に削減した、3) 前期には第 2 回目を実施していたエラーの対処法の講義を第 1 回目を実施した。

その結果、2009 年度の後期においては、授業外の取り組みに関しては、肯定的な評価を示す学生が多く見られた。さらに、全体の理解度に関しては、7 割近くの学生が肯定的な評価を示し、前期の結果と比べて大幅に増加している。プログラミングへの興味や今後の学習への意欲なども高い評価を得た(図 4 参照)。

理解度アンケートでも前期 22.9 点であった平均点が 30.2 点と 7.3 ポイントも上昇しており、実際にプログラミングへの理解度も高まっている。前期では、変数の設定などで躓く学生が多く見受けられたが、後期では、躓く学生が見受けられなかった(表 5 参照)。for 文と配列に関する問いでは、前期は、自習が追いつかず回答できない学生

が多かったが、後期では、全員が回答していた。これは、テキストの分量を減らしたことと自習に配慮した学習環境を用いたことで、最後まで学習でき消化不良が軽減されたことが推察される。

なお、学生による学習項目別の難易度の感想(図3参照)は前期と同様の傾向を示している。

このことから、2009年度の前期の評価に基づく授業や教材の改善により、2009年度後期では、学生自身の理解度の状況に関しては高低差があるものの、未経験者の躓きや消化不良が軽減したものと考えられる。

3.1.3 考察

以上の実践から、写経型学習を想定して作成した教材と学習法は学生から肯定的に受け入れられ、教員に助言を求めることも学生から肯定的な評価を得た。実際の学習成果も得られており、教材開発の意図が達成されていることが示唆されている。

同教材を利用する授業実践においては、学生の自宅などでの自習環境に配慮すること、教材として利用する例題の量を調整することが必要であることが示された。なお、実践条件から作成した教材をすべて使用することはできなかった。未使用箇所には関数など重要なプログラミング上の手法を取り扱っているものが含まれており、その評価は今後の課題として残されている。

4. おわりに

本論文では、初学者の躓きを考慮した自習を中心としたプログラミング教材を開発し、京都大学における2009年度全学共通科目「情報科学演習」において同教材の評価を行った。その結果、プログラミング初学者に対する学習効果が見られ、開発した教材がその意図を実現していることが示唆された。

今後の課題として、教材のうち、未使用箇所の評価を行うことに加え、開発した教材は主にプログラムの構成要素の学習に焦点を当てて作成されたものであり、プログラミングによる問題解決能力を養成についてはさらなる教材開発が必要である。これについても現在、研究を進めている。

参考文献

- 1) 田口浩ほか: 個々の学習者の理解状況と学習意欲に合わせたプログラミング教育支援, 情報処理学会論文誌, Vol.48, No.2, pp.958-968(2007).
- 2) 王文涌, 池田満, 李 峰榮: プログラミング教育における動機づけ教授方法の提案と評価, 日本教育工学会論文誌, Vol. 31, No.3, pp.349-357(2007).
- 3) A. Pears et al.: A survey of literature on the teaching of introductory programming, ACM SIGCSE Bulletin, Vol.39, Issue 4 (2007).

- 4) M. Linn, M. Clancy: The Case for Case Studies of Programming Problems, Communications of ACM, Vol. 35, No. 3 (1992).
- 5) A. Robins, J. Rountree, and N. Rountree: Learning and teaching programming: a review and discussion, Computer Science Education, Vol.13, No.2, pp.137-172 (2003).
- 6) Committee on Information Technology Literacy, National Research Council: Being Fluent with Information Technology, National Academy Press (1999).
- 7) D. Palumbo: Programming language/problem-solving research: a review of relevant issues, Review of Educational Research, Vol.60, No.1, pp.65-89 (1990).
- 8) 安達一寿, 中尾茂子: プログラミング学習における学生のつまづき箇所の分析, 日本教育情報学会学会誌, Vol.10, No.4, pp.11-20(1995).
- 9) 江木鶴子, 竹内章: プログラミング初心者にトレースを指導するデバッグ支援システムの開発と評価, 日本教育工学会論文誌, Vol.32, No.4, pp.369-381(2009).
- 10) 太田信宏: Java プログラミング教育に関する一考察, 文教大学女子短期大学部研究紀要, Vol.52, pp.1-16(2009).
- 11) 阿部圭一: C 言語によるプログラミング教育についての省察, 情報処理学会研究報告, コンピュータと教育研究会報告, Vol.2009, No.15, pp.205-212(2009).
- 12) 朝倉宏一, 渡邊豊英: 情報工学系学部学生に対する並列プログラミング演習教育, 情報処理学会論文誌, Vol.40, No.5, pp.2235-2245(1999).

付録 理解度アンケートの設問

- (1) 次のプログラムは、「プログラムの間違い探し」という文字列を表示するプログラムです。間違っている箇所が5ヶ所存在します。それをすべて指摘しなさい。
※エラー処理の練習になるので実際に入力して確認しながら回答してもかまいません。

```
● 以下のプログラム
1 .....
2 * プログラムの間違い探し
3 .....
4 #include <stdio.h>
5
6 main()
7 {
8     printf("プログラムの間違い探し\n");
9
10    return 0;
11 }
```

- (2) 次のプログラムは、摂氏で入力された温度を華氏に変換して表示するプログラムです。空欄に相当するプログラムを記述しなさい。なお、温度の摂氏から華氏への変換は、「華氏温度 = 摂氏温度 * 9/5 + 32」である。

• 実行例

```

    送料を自分で入力して下さい。 300
    郵便番号は、9919077 が必要です。
    
```

プログラムリスト

```

    #include <stdio.h>
    int main(void)
    {
        //
    }
    return 0;
    
```

(3) 次のページのプログラムは、数量と単価を入力して、その合計金額に応じた送料を表示するプログラムです。空欄に相当するプログラム(15行程度)を記述しなさい。なお、送料は以下の通りとする。

金額	送料
~ 1,000	1,000円
1,000 ~ 10,000	200円
10,000 ~ 20,000	500円
20,000 ~ 50,000	800円
50,000 ~	1,000円
	合計

• 実行例

```

    郵便番号を入力してください。 9900
    数量を入力してください。 1.5
    合計 3000
    送料 600
    合計金額 3600
    
```

プログラムリスト

```

    #include <stdio.h>
    int main(void)
    {
        int kante;
        int nnyou;
        int kngsu;
        int sanyou;

        printf("郵便番号を入力してください。");
        scanf("%d", &kante);

        printf("数量を入力してください。");
        scanf("%d", &nnyou);

        //
    }
    return 0;
    
```

(4) 次のプログラムは、初項3 公差5 の等差数列を計算し、8番目の値「38」を出力するものです。このプログラムについて以下の問題に答えなさい。

```

    #include <stdio.h>
    int main(void)
    {
        int a[10];
        int k = 3, s = 5;
        a[0] = 3;

        for(i = 0; i < 10; i++)
        {
            a[i] = 0;
        }

        for(i = 1; i < 8; i++)
        {
            a[i] = a[i-1] + s;
        }

        printf("8番目の値 = %d\n", a[8]);

        return 0;
    }
    
```

① 上のプログラムには、空欄の部分が1ヶ所存在する。その行番号と誤りを訂正したプログラムを記述しなさい。

行番号	訂正したプログラム

② ①の部分を変更すれば文法上は正しいプログラムとなるが、他に1ヶ所誤りがあるため、期待した結果(38)を出力することができない。誤りのある行番号と修正したプログラムを記述しなさい。

行番号	訂正したプログラム