

グラフィックライブラリを組み込んだ ポータブル C 言語学習システム

正員 本間 達^{*a)}

Portable C-language Learning System with Graphics Library

Satoru Honma^{*a)}, Member

(2020 年 5 月 20 日受付, 2020 年 9 月 23 日再受付)

A learning system of C-Language that consists of some freewares such as C-language compiler, DOS emulators, some batch files and configuration files, has been newly proposed to learn introductory C-language. A non-official graphic library for C-language has been introduced to the system to motivate student programming. Two DOS emulators have also been introduced into the system to run old applications such as this compiler on some later versions of Windows. This system is effective to interest more students than learning only ANSI C-language standard. Furthermore the system is constructed on a portable storage medium to use the same programming environment inside and outside of the university. This system enables the students to study programming in the same procedure even if they use a various PCs with different equipment configurations. Thus, this system was also suggested to be practical in Active-Learning.

キーワード: 臨床検査技師, 医療情報学, プログラミング環境, グラフィックライブラリ

Keywords: medical technologist, medical information technology, programming environment, graphic library

1. はじめに

臨床検査技師国家資格試験の受験科目の一つである医用工学概論は、臨床検査技師が取り扱う様々な機器の基礎知識を念頭に置いて出題される科目である。この科目は、具体的には、医用工学・情報科学・検査機器学の 3 科目で構成されている。上記の国家資格を受験するためには厚生労働省の認定する臨床検査技師の養成校で、所定の単位を修得する必要がある、多くの養成校はこれらの 3 科目を、受験資格を取得するための必修科目として、学校ごとのカリキュラムを設定している。

情報科学の講義では、医療情報を取り扱うことを前提として、そのための道具であるコンピュータやネットワークに関する基礎知識を習得後、医療情報システムの基礎を学習する。この科目は講義のみならず実習も必修科目として

設定されるが、実習の内容について全国共通のカリキュラムが存在せず、コンピュータの利用を共通項として、教員の判断で大学ごとに異なるカリキュラムを設定する。一部の臨床検査技師養成校では、一般的なワープロや表計算ソフトなどのアプリケーションソフト、あるいは医療分野の研究でしばしば利用する統計処理ソフトの操作を習得する実習を情報科学の実習と位置付けている。

一方、電気回路について学ぶ医用工学と関連づけ、医用機器の動作制御プログラムについての理解を深めることを目指して、プログラミングの実習をおこなう大学もある。これは、臨床検査技師が医用機器の開発を自らおこなうことを前提とするものではないが、最先端の研究を企業などと連携するとき、共有すべき基礎知識の一つとして習得することが期待されている。一方、多くの医学系の学生にとって、プログラミングの技術は必ずしも必須の技術ではないので、興味を持たない学生は単位取得のみを目指す傾向がある。このため、プログラミングの実習において、学生の興味を引き出せる学習環境の構築が必要である。

大学の实習時間として設定される 30 コマ程度の学習時間で一通り学習可能で、機器制御¹⁾に適する入門用のプログ

a) Correspondence to: Satoru Honma. E-mail: hommtec@tmd.ac.jp

* 東京医科歯科大学大学院医歯学総合研究科

〒113-8519 東京都文京区湯島 1-5-45

Graduate School of Medical and Dental Sciences, Tokyo Medical and Dental University

1-5-45, Yushima, Bunkyo-ku, Tokyo 113-8519, Japan

ラム言語の一つであり、また C++ や Perl, PHP などの近年活用されているプログラム言語と一部の文法が共通するという観点から、C 言語は大学での実習に適している。

プログラミング初心者の興味を引き出すために、グラフィック機能を活用するのが有効である⁽²⁾ことが示されている。しかし ANSI (American National Standards Institute)⁽³⁾が制定し、ISO (International Organization for Standardization) が継承した C 言語の規格にはグラフィック機能が定義されていない。それにもかかわらず、一部のコンパイラでは独自ライブラリが用意されており、あるいはフリーのライブラリを組み込むなどの手続きでグラフィック機能を利用できる場合もある。岡田らは、グラフィックの描画を可能にした C 言語の学習システムを構築し、実際の授業で用いて文系学生の興味を引き出したことを報告している⁽⁴⁾。このシステムでは、メッセージ通信を用いて、Windows の機能にアクセスし、グラフィックを描画するので、描画速度が遅いという問題点がある。田村らは、Web ブラウザベースの学習環境で、タートルグラフィックス風のグラフィック機能を備えた C 言語の学習を可能にしている⁽⁵⁾。このシステムは一定の機能を備えたブラウザ上で動作するので学習機器を選ばない利点がある。しかし、例えば「始点と終点を指定して線を描く」を「始点を指定する」+「終点に移動しながら線を描く」のように、グラフィックの描画に必要な動作を分解して記述する必要があるため、医学系の学生などの初心者にとっては簡単に利用できると言い難い。

エル・エス・アイ ジャパン社は 1993 年に一部機能を制限した LSI-C コンパイラ試食版⁽⁶⁾をフリー配布している。また、これに対応したフリー版のグラフィックライブラリ⁽⁷⁾が一部のプログラマによって作成され、インターネット上で公開されている。このグラフィックライブラリは、グラフィックの機能ごとに一つの関数を設定しているので、基本的な C 言語関数と同様の学習で手軽に利用できる利点がある。この 2 つの組み合わせにより、C 言語でグラフィックを利用可能で、小容量の実習用システムを安価に構築可能である。本研究ではこれを基本に、大学外での自主学習でも大学と同一の環境を継続して利用可能とするポータブル C 言語学習システムを構築し、その実用性について検討する。

2. フリーウェアを活用した C 言語学習環境の構築

〈2・1〉 Windows 上で動作する C 言語学習環境 現在市販されている多くの PC で、Windows の様々なバージョンが、OS として一般的に使用されている。Windows 環境で使用可能な C 言語開発環境として、Microsoft 社の Visual C++ など、研究開発にも使用可能な高機能の C/C++ コンパイラがインターネット経由で無料提供されている。これらを用いて C 言語の学習を行うことは可能であるが、C 言語の学習に使用しない C++ コンパイラを内包するので、必要な記憶媒体の容量が大きく、インストールなど利用準備に時間がかかる。また、実習時間を考慮すれば、大学の情報検索

室で PC を入れ替える数年ごとに、この作業をおこなうのは非効率的である。

ところで、グラフィック機能を利用できる環境がプログラミング学習のモチベーションを高めることが示唆されている⁽²⁾⁽⁴⁾が、ANSI で定められた C 言語⁽³⁾はグラフィック機能を規定していないので、Windows 環境のプログラム開発環境でグラフィックを利用する場合、C++ コンパイラにより Windows 固有の API (Application Programming Interface) や Direct X などの外部関数を利用してプログラムを作成する必要がある。さらに、家庭での自主学習を念頭に置くと、学生らが自分でこの環境を導入する必要があり、医学部の学生にとっては、実習向きであるとは言いがたい。一方、複数の関数を簡便に使用するために独自関数にまとめた、いわゆるラッパーと呼ばれる各種のライブラリがインターネットを経由して配布されている。これらの利用により、比較的容易に Windows アプリケーションが作成できるようになるが、C 言語の基本関数をほとんど使用しないので、基本的な C 言語によるプログラミングを習得する段階で使用するのは適切とは言い難い。また、いわゆるラッパーは、上述した Visual C++ などのコンパイル環境を利用するので、家庭での実習環境として、医学部の学生が導入するのは容易ではない。

これらの観点から、一部を除いて専門的にプログラムを作成しない医学部の学生が、限定された実習時間内で C 言語を学習するための環境として、Windows 上で動作するコンパイラを利用することは、最善の選択とは言い難い。

〈2・2〉 C 言語学習環境の理想条件 大学では情報検索室などで、学生一人一人に利用可能な PC が用意され、全員共通の環境で実習をおこなうことができる。一方、医学部情報委員会の方針により、全学生が年度ごとに提示される推奨 PC、もしくは同等のスペックを持つノート型 PC を所持しているとされるので、これを利用して実習に使用することも可能である。さらに、進行状況に応じて学生が実習時間外に家庭などで自主的な学習をおこなう可能性を考慮すれば、利用する PC のスペックは多種多様と考えなければならない。

上述の点を考慮して、本学の情報科学実習で用いるプログラム言語の学習システムは、(1) PC 本体から取り外し可能な記憶媒体上に実体が存在し、(2) 環境変数などをバッチファイルなどで設定して利用可能であれば、異なる PC 環境でも対応可能である。また、不特定多数が利用することを考慮すれば (3) ライセンスフリーのフリーウェアと OS 付属のアプリケーションのみで実習に必要な環境を構築可能であり、さらにスペックが低い PC でも安定動作するために、(4) 動作に必要な最小スペックができるだけ低いこと、および (5) システム全体の容量が小さいこと、が望ましい。

〈2・3〉 C 言語学習システムの構成 上述の条件を考慮して、Windows 以前の OS 環境で使用されていたコンパイラを利用し、取り外し可能な記憶媒体に、学習環境を構築する。このためのコンパイラとして LSI-C コンパイラ試食版

を使用する。これは 1993 年 8 月に公開されたエル・エス・アイ ジャパン社製のフリーウェアで、製品版の機能を一部制限したものであり、現在でもインターネット経由で入手可能である。300 行以内のソースファイルをコンパイル可能であり、ANSI で規定されている C 言語の関数を実装するので、入門用として適している。さらに、これと組み合わせることを前提として開発された、非公式のグラフィックライブラリ⁽⁷⁾がフリーウェアとして提供されており、この利用により直線やだ円、長方形などの基本図形を描画する機能が利用できる。これらを利用して作成した 16bit アプリケーションは、Windows XP 以前の 32bit OS に付属のコマンドプロンプト上では制限なく動作するので、グラフや任意の CG を描画するなどの実習課題に対応可能である。

しかしながら 16bit の MS-DOS 用アプリケーションである LSI-C コンパイラ試食版およびこのコンパイラで作成するアプリケーションは、64bit 対応の Windows では、OS 自体の機能制限により、動作しない。また、グラフィックライブラリを使用したアプリケーションは、PC の一部の制御コードに制限がかけられており、32bit 対応の Windows であっても VISTA 以降のバージョンでは動作しない。

一方、64bit の Windows 上で 16bit のアプリケーションを動作するために様々な DOS エミュレータが開発され、インターネット上でフリー配布されている。その一つである MS-DOS Player⁽⁸⁾は、特に LSI-C コンパイラ試食版を念頭に置いて開発されたエミュレータであり、LSI-C コンパイラおよびこのコンパイラで作成するアプリケーションを動作することが可能である。このエミュレータ上で、グラフィック機能を利用することはできないが、グラフィックライブラリを利用するソースコードをコンパイルすることは可能である。一方、事前にコンパイルされたアプリケーションについて、グラフィック機能を利用可能なエミュレータも提供されている。その一つである DOSBOX⁽⁹⁾は C 言語のコンパイルは対応しないが、グラフィック機能を実装したプログラムを実行することが可能である。

LSI-C コンパイラ試食版およびグラフィックライブラリを、これら 2 つのエミュレータを組み合わせると、64bit OS に対応する、グラフィック利用可能な C 言語学習システムを構築することができる。上述した 2 つのエミュレータを含めた場合、システムを構成するファイルの合計容量は約 85MB であるので、記憶媒体には、学生が作成するソースコードの保存領域を含めて 100MB 程度の容量が必要である。すなわち、本システムは移動運用可能な記憶媒体上に構築可能である。本システムが利用する DOS エミュレータについて、Windows のバージョンとグラフィック利用の観点からの分類を Fig.1 に示す。

〈2・4〉 USB メモリを用いた学習システムの構築

利用が簡便で運搬が容易な外部記憶装置の一つとして、いわゆる USB メモリがある。大部分の USB メモリは上述のシステムを構築し、C 言語の実習としてプログラムを作成するのに十分な記憶容量を利用可能である。一方、USB

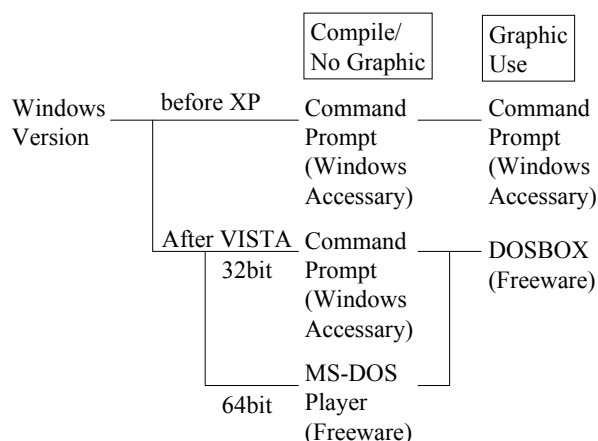


Fig. 1. Selection of emulators by the versions of Windows.

メモリは利用する PC ごとにドライブレターが異なる場合がある。大学の情報検索室内で使用する PC は、すべてが同一のドライブ構成なので、USB メモリのドライブレターを限定できるが、自宅など大学外部の PC で使用すると、学生ごとに PC 環境が異なるので、これに対応しなければならない。

LSI-C コンパイラ試食版では、利用するコンパイラ、リンカ、ライブラリのフォルダを環境変数内に絶対位置として記録しており、ドライブレターが異なると利用できない。また、DOSBOX も同様に、実行するプログラムのフォルダ名を環境変数内に記録しており、このフォルダ以外での利用は困難である。そこで、ドライブレターごとにこれらの格納場所を記録した示す環境変数をあらかじめ用意し、学習開始時に利用者がバッチファイルで差し替え、任意のドライブでの利用を可能にする。

〈2・5〉 学習システムの利用手順

学生がソースコードのコンパイル時に Windows のバージョンを識別して、必要に応じて MS-DOS Player を使用することは困難であるので、自動的にバージョンをチェックして、使用を自動的に選択するバッチファイルを作成する。学生は、これを利用して一定の操作手順でコンパイルをおこなうことができる。コンパイルして作成される実行ファイル (アプリケーション) は 16bit アプリケーションであるが、グラフィックライブラリを使用しない場合は、コンパイラと同様に MS-DOS Player を使用すれば実行可能である。そこでコンパイルから連続して自動的にこれを用いて、プログラムを実行するようにした。一方、グラフィックライブラリ使用時には、学生自身がその利用を認識している。したがって、コンパイルにより作成された実行ファイル (アプリケーション) を試行する際には、DOSBOX を学生自身が利用するように、手順をマニュアル化した。この流れを Fig.2 に、またシステムを利用している様子を Fig.3 に示す。

〈2・6〉 独自実習テキストの作成

C 言語の文法を順序良く学習できるように構成した文法解説書を、本システムによる実習のために作成した。基本的な ANSI-C 準拠の文法を学ぶ 7 章と、外部ライブラリに依存するグラフィック

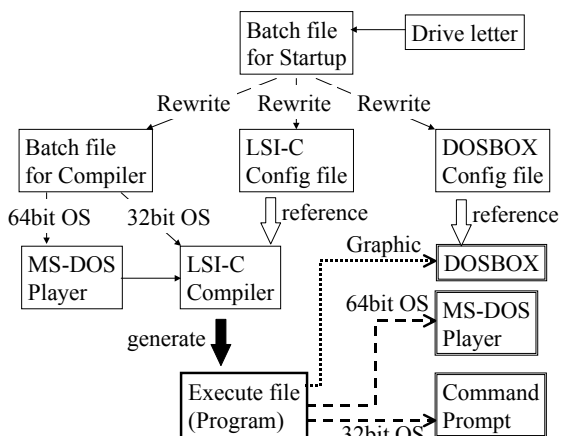


Fig. 2. Procedure for using C-language learning system.

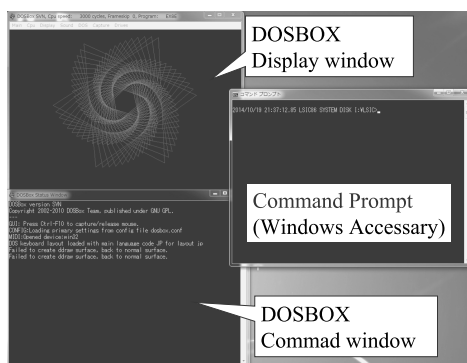


Fig. 3. Overview of using the system.

Table 1. Contents of the textbook.

Chapter	Theme of the chapter	Number of sample
1	Character output	7
2	Character input	6
3	Conditional branch	10
4	Loop	16
5	Function	11
6	Pointers	12
7	Structure	3
8	Graphics	15
9	Charts	8
10	Arithmetic calculation	9
11	Interface	3

機能を用いる 3 章, および D/A・A/D コンバータをソフトウェアで制御する手法に関する 1 章の全 11 章で構成した。この概要と章ごとに設定したサンプルプログラムの数を Table 1 に示す。また付録として MS-DOS で頻用する 13 個のコマンドを解説した。これは、実習中にファイル操作を CUI (Character User Interface) でおこなうことを念頭におき、かつ C 言語で作成するアプリケーションが、MS-DOS の外部コマンド形式であることを学生が理解するためである。構築した C 言語学習環境を利用して実行可能なサンプル

プログラムのソースコードを、各章ごとに示してから、そのプログラムに必要な文法を解説する構成とし、全体で 100 のサンプルを掲載した。

各章ごとに章末問題を提示して章ごとの力試しが可能としたが、これについては解答を示さず、学生自身の自由な発想で取り組むようにした。また、実習評価のために複数の課題を巻末に提示した。

この実習書は学生がサンプルのソースコードをコピーペーストせず、自身で入力して確認することを促すために、内容を抽出不可とした PDF ファイルで作成した。これを学生のみアクセス可能なサーバもしくはメール添付にて学生に配布して、学生の質問などに応じて内容を修正した場合にも、直ちに他の学生が共有可能とした。

〈2・7〉 学生自身の手による USB システムの構築

本システムを構成するコンパイラなどのアプリケーションはフリーウェアであり、その配布はインターネット上から入手できる状態のままであることが条件とされている。実習では、実習書と合わせて、これらのフリーウェアを入手できる URL と、これらを接続するバッチファイルおよび定義ファイルを学生に配布し、これらを用いて USB ドライブ上に本システムを構築する手順を指導する。このため、学生個人ごとに本システムを構成可能である。一方、実習の中で、利用方法を説明するタイミングを揃えることが必要な場合には、上述の指導をおこなったうえで、教員があらかじめ構成したシステムを配布すると、時間配分を計画することが容易になる。

〈2・8〉 ポータブル実習システムを利用した実習の概要

医学部におけるプログラミング実習の目的の一つは、医療機器の動作制御プログラムに対する理解を深めることである。全ての学生が制御プログラムを作成できるレベルに到達する必要はないので、医学系大学におけるプログラミング実習の最低要件は、ソースコードの構築からコンパイルまでの流れを理解し、実践できることである。しかし、与えられたサンプルコードを用いて一連の流れをなぞるだけでは、プログラミングの習得とは言い難いので、適切な課題を設定して、新規性のあるアプリケーションを完成させるのが望ましい。この前提として、課題を解決するための論理を構築できる数学力を有することが必要であり、さらに論理に基づいてコードを正しく組み上げることができるよう、プログラム言語の文法習得が必要である。実習では、解説書のサンプルプログラムを参照しながら文法を習得し、最終的に与えられた課題から数個を選択して、独自のプログラム構築を目指す。必ずしも実習の時間内で全学生がこれを達成できるとは限らないので、自宅学習など時間外におこなう自主学習が重要である。

実習では、初回にあらかじめ作成したサンプルコードを示して、ソースコードの作成とコンパイル、および作成したアプリケーションの実行までの、本システムの基本的な手順を学生に指導する。続いて、テキストエディタを用いて、解説書に記載したサンプルコードを自ら入力し、実行

して結果を確認するという作業を通じて、C 言語の関数を一つずつ習得するように誘導する。学生の適性により、全員が同一のペースで進行できるものではないので、進行については学生の判断に委ね、教員および TA (Teaching Assistant) は文法の理解あるいはトラブルの対処など、学生からの問い合わせに対してのみ対応する。

解説書のサンプルを入力するだけの作業にならないように、課題の作成を意識して進行するように指導する。実習期間内に 2 つ以上の課題を提出できることを到達基準として提示し、これを達成できるように実習の時間配分を考慮しながら必要な単元を学生自身が判断する。

C 言語のプログラムはインターネット上にサンプルが豊富に存在し、あるいは上の学年の学生が同様の課題を経験しているので、一部の課題はこれらを手入してアレンジするだけで容易に完成する。そこで最終課題の提出時には口頭試問をおこない、内容を理解していないと判断される場合には提出を認めないことを明示する。さらに、口頭試問後に提出された課題をチェックして、類似性が特に高く剽窃と判断されるものは削除することを明示している。

3. ポータブル C 言語学習システムの評価

〈3・1〉 システムの動作に関する評価 本システムの稼働試験として、Windows 2000, XP, VISTA, 7, 8.1, 10 で動作を確認した。Windows 7 までの OS では 32bit での動作、Windows 7 以降の OS では 64bit での動作を確認した。すなわち Windows 7 については 32bit, 64bit のいずれも動作確認をおこなった。

いずれのバージョンにおいても一定の操作で、ソースファイルをコンパイルし、作成した実行ファイルの動作が可能であることを確認した。学生は学習開始時に、システムのドライブレターを認識して、対応するパッチファイルで初期設定を行うが、それ以降は Windows の対応 bit 数の違いなどを自動判別するようにしたので、それを特に意識する必要はない。すなわち、大学や家庭などコンピュータの構成や OS のバージョンが異なる環境でも、同一の操作でプログラミングの学習が可能である。さらに、グラフィック利用における DOSBOX の操作も環境によらず一定であることを確認した。

〈3・2〉 学生の利用から見た多様な課題への対応状況

2015 年度から 2020 年度までにおこなった本システムを用いた実習で、提出された全課題の提出状況を Table 2 および Fig.4, Fig.5 に示す。なお、2020 年度は特に、通常の前学期間に相当する 4 月から 7 月前半まで学生が登校することなく自宅で実習および課題の作成をおこない、7 月後半に課題提出と口頭試問を対面で実施した。2015 年度から 2019 年度までも、学外での自主学習を前提としていたが、2020 年度については、この度合いが一層大きくなった。なお、2020 年度は、類似性が高い提出が 7 件あり、口頭試問後のチェックによりこれを削除している。

Table 2 に示したのは年度ごとの履修学生数と提出された

Table 2. Number of students and assignments submission by year.

year	number of student	number of assignments submission	
		no graphic	with graphic
2015	35	111	17
2016	35	104	20
2017	33	62	19
2018	34	63	16
2019	38	76	17
2020	33	77	26

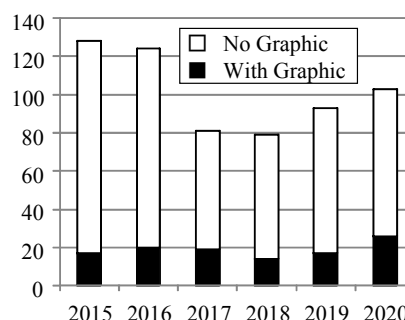


Fig. 4. Contrast between graphic and non-graphic assignments submission by year.

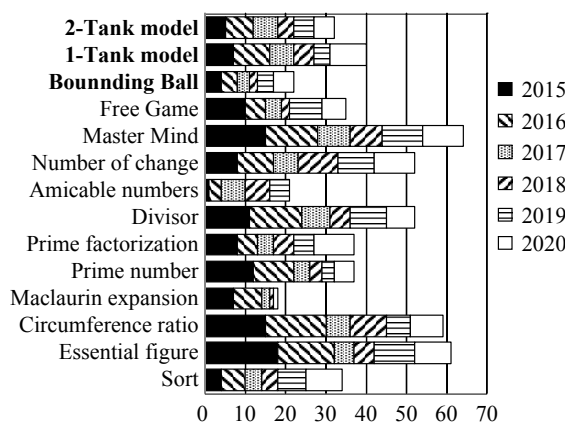


Fig. 5. Distribution of assignments submission by year.

課題のうちグラフィック関数の利用の有無の内訳である。Fig.4 は提出課題のうちグラフィック関数の使用状況を示している。図中黒塗りで示したのが、グラフィック関数を使用する課題の提出数である。2016 年度以前と比較すると、2017 年度以降の総提出数が減少したことが確認できる。一方、グラフィック関数を使用する課題は若干の差はあるものの、毎年同程度の数が提出されている。学生がランダムに課題を選択していれば、提出数の減少に応じてこの提出数も減少すると考えられる。なお、自主学習が主体となった 2020 年度は全体の提出数がやや増加し、これに伴ってグラフィック関数を使用する課題提出も増加している。しかしながら、グラフィック課題の提出割合について χ^2 検定を

おこなったところ、年度ごとの有意差がないことを確認した。したがって、年度ごとに一定割合の割合でグラフィック課題が選択されていることが示された。

Fig.5 に示すのは、同年度における主な最終課題の提出数である。25 の最終課題を用意しているが、5 年間の総提出数が 10 に満たない課題はグラフに示していない。図中に示した課題のうちグラフィック関数を使用するのは「タンクモデルの水位」(2-Tank model, 1-Tank model) と、「ボールの落下運動」(Bounding Ball) である。年度ごとの提出数は同一課題でも異なっている。この結果について一元配置の分散分析をおこない、課題ごとの有意差 ($p < 0.05$) を確認したので、学生の選択には有意な差があると判断できる。Fig.4 の結果と合わせ、円周率 (Circumference ratio), 素数 (Prime number), 約数 (Divisor), マクローリン展開 (Maclaurin expansion) などをテーマとする、一部の課題について提出数が有意に減少したことが分かる。

4. 考 察

本研究では LSI-C コンパイラ試食版⁽⁶⁾とグラフィック関数ライブラリ⁽⁷⁾および 2 種類の DOS エミュレータ⁽⁸⁾⁽⁹⁾を組み合わせ、最新の Windows にも対応可能で、グラフィックを使用可能なポータブル C 言語開発環境を構築した。このシステムでは各フリーウェアを相補的に組み合わせ、C 言語のプログラミング経験がない学生の興味を引き出しうるグラフィック機能の利用⁽²⁾⁽⁴⁾を、単純化した一定の操作で実現可能とした。さらに、初見の学生でもシステムの構築が可能なように作業手順をマニュアル化し、フリーウェアの配布条件を満たしつつ、実習での使用を可能にした。このシステムは有料のアプリケーションを利用していないので、学生自身が USB メモリなどの記憶媒体を用意すれば、実習費用を大幅に低減することができる。このため導入は容易である。また、本システムの利用で、学生は大学のみならず家庭などでも実習をおこなうことができる。すなわちこのシステムを用いてアクティブラーニングを実施することが可能であると考えられる。

Windows の機能を利用してグラフィック機能を実現するコンパイラは、インストールされるシステムの容量が大きく、導入に時間がかかる。さらに C++ の機能を利用するので、初学者には難易度が高い。本システム導入以前の、医学部における C 言語のプログラミング実習で、C++ コンパイラを利用した場合、プログラムの実行結果を表示するのに必要なウィンドウを 1 つ作成するのに、規定の実習期間全てを要し、C 言語固有の文法を用いたプログラミングの学習に至るのは困難であった⁽¹⁰⁾。これはウィンドウを作成する関数が、C 言語の文法で定義されていない OS 固有の API 関数であるので、C 言語の文法を勉強しても対応できないことによると考えられる。C++ コンパイラでは、C 言語の機能、C++ で追加された機能、OS が提供する API 関数のいずれも利用可能であるため、初学者にはその区別が困難である。これは、プログラム言語の学習において、コンパ

イラと OS の機能がそれぞれ独立し、勉強する範囲を明確にできることが必要であることを示唆している。Windows のアプリケーションであるコマンドプロンプトや DOSBOX⁽⁹⁾を利用する本システムは、ウィンドウの作成をこれらのアプリケーションに委ねているので、C 言語の文法学習のみに集中することが可能になった。これは、初学者のプログラミング実習を容易にするものである。

本システムで導入したグラフィックライブラリの関数自体は、ユーザーが定義した関数として ANSI 準拠の C 言語の関数と同様に利用できるため、C 言語の文法習得と区別する必要はない。しかしながらグラフィック機能を使用するには、ANSI 準拠の文法に加えて、別章として構成されているグラフィック関数およびその応用例を学習しなければならない。このシステムを用いた実習で、グラフィック機能を利用した課題が毎年一定程度提出されていることは、学習内容が増加しても、グラフィック機能を使用することを選択する学生が、一定数存在することを示していると考えられる。

C 言語の ANSI 規格は数年おきに改定を繰り返しており、OS やハードウェアの改良に伴って、新しい機能が追加されている。1993 年に公開されたコンパイラを基本とする本システムは、それ以降に追加された C 言語の新機能に対応していない。しかし、本システムで利用できる基本的な機能は最新の規格でも利用可能であり、プログラミングの基礎学習として必要な内容をサポートしていると考えられる。

なお、プログラミング実習では、ソースコードの頻繁な上書き、およびコンパイルごとに中間ファイルの作成と消去がおこなわれるので、USB メモリの記憶領域に対する書きこみ回数が増加する。また、しばしば抜き差しする USB メモリが物理的に破損する場合がある。このため記憶媒体ごとの定期的なバックアップを学生には推奨している。本システムでは、上述した記憶容量を確保できれば、外付けの HDD や SD メモリなども利用可能である。携帯性、記憶容量、価格等を総合的に判断して記憶媒体に USB メモリが適切と判断しているが、利用する学生自身の判断で、記憶媒体を任意に変更することは可能である。

2020 年度は、文法の学習から課題作成まで、大部分が学外でおこなわれたので、学生がインターネット上に掲載されているサンプルコードを参照する度合いが多くなった。その結果、提出された課題の一部に、同一のコードを参照したと判断される類似性の高い提出が確認されたので、削除の対象とした。2015 年度から 2019 年度においても、インターネット上のサンプル参照はおこなわれていたと考えられるが、ほぼ同一と判断せざるを得ない提出は、確認されたことはなかった。家庭での自修学習の割合が大きくなった結果、学生間の情報交換が少なくなり、同一性の高い提出を避けるという判断が減少したと考えられる。

2016 年度以前と 2017 年度から 2019 年度は特に、同一のシステムを使用しているにもかかわらず、課題の総提出数に差を生じている。これは、検定の結果から、数学の知識

に関連する内容の課題が有意に減少したことによると判断出来る。上述の観点から、2017年度から2019年度は、学生間で同一性の高い課題提出を避けるために、アルゴリズムの変更やソースコード記述の工夫ではなく、単位取得可能な水準まで提出数を減らすという判断をしていた可能性が考えられる。なお、グラフィック関連の課題は、中学校の教育課程水準の数学的な課題と比較して、ソースコードの記述量が増加するので、作成に必要な時間は多くなると考えられる。また、他者と同一にならないようにプログラムの内容を工夫したり、提出課題の選択のために文献を調査することも、学生の学習になっていると考えられる。このため、課題の提出数を減少しても、学外での学習時間の総計は一定程度維持されている可能性が考えられる。

プログラム作成のために数学の手順を理解してアルゴリズムを工夫することが減少したのは、大学入学以前の数学教育と関連している可能性が考えられるが、明確にこれを判断するデータはなく、今後の研究課題である。

5. おわりに

本研究では、フリーウェアとして配布されている C 言語コンパイラとグラフィックライブラリおよび DOS エミュレータを組み合わせ、持ち運び可能な記憶媒体上でのプログラミング学習用のシステムを開発した。これらの利用には一切ライセンス料などが不要であり、実習経費を低減可能である。このシステムは、Windows の API 関数や、C++ コンパイラなどによらず、単純に C 言語の文法のみを学習することに特化している。また C 言語の文法に沿う形で組み込まれたグラフィック関数も利用可能で、多様な課題に対応可能なシステムであり、かつ、Windows の各バージョンで同一の操作可能な点で、長期間利用可能な実習環境である。基本的なプログラミングの実習で利用するのに十分な機能を有する点で、プログラミング実習への実用性が示された。

大学だけでなく家庭でも同一環境でのプログラミング実習ができる本システムは、アクティブラーニングにも対応可能であることが示唆された。さらに C 言語の規格で規定されていないグラフィック機能を実装した結果、年度ごとに一定程度の割合でグラフィック課題が選択されていることが示された。

プログラミングの習得を主たる目的としない医学部学生などの初学者でも、本システムを用いて実習可能であることが示されたので、中学校の教育課程で導入されたプログラミングの演習にも利用可能であると考えられ、今後の活用が期待できる。

本論文の要旨は 2012 年度第 3 回電気学会教育フロンティア研究会⁽¹⁰⁾および 2016 年度第 3 回電気学会教育フロンティア研究会⁽¹¹⁾で発表した。

文 献

- (1) 後閑哲也：C 言語による PIC プログラミング大全，技術評論社，東京 (2018)
- (2) Y. Hiratsuka and T. Fikui : "Effect of a Graphics Library that C Language Programming Education is Helped", Bull. Mukogawa Women's Univ., Nat. Sci., Vol.44, pp.75-80 (1996) (in Japanese)
平塚由花子・福井哲夫：「C 言語プログラミング教育を支援するグラフィックライブラリーの効果」，武庫川女子大紀要 (自然科学)，Vol.44, pp.75-80 (1996)
- (3) D. Kernighan and B. W. Ritchie : THE C PROGRAMMING LANGUAGE SECOND EDITION, Prentice Hall, London (1988)
- (4) K. Okada, Y. Murai, H. Yaysumi, et al. : "Simple graphic display system for C language beginners education", Forum on Information Technology, pp.337-338 (N-017) (2005) (in Japanese)
岡田行平・村井保之・巽 久行，他：「簡易グラフィック表示システムを用いた C 言語入門教育」，FIT2005 (第 4 回情報科学技術フォーラム) 講演論文集，pp.337-338 (N-017) (2005)
- (5) 田村謙次・鳥居隆司・中野健秀，他：「グラフィック出力を備えた Web ブラウザベースのプログラミング実行環境」，PC カンファレンス論文集，pp.20-23 (2014)
- (6) エル・エス・アイ ジャパン (株)：「LSI C-86 v3.30c 試食版」，<http://www.vector.co.jp/soft/maker/lsi/se001169.html>
- (7) 小山佳孝：「LSI-C 試食版用 簡易グラフィックライブラリ 1.093」，<http://www.vector.co.jp/soft/dos/prog/se243629.html>
- (8) 武田俊也：「MS-DOS Player for Win32-x64」，<http://homepage3.nifty.com/takeda-toshiya/msdos/>
- (9) GNU General Public License : 「DOSBOX」，<http://www.dosbox.com/>
- (10) S. Honma and H. Wakamatsu : "Programming exercise using the C-Language compiler on the removable storage medium", The Papers of Technical Meeting on Frontiers in Education, IEE Japan, FIE-13-003, pp.13-16 (2013) (in Japanese)
本間 達・若松秀俊：「持ち運び可能な C 言語学習環境を用いたプログラミング実習」，電気学会教育フロンティア研資，FIE-13-003, pp.13-16 (2013)
- (11) S. Honma and H. Wakamatsu : "Portable C-language learning system with graphics library for programming exercise", The Papers of Technical Meeting on Frontiers in Education, FIE-17-002, pp.7-10 (2017) (in Japanese)
本間 達：「プログラミング実習のためのグラフィックを使用可能なポータブル C 言語学習システム」，電気学会教育フロンティア研資，FIE-17-002, pp.7-10 (2017)

本 間 達 (正員) 1969 年 3 月 26 日生。1993 年日本大学理工学部卒業。1997 年東京医科歯科大医学部卒業。2002 年同大学大学院保健衛生学研究科博士後期課程修了。2003 年東京医科歯科大学大学院保健衛生学研究科助手。2004 年同大学大学院同研究科助教，現在に至る。博士 (保健学)。

