

# アルゴリズム的思考法に関する学習を取り入れたC言語による プログラミング教育の実践と評価

—2010年度から2013年度の4年間の実践より—

## Practice and Evaluation for C Programming Education including Algorithmic Thinking

Research Report for 2010 to 2013

萱津 理佳<sup>\*1§</sup>、香山 瑞恵<sup>\*2</sup>、國宗 永佳<sup>\*2</sup>、永井 孝<sup>\*3</sup>、不破 泰<sup>\*4</sup>

Rika KAYATSU, Mizue KAYAMA, Hisayoshi KUNIMUNE, Takashi NAGAI, Yasushi FUWA

### あらまし

C言語によるプログラミング教育において、アルゴリズムを考える力を養い、プログラム作成に必要な十分なスキルを身につけさせることを目的とし、アルゴリズム的思考法に関する学習を取り入れた。本研究では、思考したアルゴリズムの外化にあたり、表現変換の負荷を減らすため、グラフィカルなインタフェースによりアルゴリズムを組み立てるビジュアル・ブロック・プログラミング可能なアルゴリズム学習向けWebツールを利用した。本論文では、ツールを利用した授業の概要及び2010年度から2013年度の4年間の実践について報告する。また、学習導入の効果を評価するため、アルゴリズム的思考能力およびプログラミング能力について本学習の導入前の受講生との比較を行った。プログラミング学習の初期段階でのアルゴリズム的思考法の導入がプログラミング学習において効果的であることが明らかとなった。

キーワード アルゴリズム的思考法 プログラミング教育 C言語 教育支援ツール

### 1. はじめに

社会の情報化に伴い、プログラミング教育の重要性は高まってきている。しかしながら、プログラミング教育における現場では、プログラミング教育が開始された当初からほとんど変わらない形式で授業が行われることが多いのが現状で、まず講義形式で対象となるプログラミング言語の文法について解説を行い、実習としてプログラムを作成せよという形式が一般的である。

筆者は、国立高等専門学校電気電子工学科3年生のプログラミングの授業で初学者を対象としC言語を指導している。この従来のプログラミング教育における授業方法では、文法知識については理解していても、実際に課題としてプログラムを作成させると、プログラムが記述できない受講生や、例題に提示したプログラムを組み合わせ、プログラムの動作から場当たりに問題点を修正していくといった受講生が多いことが顕在化してきた。プログラムを

作成するためには、プログラミング言語の文法の習得だけでなく、実行したい処理をアルゴリズムの形に分解・整理する思考（アルゴリズム的思考）の習得が必要である。これまでの授業では文法の学習に重点が置かれ、例題を通して経験的にアルゴリズムを考える力を養うという形態になっていた。しかしながら、プログラミング教育の過程でアルゴリズムを学ぶ形式は、アルゴリズムを考え、表現する力を養う体系として未整理の段階である。その結果、アルゴリズム的思考が十分に身につけていない受講生が顕在化してきたと思われる。

そこで本研究では、プログラミング教育にアルゴリズム的思考法に関する学習を取り入れることで、アルゴリズムを考える力を養い、プログラム作成に必要な十分なスキルを身につけさせることを目的とし、授業実践を行った。まず、課題が与えられた際に最初からプログラムコードの作成に入るのではなく、課題に対するアルゴリズムを日本語箇条書きで表現させる方法を実践した。しかしながら、アルゴリズムを自然言語で記述させた場合、自由度が大き

\*1 長野県短期大学多文化コミュニケーション学科国際地域文化専攻 \*2 信州大学工学部情報工学科

\*3 信州大学大学院総合工学系研究科 \*4 信州大学総合情報センター

§1 連絡先 〒380-8525 長野県長野市三輪8-49-7 TEL 026-234-1221 FAX 026-235-0026

すぎ、受講生によって処理の粒度がまちまちで、曖昧な表現も多くみられた。また、作業を基本的な操作に分解できていない受講生もみられた。そこで、いくつかの記述ルールを形式化し、グラフィカルなインタフェースによりアルゴリズムを組み立てるビジュアル・ブロック・プログラミング可能なアルゴリズム学習向け Web ツール (Algo Tool) [1][2][3] の導入を試みた。

本論文では、プログラミングを初めて学ぶ「プログラミング言語 I」における 2010 年度から 2013 年度までの 4 年間の授業実践について報告する。以下、2 節でアルゴリズム的思考法に関する学習の先行研究について、Algo Tool の概要、そして授業の概要について述べる。3 節では、アルゴリズム的思考法に関する学習を導入した効果を、アルゴリズム的思考能力、プログラミング能力、配列に関する理解の三つの視点から考察する。また、Algo Tool を利用した受講者を対象としたアンケートよりツールの利用に関しての評価を示す。最後に 4 節でまとめを行う。

## 2. アルゴリズム的思考法に関する学習の導入

### 2.1 アルゴリズム的思考法について

アルゴリズム的思考法に関する学習を取り入れるにあたり、本研究ではアルゴリズムを「ある目的を達成するための処理の手順を順次・判断・繰返しのみの組み合わせで表現すること」ととらえた。その上でアルゴリズム的思考法を「与えられた目的を達成するための処理手順を考え、それを他者に伝えるために形式化すること」と定義する[4]。この形式化とは特定の表記規則で処理手順を記述することである。

アルゴリズムに関する研究では、わが国では 1960 年代に端を発し、まずは教授行為への応用が試みられた[5]。駒林は小学校 5 年生の幾何算数の教授に際して、論理操作を表現する手段としてアルゴリズムモデルを適用した結果、論理操作の形成にアルゴリズムを手段とすることが有益であることを実験的に示している [6]。

アルゴリズム教育は、アルゴリズムそのものを教える教育であって、ノイマン型のコンピュータの仕組み、プログラミング言語の文法、アルゴリズムという順序を経るのが一般的である。このため、プログラミング学習に付属した形で、既存のアルゴリズムのサンプルプログラムを実行して、目的の結果を得ることによって行われることが多かった。これに

対し、飯田らはアルゴリズム教育とは分けて、アルゴリズム的思考法の教育を行った[7]。ここでは、プログラミングとは独立に授業が行われ、表記にフローチャートが用いられている。実際に行ったテストの分析から、十分に理解している学生と同時に、アルゴリズムの理解を暗記に頼っている学生やアルゴリズムの流れを理解していないと判断できる学生も数多く確認されている。また、この方法では、フローチャートの文法説明が必要であり、本質的なアルゴリズムの議論が十分に行えないなどの課題を示唆されている。加えて、「アルゴリズムとはどういうものか」ということを体験的に学習する必要があると提案している。村上らはアルゴリズムの基礎力を育成するために工夫された教材や指導法、試験法を提案している[8]。ここでは自動販売機の演習や、ハノイの塔での記号表現の演習などいくつかの教材が紹介されている。また、学習者の理解の程度を確認するための試験法として、既存のフローチャートを読み取り、何を目的としているのかを自然言語で記述させることで、理解を確認するワンボックスフローチャートによる試験法、問題と答えを与え、その問題の解釈と答えに至るまでの過程を説明させるプロトコル試験法、そして言葉を提示し、その言葉が表す概念を説明する文章を単文で表すステートメントテストによる試験法が提案されている。ただし、これらの提案を継続的に実施した教育効果についてはまだ報告されていない。

### 2.2 アルゴリズム学習向け Web ツール: Algo Tool

アルゴリズム的思考法の学習を導入するにあたり、より効果的な学習を目指し、独自に開発されたビジュアル・ブロック・プログラミング可能なアルゴリズム学習向け Web ツール (Algo Tool) [1][2][3] を利用した。

本ツールは、学習者自身によるアルゴリズムの記述の検証、解答の見直しや参照を可能とし、教員による採点の労力の軽減を図り、より曖昧さが少ない形式化を意識させるために開発されたツールで、アルゴリズムを組み立てながら学習する Web 教育環境である。学習者がログインした際のインタフェース例を図 1 に示す。ホーム画面では、保存済みの課題と提出済みの課題の一覧が表示される。

本ツールの特徴は以下の 5 点である。

#### 1. 構造記述と内容記述の分離

課題作成画面の例を図 2 に示す。アイコンで示される構成要素 (ブロック) には、計画、変数

宣言、計算、判断、繰り返し、出力といったアルゴリズムを組み立てるための要素が対応付いている。各ブロックはマウスにより挿入や移動、削除等の操作ができ、所定の場所にドラック&ドロップすることでアルゴリズムの構造を表現する。これにより表記法を意識することなく、思考をブロックの組み合わせに集中させることができる。さらに、各ブロック内の空欄に変数や数値、処理内容等を入力することで具体的な処理を表現する。ブロック内での処理は、日本語に近い表現となっており、学習者の思考とアルゴリズム表現をより近似させる意図がある。

## 2. 構成要素の利用の可否を制御

管理者権限のユーザは、学習進度に応じて判断や繰り返し等のアルゴリズムを構成する要素（ブロック）や演算子の利用の可否を制御できる。これにより、使用可能なブロックや演算子を制御された中でアルゴリズムを考える状況を指導者が意図的に設けることができる。図3にブロック利用に関するルール登録画面例を示す。利用できないブロックは課題作成画面において非表示となる。利用可能なブロックが制限された課題の例を図4に示す。図2の課題作成画面と比較すると表示されているブロックの数が



図1 学習者のログイン画面



図2 課題作成画面



図3 利用ブロックルールの登録画面



図4 使用ブロックが制限された課題の例

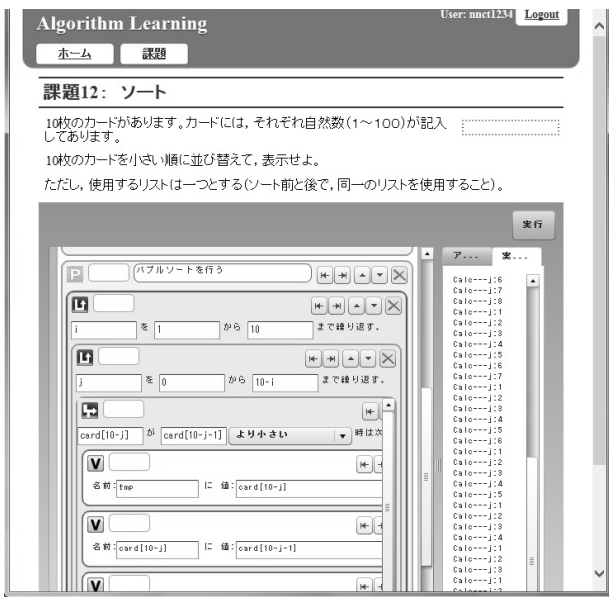


図5 トレース実行例

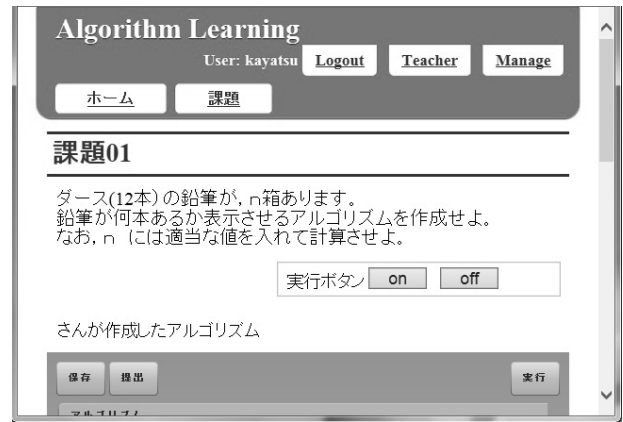


図6 指導者による実行制御



図7 課題一覧・提出状況の確認



図8 提出課題の閲覧

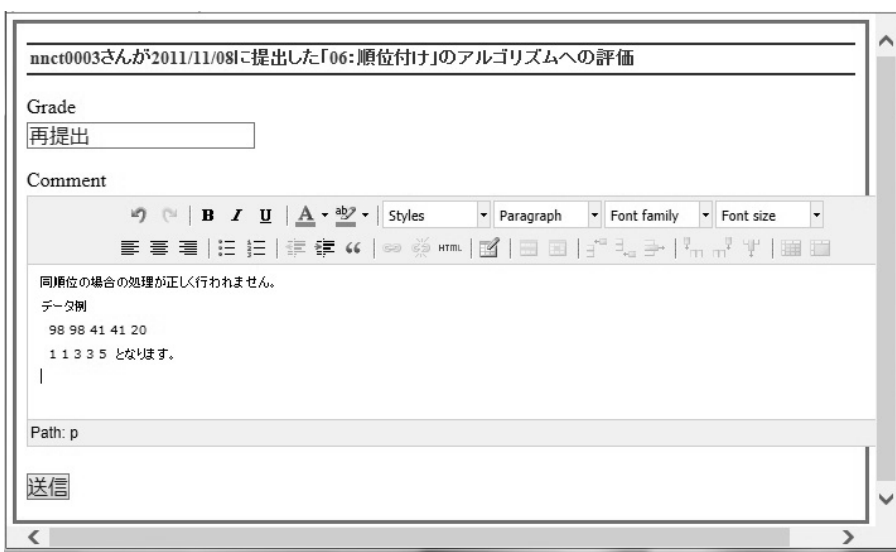


図9 指導者による評価

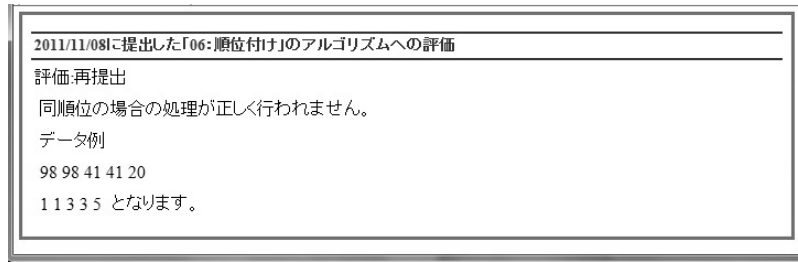


図 10 学習者による評価閲覧

少なくなっている。未学習のブロック利用を否にすることで、初学者がアルゴリズムを考える際の認知的負担を軽減できる。

### 3. アルゴリズムのトレース機能

アルゴリズムの振る舞いを検証する変数のトレースができる。トレースの実行例を図5に示す。また、指導者は図6に示す実行ボタンのオン・オフによりトレース機能の利用の可否や利用間隔を制御可能である。課題や学習進度に応じて、トレース機能をアルゴリズムの確認やセルフデバッグに利用する方法と、トレース機能を使用させずに学習者の思考を深める方法とを切り替えることができる。

### 4. 課題のテンプレート登録機能

課題とともにアルゴリズムの一部もしくは全体を学習者に提示することができる。これにより、アルゴリズムの記述だけでなく、穴埋め問題や与えられたアルゴリズムを読ませる（流れを追わせる）、修正させる等、多様な学習形態が可能である。

### 5. 提出物の状態管理

指導者権限のユーザは、課題の登録のほか提出物の作成中・提出済、未評価・評価済、再提出等の状態を学習者ごと、課題ごとに確認できる。図7に課題一覧表示による提出状況確認画面、図8に提出課題の閲覧および評価一覧画面の例を示す。指導者は個々の提出課題ごとに評価やコメントを与え（図9）、学習者にフィードバックすることができる（図10）。

## 2.3 授業実践の概要

プログラミングを初めて学ぶ「プログラミング言語I」の授業において2010年度よりアルゴリズム的思考法の学習を取り入れた。本科目は、国立高等専門学校電気電子工学科3年生の通年、必修科目である。授業は30回実施され、定期試験を4回（前期中間、前期期末、後期中間、学年末）行う。主な授業内容（範囲）を以下に示す。

- ・前期前半：コンピュータおよびプログラムの基礎、データの入出力、選択処理
- ・前期後半：繰返し処理
- ・後期前半：配列
- ・後期後半：ユーザ関数、ポインタ

これまで本授業ではアルゴリズム的思考の欠如に加え、C言語の学習において配列で躓く受講生が多いことが問題として挙げられた。そこで、2010年度および2011年度はアルゴリズムを考え記述する力のほか、C言語の配列理解に焦点をあて、後期前半の配列学習に入る前の段階でツールを利用したアルゴリズム的思考法に関する学習を5回実施した。両年度とも、前期よりC言語によるプログラミングの学習を始め、ツール導入時点で入出力、選択処理、繰返し処理まで学習済みであった。2010年度は、ツールの操作に慣れてもらうための導入課題2題、配列を使用する課題7題の全9題を課した。2011年度は配列の課題に入る前に、ツールの操作に慣れるための課題に加え、これまでC言語で学習した範囲の選択処理、繰返し処理に関するアルゴリズムを記述できるか、また記述されたアルゴリズムの動作が理解できるかを確認および復習するための課題を8題新たに導入した。

2年間の実践から、配列に関する理解については一定の効果が現れたが、C言語を半期学習した時点でツールを利用することに、戸惑う受講生がいることが明らかとなった。アルゴリズム的思考法の学習で利用したツールは、プログラミング言語の文法にとらわれずアルゴリズムをブロック単位で組み立て、日本語による箇条書きでアルゴリズムを記述することができる。しかしながら、プログラミングスキルの高い受講生には煩わしく感じる一面があることも否めない。

そこでツール利用の三年目である2012年度は、C言語の学習を始める前の授業の初期段階からツールを利用したアルゴリズム的思考法の学習を取り入れた。ツールを利用した授業の回数は、2010年度、2011年度と同様の5回とし、そのうち3回を前期

前半で、残り2回を後期前半の配列で利用した。ツールの利用時期は異なるものの、最終的にツールを利用した課題の内容・課題数は2011年度と2012年度は同じであった。ツール利用の四年目である2013年度は、前期前半で選択処理、繰り返し処理のアルゴリズムの思考法学習を3回取り入れ、後期の配列を利用したアルゴリズムではツールを利用しなかった。

### 3. 評価および考察

本節では、アルゴリズム的思考法に関する学習の導入による効果を評価するため、アルゴリズム的思考法の学習を取り入れる前の受講生と導入後の受講生との比較を行う。これを、アルゴリズム的思考能力、プログラミング能力、配列に関する理解の三つの視点から考察する。

また、Algo Toolを利用した受講者を対象としたアンケートよりツールの利用に関する評価を示す。

#### 3.1 アルゴリズム的思考に関する評価

アルゴリズムを記述する力が身についたか評価するために、アルゴリズムの記述実験を実施した。

以下に、記述実験の概要を示す。

- (1) 目的：アルゴリズムを記述する力を調査
- (2) 被験者：該当年度の「プログラミング言語I」の受講生全員（欠席者を除く）

(3) 時期：後期後半が始まる最初の授業にて実施（12月中旬）

(4) 時間：30分

(5) 内容：二つの問題（問題1、問題2）のアルゴリズムを日本語で記述

問題1から取りかかるものとしたため、問題2については時間がなく白紙や途中の解答が多かった。本論文では、問題1について報告する。問題1は、米Vertigo Softwareの開発者Jeff Atwood氏が自分のブログで紹介して[9]、話題を呼んだ、まったくプログラムを書けないのにプログラムの採用試験に応募してくる人が後を絶たないことに業を煮やした開発者が、ふり分けのために使っているというFizzBuzz問題を利用した。FizzBuzz問題は、「1から100までの数を順に出力せよ。ただし、その数が3で割り切れるときは“Fizz”という文字列を、5で割り切れるときは“Buzz”という文字列を、3と5のどちらでも割り切れるときは“FizzBuzz”という文字列をそれぞれ数の代わりに出力せよ」というものである。

記述されたアルゴリズムを大まかに、正解、一部不備を含む、不正解の3種類に分類した。アルゴリズム的思考法の学習を導入する前の2009年度を受講生とアルゴリズム的思考法の学習を取り入れた2010年度以降の受講生の結果を比較する。この結果を図11に示す。正解の割合は2009年度及び2013年度が高い結果となっているが、一部不備を

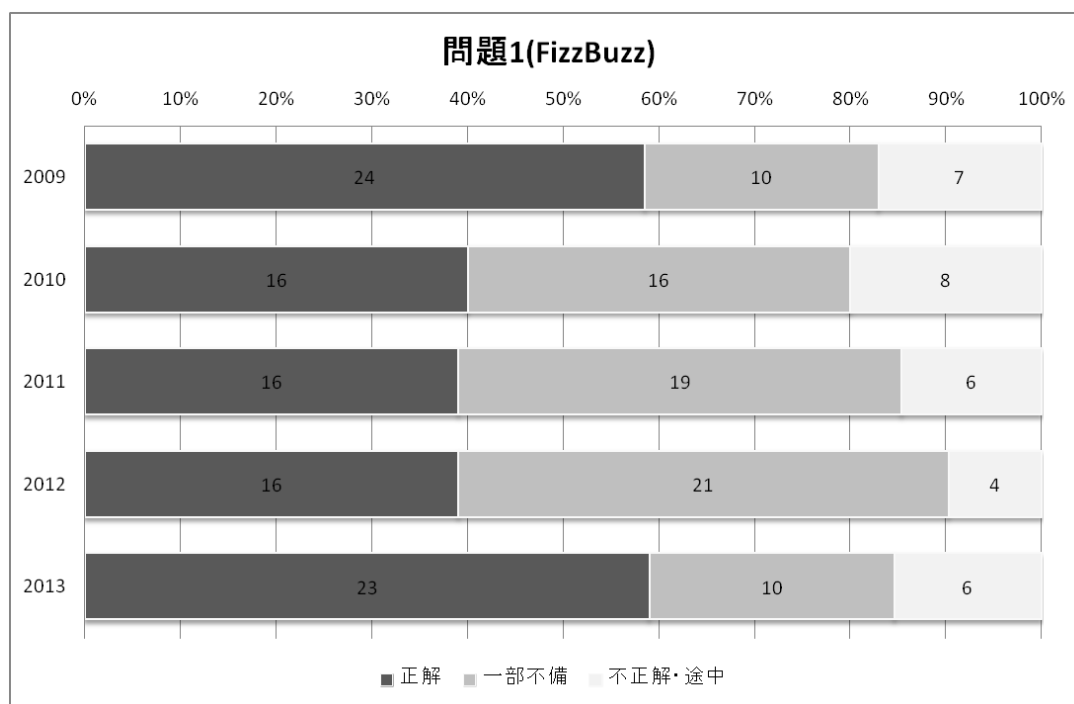


図11 FizzBuzz問題のアルゴリズム記述実験結果

含むが基本的な処理の手順は正解であるものを含めると、どの年度も大きな差はない。アルゴリズム的思考法の学習を取り入れた効果はみられなかったが、受講生の8割ほどがアルゴリズムを記述できていると言える。

不備なものとして多かったのが、(1) 無理に配列を宣言して使おうとするもの、(2) 変数の型の理解が不十分なもの(文字列と数値を同一変数に代入)、(3) 問題文の順序で処理を記述し、3と5両方の倍数の場合の処理に不備があるもの、の3種類であった。(1)に関しては、配列の学習を終えたところでの実験だったため、配列の問題だと思いこんで問題に取り組む受講生が多かったのではないと思われる。また(1)、(2)についてはプログラミング言語の知識・文法に依存した誤りで、(3)に関しては課題文の内容理解が正確でないためにおこった誤りといえる。

### 3.2 プログラミング能力についての評価

アルゴリズム的思考法に関する学習を取り入れることにより、プログラミング能力にも影響があるか考察する。ここではプログラミング能力を測る指標として定期試験の結果を利用し、アルゴリズム的思考法に関する学習を導入する前の受講生と導入後の受講生の試験結果を比較する。

「プログラミング言語I」の2008年度から2013年度までの定期試験の平均点を図12に示す。定期試験の実施時期および試験範囲はどの年度もほぼ同じとなっており、前期中間でコンピュータおよびプログラムの基礎・データの入出力・選択処理、前期期末で繰り返し処理、後期中間で配列、学年末でユーザ関数を中心となっている。難易度についても同等の間を出題している。2008, 2009年度はアルゴリズム的思考法に関する学習を導入していない。2010, 2011年度は配列の理解に焦点をあて後期の最初にアルゴリズム的思考法の学習を導入、2012年度はC言語の学習を始める前(前期の最初)と配列学習前(後期の最初)、2013年度はC言語の学習を始める前(前期の最初)のみ Algo Tool を利用しアルゴリズムに特化した授業を行った。

アルゴリズム的思考法の学習を後期に実施した2010年度, 2011年度では、学習を導入する前の2008年度, 2009年度と比較して前期では平均点が低い、後期では平均点が高くなっていることがわかる。また、アルゴリズム的思考法の学習をC言語の学習前に導入した2012年度, 2013年度では前期の平均点が学習導入前の過年度の平均点より高くな

っている。2013年度の後期中間試験において過年度より平均点が低くなっているのは、問題の方式を穴埋めからプログラムコード全てを記述する形式に変更したことによる影響が大きいと思われる。学年末ではアルゴリズム的思考法の学習を取り入れた2010年度から2013年度までの4年間全ての平均点が、学習導入前と比較して7点以上高い値となっている。平均点の差は、受講生の理解度だけでなく、問題の難易度にも左右されることから、平均点のアップがアルゴリズム的思考法の学習の導入による効果であるとは断定できないが、C言語のプログラミング能力の向上に一定の効果があったと考えられる。

また、前期中間試験において特定の問題(選択処理を含むプログラムシミュレーションの理解度を測る問題)について、2009年度~2013年度の得点率(問題の配点を100点満点として換算した場合の得点)の平均・分散を比較したところ、2010年度の得点率の平均が84.4点と一番高く、次いで前期からアルゴリズム的思考法に関する学習を取り入れた2012年度が82.9点、2013年度が78.0点と高い結果であった。しかしながら分散に関しては、2012年度が一番小さく、2013年度が一番大きくなっている。本授業は40人程度の規模であるため、試験結果が年度ごとの受講生のレベルにも影響を受けていることが予想される。

### 3.3 配列の理解についての評価

配列に関する理解度を評価するため、後期中間および学年末試験における同一の配列問題(配列問題01から配列問題04)の得点率を比較する。2009年度から2013年度までの配列問題の平均得点率の結果を図13に示す。なお、2013年度の後期中間試験では問題の穴埋め方式をやめ、試験問題を大幅に変

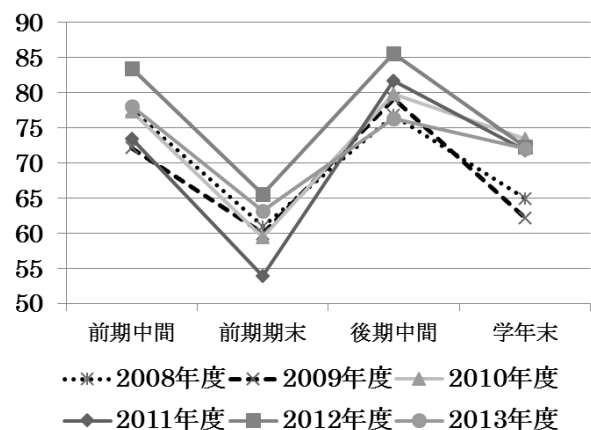


図12 2008年度~2013年度までの定期試験の平均点

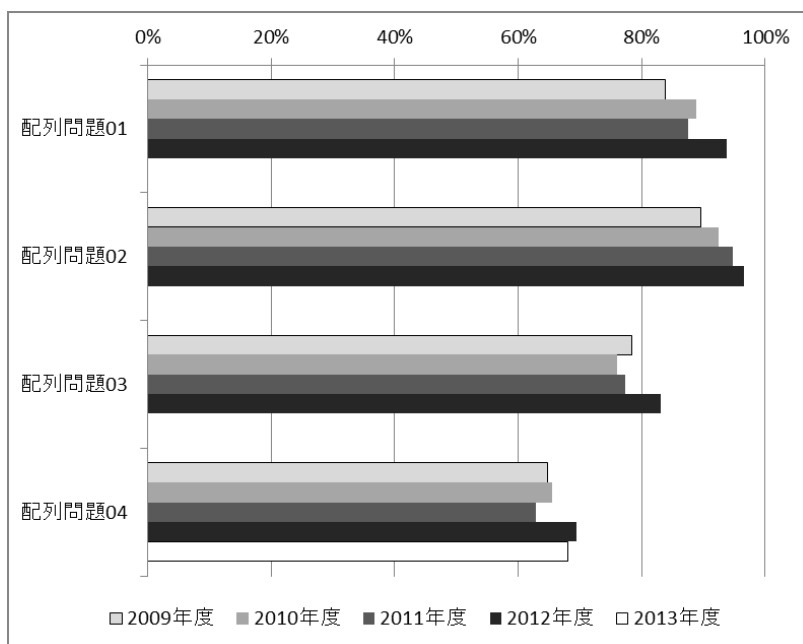


図 13 配列問題に関する得点率の比較 (2009 年度～2013 年度)

更したため、配列問題 01 から配列問題 03 の比較該当データがない。

一次元配列を使用する基本的な問題（配列問題 01, 配列問題 02）については、アルゴリズム的思考法に関する学習を導入した全ての年度で、導入していない 2009 年度の得点率を上回っている。しかしながら、有意水準 5% で検定したところ有意差は認められなかった。

C 言語を学習する前にアルゴリズム的思考法の学習を導入した 2012 年度が、4 題とも一番高い得点率となっている。また、2013 年度の後期中間試験における配列のプログラム作成においても、一次元配列の合計を求める問題の得点率が 83.3%（得点率 90% 以上 6 割）、二次元配列の 2 重ループの問題の得点率が 90.0%（得点率 90% 以上 5 割）と高得点であった。配列問題 04（ユーザ関数の作成・配列データの引き渡し・最大値）においても、C 言語を学習する前にアルゴリズム的思考法の学習を導入した 2012 年度、2013 年度の得点率が他の年度より高い結果を示している。

これらの結果より、プログラム言語を学ぶ初期段階でアルゴリズム的思考法に関する学習を導入し、アルゴリズム的思考能力を身につけてからプログラミングの授業に移行することで、C 言語の学習がより効果的に行われ、プログラミング能力の向上にも役立つと考えられる。

### 3.4 ツール利用に関する評価

2012 年度の受講生に対し、ツール利用に関する

アンケートを実施した。『ツールの利用がアルゴリズム的思考を身につけるのに役だったか』の間に対し、「役に立った」が 68%、「どちらでもない」が 25%、「役に立たなかった」が 7% であった。「役に立った」と回答した人の理由としては、

- ・文字ではなく、見た目でもわかりやすく理解しやすかった。ツールを使った後のプログラミングは楽にできた気がする。
- ・実際に C プログラムを書く前に頭の中でやる手順を準備することができる。そのため、プログラムがスムーズにかけるようになる。
- ・考える手順を理解できた。

などが挙げられ、アルゴリズム的思考を身につけるのに役立ったといえる。

また、「どちらでもない」「役に立たなかった」の理由としては、

- ・二度手間な感じがした。
- ・リストが 1 番から始まっていたり、C 言語と機能が異なる部分があって紛らわしい。
- ・ツールを利用してなくても、最初から C 言語で考えた方がわかりやすい。

などの意見があった。ツールの利用はプログラミングを始めて学ぶ受講生には好評であった一方、プログラム言語を授業以外で既に習ったことのある受講生の中には、ツールの利用を有効と感じない人もいることがわかった。アルゴリズム的思考が身につけてくれば、このようなツールのサポートを利用しなくても、アルゴリズムを思考できることが考えられる。



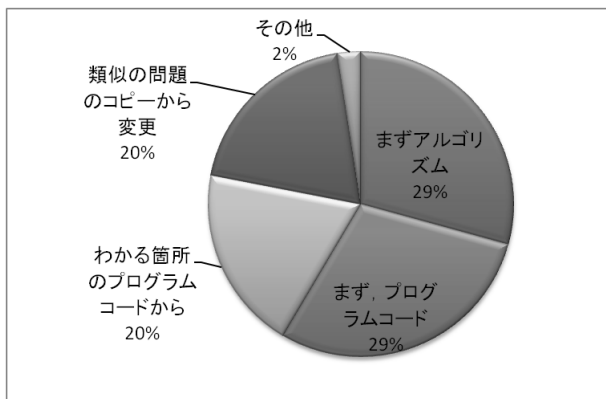


図 14 2012 年度プログラム作成手順

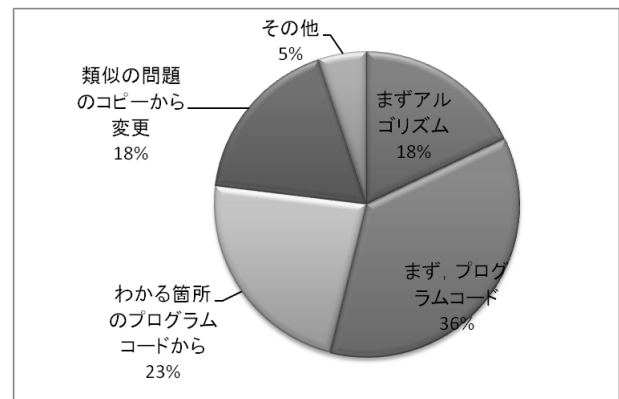


図 15 2013 年度プログラム作成手順

アルゴリズム的思考法の学習を導入する目的として、プログラム課題が与えられた際に、アルゴリズムを意識させ、試行錯誤的にプログラミングする受講生を減らしたいという狙いがあった。そこで、プログラム作成手順に関するアンケートを2012年度、2013年度に実施した。

『プログラムを作成する際にどのような手順で作成することが多いですか』の問に対して、「まずアルゴリズムを考える」「アルゴリズムを意識せず、プログラムコードの先頭から書き始める」「プログラムコードのわかる箇所から」、「類似問題のプログラムを書き写し、問題に合わせて変更していく」「その他」の5つの選択肢から最もあてはまるものを回答してもらった。2012年度、2013年度の受講生の結果をそれぞれ図14、図15に示す。「わかる箇所からプログラムコードを書き始める」「類似の問題等のプログラムコードを書き写し、問題に合わせて変更していく」という、試行錯誤しながらプログラムを作成していると思われる受講生が、2012年度、2013年度ともに4割いることが明らかとなった。これは、練習問題としてあえて類似の問題を課すことが多いこと、また、授業で扱う問題の規模が小さいために試行錯誤でもプログラムが完成する経験を積んでしまっていること等が要因として考えられる。

#### 4. おわりに

プログラミング教育にアルゴリズム的思考法に関する学習を取り入れることで、アルゴリズムを考える力を養い、プログラム作成に必要な十分なスキルを身につけさせることを目的とし、2010年度より授業実践を行っている。

アルゴリズムの記述実験からは、アルゴリズム的思考法の導入によるアルゴリズム的思考力の向上を検証することは出来なかった。しかしながら、プロ

グラミング能力、配列に関する理解度については、定期試験の結果の分析より導入前と比較して向上が見られた。これまでの4年間の実践から、アルゴリズム的思考法を学んでからプログラムの学習に入ること、C言語の文法理解もスムーズに進み、プログラミング能力の向上にも効果があると考えられる。

アルゴリズム的思考法の学習にツールを導入したメリットとして以下の三点が挙げられる。1) アイコンで示される構成要素を所定の場所にドラック&ドロップすることでアルゴリズムの構造が表現できるので、初心者を使いやすく、分かりやすい。これにより、C言語の細かな文法を気にすることなくアルゴリズムを思考できる。2) 上位の計画を記述してから、それを処理ごとに詳細化していくことで、全体のアルゴリズムを理解しやすい。3) ツールによるアルゴリズムの記述の検証が可能で、問題の提示・提出の労力も軽減されることにより、授業時間内により多くの問題を解かせることができ、効率よく学習できる。また、受講者自身で動作の確認ができることがアルゴリズムを考えるモチベーションにもつながっていた。

本論文では、プログラミング学習の初期段階でのアルゴリズム的思考法の導入がプログラミング学習において効果的であることを明らかにした。今後は、アルゴリズム的思考力と基礎プログラミング能力の関連性について考察を行いたい。さらには、アルゴリズム的思考法に関するより効果的な教育方法を検討するため、アルゴリズム的思考法で利用する課題の適切さに関する検討、および、初学者の誤答傾向の分析により学習者の理解状況に応じた課題の提示方法について検討していく予定である。

## 参考文献

- 1) 香山瑞恵, 永井孝, 國宗永佳, 不破泰, 萱津理佳, 山本樹: "アルゴリズム的思考法のための教育支援ツールの開発", 日本情報科教育学会学会誌, Vol.4, No.1, pp.75-76 (2010)
- 2) 香山瑞恵, 永井孝, 山本樹, 國宗永佳, 不破泰, 萱津理佳: "グラフィカルなインターフェースによるアルゴリズム的思考法教育支援の試み", 教育システム情報学会第35回全国大会論文集, 26-D1-04 (2010)
- 3) 永井孝, 香山瑞恵: "ビジュアル・ブロック・プログラミング可能なアルゴリズム学習向けWEBツール: Algo Tool", 日本情報科教育学科第5回全国大会講演論文集, p.49 (2012)
- 4) 不破泰, 國宗永佳, 香山瑞恵, 新村正明, 宮尾秀俊: "情報工学科学生に対するアルゴリズム的思考法教育手法の提案と実践", 教育システム情報学会研究報告, Vol.23, no.6 pp.34-41 (2009)
- 5) ランダ著, 駒林邦男, 宮坂瑠子, 土井捷三訳: "アルゴリズムの思考方法: その教授と学習", 明治図書, 東京 (1970)
- 6) 駒林邦男: "教授における数学的理論学の適用の試み: アルゴリズムの教授を中心として", 日本教育学会大曾研究発表要項, Vol.22, pp.30-31 (1963)
- 7) 飯田周作, 飯田千代, 清藤武暢, 佐藤創: "アルゴリズム的思考法の教育", 情報処理学会教育研究報告, 2008-CE-93, pp.57-64 (2008)
- 8) 村上和繁, 大隅敏明, 稲浦綾, 岩崎重剛, 松永公廣, 石桁正士, 横山宏: "アルゴリズム的思考法の指導", 電子情報学会研究報告, ET2008-36, pp.29-34 (2008)
- 9) raganwald2008, <http://weblog.raganwald.com/2007/01/dont-overthink-fizzbuzz.html>, (January 24, 2007)  
(平成 26 年 10 月 1 日受付、平成 26 年 11 月 28 日受理)