

Web を利用した C 言語学習における入力方法の提案

2013SE037 広瀬千咲 2013SE072 加藤利沙 2013SE106 町野麗音

指導教員：蜂巢吉成

1 はじめに

情報系の大学などで行われているプログラミング学習において、初学者への入門言語として、実用的な手続き型言語である C 言語を学ばせることが多い。C 言語学習の現状として、学習者は端末を使用し、コマンドラインでの学習環境でプログラミング学習を行っている。しかし、学習者が普段コンピュータを利用する際、端末で操作することは少ないので、コマンドラインでのプログラミング学習は、普段のコンピュータ利用環境とギャップを感じてしまい、プログラミング学習への障壁となる恐れがある。大学の講義や演習のように多くの学習者が同時にプログラミングを行う場合は、Web を用いて学習者のプログラミング状況などをリアルタイムに把握する研究などが行われており、Web によるプログラミング学習環境の発展が期待される。コマンドラインでの学習に普段とのギャップを感じてしまう学習者でも、Web ならば普段の利用環境と近く、ギャップを感じることなくプログラミング学習ができる。

しかし、従来の C 言語学習の入力はコマンドラインを前提とした対話的なストリーム入力であるのに対して、Web の入力は処理に必要なデータを一度に送信する一括入力である。従来の C 言語学習の入力は Web に適しておらず、Web のプログラミング環境の入力は単一の入力欄にすべてのデータを入力する方法となりわかりにくい。

本研究では Web に適した C 言語学習における入力方法の概念を整理し、そのための学習環境や関数を提案することを目的とする。C 言語学習における入力モデルを従来のストリームから連想配列とストリームの組み合わせに変更する。それに合わせた入力インタフェースや入力関数を提案し、C 言語学習の学習環境を作成する。提案した入力インタフェースと入力関数を用いて、実際に教科書に記載されているプログラム例を書き直し、本学習環境でコンパイル・実行できるかを評価した。

2 C 言語入力方法の調査

我々が調べた限り Web を用いた C 言語学習の入力方法に関する研究やツールがなかったので、教科書の調査と Web の学習サイトの調査を行った。

2.1 教科書における入力方法の調査

現在の C 言語学習において、入力がどのように学習されているかを調査するために次の 3 つの観点から、教科書 [4] に記載されている全単元のプログラム例 (185 例) を分析した。

1. 入力先
2. 入力関数と使用方法

3. 入力されるデータ

1 は、標準入力とファイルでの入力があった。
2 は、scanf 関数と getchar 関数が使用されており、ほとんどが scanf 関数であった。ファイルへの入力には fscanf 関数、fgetc 関数、fread 関数が使用されていた。

従来の教科書は端末での対話的な入出力を前提としている。どのような値を入力するかを促す printf 関数による出力があり、それに従って scanf 関数で値を入力する。複数の値を入力する場合もこの繰り返しで、入力はストリームとして扱われる。

従来の学習で多く使われている scanf 関数では変数へ値を格納するためにポインタ型が使われるが、学習の初期では変数名の前には&演算子をつけることを「おまじない」として教えることが多い。手続き型言語では、代入により変数の値を書き換えながら計算が進むが、scanf 関数は代入の形式 (変数名=式) をしておらず、学習者の理解に繋がりにくい。

3 は、入力されるデータを型の種類で分類すると int 型が 123 回、double 型が 11 回、char 型の配列 (文字列) が 13 回、unsigned 型が 5 回使用されていた。入力されるデータをデータ数の観点から分類すると、常に一定回数の場合と、実行時に入力個数が変わる場合があった。個数が変わる場合は、最初にデータ数を入力してその個数分のデータを入力する場合や、負の数を入力するまでのようにある条件になるまで繰り返し入力する場合、“Yes...1/No...0” のようにメニューのように使う場合などである。入力される変数の観点から分類すると、単一の変数に 1 回だけ入力する場合と、単一の変数に複数回入力する場合、配列の各要素に 1 回だけ入力する場合があった。

2.2 Web の C 言語学習サイトの調査

Ideone[1]、paiza.IO[2] などの Web のプログラミング環境はオンライン上でプログラムを実行することができる。学習者による環境構築が不要なことや、コンパイル・実行を一つのボタンで行えることが特徴である。

Web における通信はステートレスであり、対話的に入出力を行うのは困難なので、処理に必要なデータを一括して入力する。そこでは入力を促す出力は不要である (対話的に入力をしないので意味をなさない)。既存の Web のプログラミング環境は単一の入力欄にデータを記述する方法であり、複数のデータを入力する際に、データがどのような意味を持つのかがわかりづらい。

Web プログラミング環境ではコンパイルと実行が同時に行われることも多い。ソースコードを直接書き換えて変数に代入する値を変更するのと、単一入力欄の値を書き換

えるのとて、作業的には大きな違いはなく、プログラムの入力概念が理解しづらい。paiza learning[3]でのC言語学習での入力を調査した結果、scanf関数は一度も使用されず、ソースコード内で直接代入または入力ではないrand関数を使用することで変数への値の入力を行っていた。

3 WebによるC言語学習用入力方法の提案

2節の調査を踏まえてWebに適した入力インタフェースと入力関数を提案する。

Webの実行モデルはキーと値(パリュウ)を一对とし、必要な入力データをすべて用意してから送信している。Webの入力モデルのキーと値を一对として扱う考え方は連想配列の考え方に近い。よって、Web上での入力をストリームモデルから連想配列のモデルに変更することを提案する。一つのキーに対して複数の値を扱う場合は、値をリストとし、ストリームで読み込むことにする。

3.1 前提

本研究では、学習者をプログラミングに対しての初学者とする。手続き型プログラミング言語において学習すべき基本的なことを、構造化プログラミングの接続、分岐、反復、基本型と演算、配列・構造体などのデータ構造、関数・手続き、ファイルの入出力とする。

3.2 設計指針

入力方法を考えるにあたり、次の3点を設計指針とする。

1. Webらしさ
2. 分かりやすさ
3. 学習用であること

1は「Webの入力として自然であること」を表す。本研究は学習環境によって生じる学習者のプログラミング学習への壁をなくしたいと考えているので、2は学習者にとって使いやすく、操作性が高いことを表す。3はC言語学習において、概念が理解しやすいことや、学習の障壁にならないようにすることが重要である。

3.3 Webでの入力インタフェース

入力形式はフォームを使用し、その中でもテキストボックスとテキストエリアを使用する。提案する学習環境は、学習者のプログラムに記述された入力関数をコンパイル・実行前に解析して入力フォームを生成する。

Web上の入力形式は、フォームとダイアログボックスがある。フォームは自分が入力した値を確認することができるが、ダイアログボックスは複数入力するとき、ひとつずつ入力し、そのつど消えてしまうので、後に自分が何を入力したかを確認できなくなり、指針3に反する。

学習時における入力場面は2.1節の3より、単一の変数に1回だけ入力する場合、単一の変数に複数回入力する場合、配列の各要素に1回だけ入力する場合、メニューのように入力する場合がある。これらの場合に適応するフォームはテキストボックス、テキストエリア、ラジオボタン、

セレクトボックスである。しかし、フォームの種類が多いと複雑化し、学習者の混乱の要因となり、指針2に適さない。学習において使用される回数の少ないメニューの入力は変数1つの入力のみならず、テキストボックスとテキストエリアのみを扱うこととする。

テキストボックスは、1つの値を変数に代入するとき使用する。1つの変数に対して1つのテキストボックスを作り、連想配列として扱う。テキストエリアは、入力する値を単一の変数に複数回入力する場合や配列の各要素に1回入力する場合に使用する。このとき、値をリストとして扱う。

入力フォームの生成方法について、Webの特性に合わせて値をプログラム実行前に入力しなくてはならないので、入力フォームはプログラムの実行と独立した方法で生成する。

作成方法としてHTMLなどの専用言語を使用し記述する方法、インタフェースビルダのようにGUIを用いて作成する方法、学習者が記述したプログラム内の入力関数によって入力フォームを生成する方法が挙げられる。しかしプログラム外に記述する方法はC言語学習外のことを考えなくてはならないので学習者の負担になり、指針3に反する。

3.4 C言語関数の設計

2.1節の2の分析結果の通り、現在のC言語学習で使用されている入力関数のほとんどをscanf関数が占めている。しかしscanf関数は指導者にとって好ましくない関数である。scanf関数の代わりとして入力関数の役割とフォーム生成の役割をもつ関数を提案する。

3.3節で述べたようにフォームの種類は2種類である。値が1つの場合はテキストボックスを生成する関数、不定個数または配列等、値の個数が複数である場合はテキストエリアを生成する関数を用意する。配列用の関数を作り、一括代入させる方法も考えられるが配列の理解に繋がらないので、指針3に反する。

C言語は静的型付け言語なので型ごとに入力を行う必要がある。教科書[4]で多く使用されているのは2.1節の3の分析結果からint型、double型、char型の配列である。しかし文字列を文字の配列とする概念はC言語特有なものである。また2.1節の2の分析でgetchar関数によって一文字ずつ入力する問題は少ない。プログラミング言語の型の概念を学習するために文字列はchar*をtypedefしてstring型として用意し使用させる。

以上より次の関数を提案する。getInt関数、getDouble関数、getString関数は単一の変数に1回だけ入力するソースコード1のような場合に使用し、テキストボックスを生成する。nextInt関数、nextDouble関数、nextString関数は単一の変数に複数回入力または配列に入力するソースコード2のような場合に使用し、テキストエリアを生成する。hasNext関数は不定個数の入力があった場合に値があ

るか判断する。fgetInt 関数, fgetDouble 関数, fgetString 関数はファイル入力において使用する。

ソースコード 1 テキストボックス

```
1 int a, b, c;
2 a = getInt("a");
3 b = getInt("b");
4 c = getInt("c");
```

ソースコード 2 テキストエリア

```
1 int a[3], i;
2 for (i=0; i<3; i++) {
3     a[i] = nextInt("a");
4 }
```

getTYPE 関数と nextTYPE 関数の引数は入力フォームのラベルを表している (TYPE は Int, Double, String のいずれかを表す)。代入の分かりやすさから変数名を記述することを想定している。戻り値はフォームに入力された値をそれぞれの型で返す。fgetTYPE 関数はファイルポインタを引数とし、ファイルから型に合う中身をひとつずつ取り出して返す。fgetTYPE 関数はこれまで定義したような Web による入力関数と同様にファイルから入力できるような関数として設計した。nextTYPE 関数, hasNext 関数はテキストエリアに入力するものが複数の値を入力するときであるという性質上、リストになった値をストリームで読み込むよう設計した。

使用例をソースコード 3 に示す。このように関数を使用することによって図 1 のようにフォームが生成される。図 1 ではフォーム生成後、入力データを記述している。

ソースコード 3 プログラム例

```
1 int m, n, a[10][10], i, j;
2 m = getInt("m");
3 n = getInt("n");
4 for (i=0; i<m; i++) {
5     for (j=0; j<n; j++) {
6         a[i][j] = nextInt("a");
7     }
```

図 1 フォーム生成

4 学習環境の設計と実現

学習者が Web にあった入力を行うために必要となる学習環境は、ソースプログラムの編集、コンパイル、実行、ファイル管理の学習機能に加え、フォーム生成を行う機能が必要である。

多くの統合開発環境ではコンパイルと実行をひとつのボタンで行うことができたが、一括で行うとコンパイルの概念を理解しにくいので、本研究ではコンパイルと実行を分けるような設計にする。学習者の C 言語プログラムは、サーバ上でコンパイル、実行を行うことにする。コンパイル・実行は、JavaScript に変換しクライアント側で実行する方法も考えられる。本研究では C 言語プログラムを安定して実用的に使うことができる従来型のコンパイラを使用する方法を用いる。

フォーム生成、コンパイル、実行はサーバで処理を行うために、Ajax を利用する。実行時に JSON を使用しフォームに入力した値をキーと一対になるようなデータ型でサーバに送って実行を行う。

学習環境の画面例を図 2 に示す。

フォームボタンを押すとソースプログラムを解析し、変数入力欄にフォームが生成される。フォーム生成、コンパイル、実行の機能を分けており、プログラムの入力関数を変更しなければ、一度生成したフォームを再利用できる。プログラムを変更し、再コンパイルした場合でも、以前にフォームに入力した値を再度使用することができる。異なる入力値で実行する場合には、コンパイルをせずに実行できればよい。

図 2 学習環境の画面例

表 1 入力の種類ごとによる評価

	問題数	実現可能	実現可能でない
テキストボックス	78	78	0
テキストエリア (配列)	11	11	0
テキストエリア (メニュー)	7	0	7
ファイル	2	2	0

5 提案した入力方法の評価

5.1 評価方法

3.1 節で述べた学習項目に該当する教科書 [4] の問題 (計 98 問) を, 提案した入力インタフェースと入力関数を用いて書き直し, 本学習環境でコンパイル・実行できるか確認した。さらに, 入力を促す出力文を除いて教科書と同じ実行結果が得られるかを確認した。同じ実行結果が得られた場合を「実現可能」とした。

5.2 評価結果

98 問中 91 問実現可能であった。入力の種類ごとによる評価を表 1 に示す。入力が 1 つである場合, 入力が複数回である場合, ファイル入力である場合に, それぞれ使用する関数を使い分けることで, フォームを生成し, 教科書のプログラムを書き直して実現することができた。

実現可能でない 7 問は実行中の入力によって処理が変わるプログラムである。プログラム実行中に値の入力ができないので, 値が入力されていないと判断され, エラー処理によりプログラムが停止してしまう。プログラム実行前に終了するまでの一連の値をテキストエリアに入力するとプログラムの実行自体は可能だが, 学習としては不適切であると考えられる。このような対話型プログラムに関しては 6.2 節で考察する。

6 考察

6.1 Web での出力

Web であることを利用し学習のために出力を工夫することで学習者の学習効率を上げられると考えた。出力の種類として, 標準出力と標準エラー出力がある。本研究での学習環境で, 出力の種類を区別するために表示する時のテキストを標準出力のときは黒色, 標準エラー出力の時は赤色で表示した, 端末での学習の場合, 標準エラー出力は標準出力と同じように出力されるので, 違いが分かりにくい。表示の仕方を変えることにより出力の概念の理解につながると考えられる。

6.2 対話型プログラムの入出力

本研究では Web の実行モデルに合わせてデータは一括送信の形態をとっているため, 対話的に行うプログラムに対応していない。しかし, 例題数が少ないとはいえ, 対話的なプログラムは, 外部の入力によって処理が変わるプログラムの学習に役立つ。

Web で対話的に実行するプログラムの入出力を図 3 の

ようにスマートフォンのメッセージアプリのように左側に出力, 右側に入力を表示することで学習者が理解しやすいと考えた。これを実現するには, 学習者のプログラムを C 言語から JavaScript に変換することで実行時の環境をクライアント側へ変える。クライアント側で入力と実行を行うことでプログラム実行中の入力にも対応し, 対話型プログラムの入出力を実現できる。このような入出力方法を実現すると手続き型言語の逐次処理していく概念や入出力の概念が理解しやすくなると考えた。



図 3 対話型プログラムの入出力例

7 おわりに

本研究では Web に適した C 言語学習における入力方法を提案した。入力モデルを従来のストリームから連想配列モデルとストリームの組合せに変更し, フォームを使用し学習環境を実現した。実際に教科書に記載されているプログラム例を提案した入力インタフェースと入力関数を用いて書き直し, 本学習環境で実現できるか評価した。

今後の課題は, Web に適した学習用の出力を検討することや, 対話型入出力の実現, 本研究で作成した学習環境等が学習者にとって使いやすいものであり学習環境によって生じるプログラミング学習への障壁を低くするものであるかを評価することである。

参考文献

- [1] SphereResearchLabs : Ideone.com, <<http://ideone.com/>> .
- [2] ギノ株式会社 : paiza.IO, <<https://paiza.io/>> .
- [3] ギノ株式会社 : paiza learning, <<https://paiza.jp/works/c/primer/>> .
- [4] 柴田望洋 : 新版 明解 C 言語 入門編, ソフトバンククリエイティブ株式会社 (2012).