短冊型問題: プログラミングの技能を評価可能な試験出題形式

角田 博保[1,a)]　久野 靖[2]

概要：プログラミングの学習成果を評価する方法として試験は最善のものではないが、さまざまな事情から試験形式での評価が必要になることは多い。従来はその際の形式として、「穴埋め問題」と「自由記述(プログラムを全部書かせる)」の 2 つが主流であったが、前者は「対策」によって正解できてしまう (プログラミングの能力を測れていないことがある)、後者は採点が難しいという問題があった。筆者らは情報入試研究会の活動の一環として、プログラムの 1 行ずつを選択肢として用意し、回答者が必要な行を自由に選択して並べることでコードを完成させる「短冊型問題」を開発し、模擬試験で評価してきた。本論文ではこれまでの出題例や成績データをもとに、短冊型がプログラミング技能をどの程度測れているかについて議論する。

1. はじめに

初中等教育においてプログラミング教育が盛んになってきたが、プログラミングの学習成果を評価することは重要である。筆者らは情報入試研究会 [1] において、過去 4 回大学情報入試全国模試に関わっている。

情報入試研究会は「高校における情報教育の達成度合いを正しく評価し，また情報教育に対する適切な指針を提供する上で，関係者が共に認める，適正な範囲・内容・水準を持った試験問題・試験方式を構築」するために「具体的な入試問題の試作を行い世の中に公開」することを目標に 2012 年 3 月設立された。

第 1 回は 2013 年 5 月に 90 分の試験時間で行い、受験者は 80 人であった。出題形式は「多肢選択式」および「記述式」（20〜50 文字）であった。

第 2 回以降は 90 分の試験を 45 分× 2 に分割し、以下の構成となっている。

問題		選択方法	配点	分野
セット A	第 1 問	必答	30 点	共通
	第 2 問	必答	35 点	情報の科学
	第 3 問	必答	35 点	社会と情報
セット B	第 1 問	必答	30 点	共通
	第 2 問	必答	35 点	情報の科学
	第 3 問	必答	35 点	社会と情報

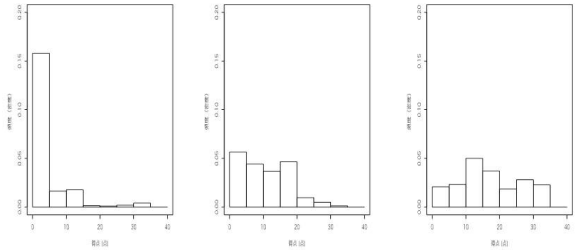
第 2 回〜第 4 回ともセット A の第 2 問がプログラミング問題である。また、その得点に関する統計量は以下の通りであった。

	第 2 回	第 3 回	第 4 回
対象者数	859	1943	649
満点	35	35	35
平均点	3.3	10.8	17.2
標準偏差	6.89	7.42	9.50
最小値	0	0	0
第 1 四分位数	0	4	11
第 2 四分位数	0	10	17
第 3 四分位数	3	16	26
最大値	35	35	35

グラフ化すると以下の通りである。

[1] 電気通信大学非常勤講師
[2] 電気通信大学
The University of Electro-Communications, Chofu, Tokyo 182–8585, Japan
[a)] kakuda@acm.org

結果　セットA「科学」プログラミング



対象者数 859人　　対象者数 1948人　　対象者数 649人

(なお、詳しくは文献 [2] を参照されたい。)

これを見てわかるように、得点は低いが、回を追う毎に平均点があがっており、問題の出題方法の調整が良好に進んだ結果といえる。

以下、出題した問題と解答状況を順に、細かく見て行くとともに、短冊型問題の特徴について考察する。

2. プログラミング試験形式

プログラミングの能力を測るには、実際にコンピュータを使い、プログラムを作って実行することを観察するのが一番ではあるが、紙の試験による評価が必要となる場合も多い。ここでは、紙の上での試験という形での出題形式を考える。

紙の上でプログラミング能力を試験する形式としては、従来より穴埋め型と記述型がある。

穴埋め型は穴の空いたプログラムを示し、プログラムが目的を達するように穴を埋める形式である。与えられた選択肢から選ぶことになる。

- 利点：採点が簡単。
- 欠点：プログラムの構造を問うような問題は作れない。

記述型はプログラムの一部を自由に書かせるもので、回答者がまぐれで高得点をとるということがない。実力が測定できる。

- 利点：正確に回答者の評価ができる。
- 欠点：採点が大変。

人数が少なければ、記述型で良いが、多人数になると採点の負担、公平性から記述型は厳しくなる。しかし、穴埋め型では、偶然正解になることを避けられず、回答者の能力を正しく評価できるかが怪しくなる。そこで、短冊型の出題形式を開発した。

短冊型では、プログラムの 1 行ずつを選択肢として用意 (短冊) し、回答者が必要な行を自由に選択して並べることでコードを完成させる。

以下に例を示す。

以下の解答群の行を必要なものだけ並べて次のプログラムを作成せよ。解答群にある行は何回使っても良い。

問1「正の整数nを入力したとき、nを超えない奇数の合計を出力する」プログラム

問 1

解答群

ア n に整数値を入力
イ $j \leftarrow 0$
ウ $k \leftarrow 1$
エ $k \leftarrow i$
オ $k \leftarrow i \times i$
カ $j \leftarrow j + i$
キ $j \leftarrow j + k$
ク i を 1 から n まで 1 ずつ増やしながら、くり返し
ケ j を 1 から k まで 1 ずつ増やしながら、くり返し
コ ここまでが「くり返し」の範囲
サ もし、i が偶数ならば
シ もし、i が偶数でないならば
ス ここまでが「もし」の範囲
セ i を出力
ソ j を出力

2.1 短冊型問題の特徴

短冊型の問題の特徴は以下の通りである。

- 使える構文を既定できる
  たとえば、短冊を制限すれば、構造的プログラミングを強要できる。
- 語句レベルのエラーは起きない
  スペルミスが発生しない。
- 自由度が大きい
  穴埋めと違い、考えないと解けない。

以下、第 2 回〜第 4 回の模擬試験でのプログラミング問題について順に見て行く。

3. 第 2 回模擬試験の問題

ごく普通の整数計算のプログラムである。配点は次のようになっている。

問 1：　1 重のループ　　15 点
問 2：　2 重のループ＋ if 文　　20 点

問題は以下の通り。

第2回模擬試験 #003 セットA 第2問

問1 「正の整数 n を入力したとき、1から n までの数それぞれの2乗の和を出力する」プログラムを書け。(たとえば4を入力したら、$1+4+9+16=30$ なので、30と出力する。)(15点)

問2 正の整数 n を入力したとき、1から n までの数を順番に、奇数については1回、偶数についてはその数の回数だけくり返して、出力するプログラムを書け。(たとえば4を入力したら、「1 2 2 3 4 4 4 4」と出力する。)(20点)

解答群1

ア. n に整数値を入力	ケ. j を1から k まで1ずつ増やしながら、くり返し
イ. $j \leftarrow 0$	コ. ここまでが「くり返し」の範囲
ウ. $k \leftarrow 1$	サ. もし、i が偶数ならば
エ. $k \leftarrow i$	シ. もし、i が偶数でないならば
オ. $k \leftarrow i \times i$	ス. ここまでが「もし」の範囲
カ. $j \leftarrow j+i$	セ. i を出力
キ. $j \leftarrow j+k$	ソ. j を出力
ク. i を1から n まで1ずつ増やしながら、くり返し	

3.1 問1の採点基準と正解

正解は「アイクオキコソ」か「イアクオキコソ」である。

ア	n に整数値を入力
イ	j ← 0
ク	i を 1 から n まで 1 ずつ増やしながら、くり返し
オ	k ← i × i
キ	j ← j + k
コ	ここまでが「くり返し」の範囲
ソ	j を出力

採点基準としては、初期設定の「イ」を抜かすと 2 点減点、「コソ」が「ソコ」となってるのは 2 点減点、1 つ余分な行が入っていると 3 点減点とした。回答を調べて、採点基準を決めた。

全部で 859 件ある回答の種類数は 342 であった。

(1) 繰り返しがあるもの 212
(2) 繰り返すものがあるもの 184
(3) 不要な構文を使わないもの 104
(4) `k ← i × i` を使うもの 98
(5) 繰返し構文多重使用を排除 91
(6) 入力、出力がないものを排除 75

これで 75 種類となった。その件数は 321 件である。このように、短冊型では種類数が非常に多くなる。偶然正解になることはまずない。

平均で 2.4 点/15 点 (100 点満点換算 16 点) である。859 人中 満点は 47 人、0 点は 618 人であった。

3.2 問2の採点基準と正解

2 通りの正解群がある。

(1) k に繰り返し回数を設定し、繰り返す方法

ア	n に整数値を入力
ク	i を 1 から n まで 1 ずつ増やしながら、くり返し
★	ここに k の設定部分が入る
ケ	j を 1 から k まで 1 ずつ増やしながら、くり返し
セ	i を出力
コ	ここまでが「くり返し」の範囲
コ	ここまでが「くり返し」の範囲

k を設定する場所★には以下の 4 通りが入りうる。

(1.a)

ウ	k ← 1
サ	もし、i が偶数ならば
エ	k ← i
ス	ここまでが「もし」の範囲

(1.b)

エ	k ← i
シ	もし、i が偶数でないならば
ウ	k ← 1
ス	ここまでが「もし」の範囲

(1.c)

サ	もし、i が偶数ならば
エ	k ← i
ス	ここまでが「もし」の範囲
シ	もし、i が偶数でないならば
ウ	k ← 1
ス	ここまでが「もし」の範囲

(1.d)

シ	もし、i が偶数でないならば
ウ	k ← 1
ス	ここまでが「もし」の範囲
サ	もし、i が偶数ならば
エ	k ← i
ス	ここまでが「もし」の範囲

(2) 奇数、偶数ごとに出力する方法

ア	n に整数値を
ク	i を 1 から n まで 1 ずつ増やしながら、くり返し
★	ここに奇数ごとか偶数ごとに印刷する部分が入る
コ	ここまでが「くり返し」の範囲

★の箇所に以下の 2 通りが入りうる。

(2.a) 奇数、偶数ごとに印刷：

(エ)	k ← i ●
シ	もし、i が偶数でないならば
セ	i を出力
ス	ここまでが「もし」の範囲
(エ)	k ← i ●
サ	もし、i が偶数ならば
(エ)	k ← i ●
ケ	j を 1 から k まで 1 ずつ増やしながら、くり返し
セ	i を出力
コ	ここまでが「くり返し」の範囲
ス	ここまでが「もし」の

「エ」は●の 3 箇所のどこかになる。

(2.b) 偶数、奇数ごとに印刷：

(エ)	k ← i ●
サ	もし、i が偶数ならば
(エ)	k ← i ●
ケ	j を 1 から k まで 1 ずつ増やしながら、くり返し
セ	i を出力
コ	ここまでが「くり返し」の範囲
ス	ここまでが「もし」の範囲
シ	もし、i が偶数でないならば
セ	i を出力
ス	ここまでが「もし」の範囲

「エ」は●の 2 箇所のどこかになる。

構文の形により、何種類も正解がでてくる。採点基準では、多数ある正解からの 1 つのずれに対して 3 点減点している。

平均で 0.8 点/20 点　(100 点満点換算 4 点) であり、859 人中 満点は 14 人、0 点は 791 人であった。2 重のループになると正解率が非常に落ちることがわかった。

4. 第 3 回模擬試験の問題

第 3 回のプログラミング問題は図形を出力させる問題である。短冊だけだと難しすぎるので、プログラムを導入するような問を追加した。問 2 にプログラム設計の穴埋め問題を加えた。短冊型は問 1、問 3、問 4 である。配点は次のようになっている。

問 1：	1 重ループ	10 点
問 2：	プログラムの設計 (穴埋め)	8 点
問 3：	2 重ループ	10 点
問 4：	より複雑な 2 重ループ	7 点

問題は以下の通り。

図 1 は図 2 のような図形文字の列を表示するプログラムである。

```
n ← 5
i を 1 から n まで 1 ずつ増やしながら、くり返し
    "■"を表示する
ここまでが「くり返し」の範囲
改行する
"■"を表示する
"□"を表示する
"☆"を表示する
改行する
```

図 1　プログラム例

```
■■■■■
■□☆
```

図 2　表示例

問 1　次の動作をするプログラムを解答群 1 の選択肢を使って作成せよ。

"□"を 30 個表示した後に"■"を 18 個表示し、最後に改行する

解答群 1 の行を必要なものだけ並べて実現すること。なお、解答群 1 にある行は何回使ってもよい。

問 2　次の文章を読み、(1)～(4)に入るものを解答群 2 の選択肢から選んで、記号で答えよ。

文字を使って、三角形を 2 つ組み合わせた幅 $n+1$ 文字、高さ n 行の図形を表示することを考える。n が 4 の場合、次のような図形を描きたい。

□□□□■
□□□■■
□□■■■
□■■■■

そのためには、1 行目は"□"を 4 個表示したあとで、"■"を 1 個表示して改行する。3 行目では"□"を (1) 個表示したあとで、"■"を (2) 個表示して改行する。

これを一般化すると、i 行目 ($1 \leqq i \leqq n$) では、"□"を (3) 個表示したあとで、"■"を (4) 個表示して改行することになる。

解答群 2

第 3 回模擬試験 #004 セット A 第 2 問

問 3　n に整数を入力し、三角形を 2 つ組み合わせた以下のような幅 $n+1$ 文字、高さ n 行の図形を描くプログラムを書け。(10 点)

n が 4 の場合

□□□□■
□□□■■
□□■■■
□■■■■

問 4　n に整数を入力し、次のような三角形を 3 つ組み合わせた図形を描くプログラムを書け。(7 点)

n が 3 の場合

□■■■■■☆
□□■■■☆☆☆
□□□■☆☆☆☆☆

解答群 1

ア. n に整数を入力する	サ. $p \leftarrow p-2$
イ. $n \leftarrow 5$	シ. $q \leftarrow 1$
ウ. $k \leftarrow n-i$	ス. $q \leftarrow q+2$
エ. $k \leftarrow n-i+1$	セ. i を 1 から n まで 1 ずつ増やしながら、くり返し
オ. $k \leftarrow 30$	ソ. j を 1 から k まで 1 ずつ増やしながら、くり返し
カ. $k \leftarrow 18$	タ. ここまでが「くり返し」の範囲
キ. $k \leftarrow i$	チ. "□"を表示する
ク. $k \leftarrow p$	ツ. "■"を表示する
ケ. $k \leftarrow q$	テ. "☆"を表示する
コ. $p \leftarrow 2 \times n-1$	ト. 改行する

4.1 問 1 の採点基準と正解

問 1 の採点基準は以下の通りである。

- 解答を、1 文字の置換、1 文字の追加、1 文字の削除によって正解に変換できる場合の、置換数、追加数、削除数の合計が 3 件までは、1 誤りあたり 2 点減点とする。
- ただし「タ」を追加することによって正解に変換できる箇所（「タ」が抜けてる場合）は、まとめて 1 件と数える。
- 先頭に無駄な代入文があるものは減点しない。たとえば「エオ」と始まるものは、減点しない。

問 1 の正解は以下の通りである。

```
オ | k ← 30
ソ | j を 1 から k まで 1 ずつ増やしながら、くり返し
チ |   ”□”を表示する
タ | ここまでが「くり返し」の範囲
カ | k ← 18
ソ | j を 1 から k まで 1 ずつ増やしながら、くり返し
ツ |   ”■”を表示する
タ | ここまでが「くり返し」の範囲
ト | 改行する
```

平均は 5.3 点/10 点　(100 点満点換算 53 点) であった。1179 人中 満点が 459 人、8 点が 114 人、6 点が 91 人、4 点が 47 人、0 点が 468 人であった。単純なループが連続してでてくる簡単な問題であった。

4.2 問 3 の正解

正解は以下の通りである。

```
ア | n に整数を入力する
セ | i を 1 から n まで 1 ずつ増やしながら、くり返し
エ |   k ← n - i + 1
ソ |   j を 1 から k まで 1 ずつ増やしながら、くり返し
チ |     ”□”を表示する
タ |   ここまでが「くり返し」の範囲
キ |   k ← i
ソ |   j を 1 から k まで 1 ずつ増やしながら、くり返し
ツ |     ”■”を表示する
タ |   ここまでが「くり返し」の範囲
ト |   改行する
タ | ここまでが「くり返し」の範囲
```

採点基準は問 1 と同じである。

平均 0.8 点/10 点　(100 点満点換算 8 点) であった。1180 人中、満点は 39 人、8 点が 16 人、6 点が 27 人、4 点が 47 人で、0 点が 1051 人であった。2 重のループになると正解率が下がる。

4.3 問 4 の採点基準と正解

問 4 は正解が多数ある。代入文が現われる場所が限定されないので、場合の数が増えてしまう。採点基準は問 1 の基準を適用する。

正解は以下の通りである。

ア {コ シ|シ コ} セ キ ソ チ タ ク (サ) ソ ツ タ (サ) ケ (サ)(ス) ソテタ (サ)(ス) ト (サ)(ス) タ

ここで、

・{コシ|シコ}はコシかシコをあらわす。
・「サ」は上記の括弧の 1 箇所のみに存在していること。
・「ス」は上記の括弧の 1 箇所のみに存在していること。
・「(サ)(ス)」は順番が入れ替わって「(ス)(サ)」でも可。

問 4 の正解の 1 つ「アシコセキソチタクソツタケソテタサストタ」は以下の通りである。

```
ア | n に整数を入力する
シ | q ← 1
コ | p ← 2 × n - 1
セ | i を 1 から n まで 1 ずつ増やしながら、くり返し
キ |   k ← i
ソ |   j を 1 から k まで 1 ずつ増やしながら、くり返し
チ |     ”□”を表示する
タ |   ここまでが「くり返し」の範囲
ク |   k ← p
ソ |   j を 1 から k まで 1 ずつ増やしながら、くり返し
ツ |     ”■”を表示する
タ |   ここまでが「くり返し」の範囲
ケ |   k ← q
ソ |   を 1 から k まで 1 ずつ増やしながら、くり返し
テ |     ”☆”を表示する
タ |   ここまでが「くり返し」の範囲
サ |   p ← p - 2
ス |   q ← q + 2
ト |  改行する
タ | ここまでが「くり返し」の範囲
```

平均で 0.1 点/7 点　(100 点満点換算 1.4 点) であった。1179 人中、満点は 9 人、4 点が 3 人、0 点が 1167 人であった。複雑な 2 重ループであったのでなかなか難しいと思われる。

5. 第 4 回模擬試験の問題

第 4 回のプログラミング問題はお絵かきの問題である。今回もプログラムを導入するような問を追加した。短冊型は問 3、問 4 である。配点は次のようになっている。

問 1：	動作の説明	8 点
問 2：	プログラムの設計 (穴埋め)	9 点
問 3：	1 重ループ	10 点
問 4：	2 重ループ	8 点

第 4 回模擬試験 #005 セット A 第 2 問

問 3　n を正の整数とし、高さが矢印 1 個分、幅が矢印 n 個分の図 3(c) の形の長方形を作り出したい。この動作をするプログラムを作成せよ。(10 点)

問 4　n を正の整数とし、幅と高さが矢印 n 個分の図 3(d) の形の正方形を作り出したい。この動作をするプログラムを作成せよ。(8 点)

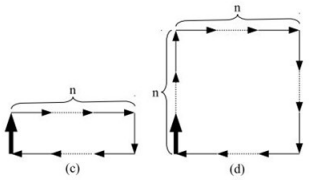


図 3: 矢印をつなげて作った図

解答群 1

ア まっすぐ	イ みぎ
ウ ひだり	エ $n-1$ 回 くり返し
オ n 回 くり返し	カ $n+1$ 回 くり返し
キ 3 回 くり返し	ク ここまでが「くり返し」の範囲

5.1　問 3 の採点基準と正解

採点基準は、解答を、1 文字の置換、1 文字の追加、1 文字の削除のいずれか 1 つによって正解に変換できる場合は 3 点減点とする。つまり、Levenshtein 距離 [*1](以下、短に距離と呼ぶ) が 1 の場合だけ部分点をつけた。

問 3 の正解は以下の通り。正解率 32 %であった。649 人中 満点が 172 人、7 点が 50 人、0 点が 427 人であった。

イ	みぎ
エ	n - 1 回くり返し
ア	まっすぐ
ク	ここまでが「くり返し」の範囲
イ	みぎ
イ	みぎ
エ	n - 1 回くり返し
ア	まっすぐ
ク	ここまでが「くり返し」の範囲

5.2　問 4 の採点基準と正解

採点基準は問 3 と同じ。

正解は「エアクキイエアクク」または「キエアクイクエアク」である。

問 4 の正解の一つは以下の通り。正解率 14 %であった。649 人中 満点 (8 点) が 69 人、5 点が 36 人、0 点が 544 人であった。

エ	n - 1 回くり返し
ア	まっすぐ
ク	ここまでが「くり返し」の範囲
キ	3 回くり返し
イ	みぎ
エ	n - 1 回くり返し
ア	まっすぐ
ク	ここまでが「くり返し」の範囲
ク	ここまでが「くり返し」の範囲

6.　第 2 回〜第 4 回のまとめ

第 2 回から第 4 回の得点をまとめると以下の通りである。

	行数	得点	100 点換算
第 2 回：ごく普通の整数計算			
問 1：1 重ループ	7	2.4/15	16/100
問 2：2 重ループ＋ if 文	10-12	0.8/20	4/100
第 3 回：図形			
問 1：1 重ループ	9	5.3/10	53/100
問 3：2 重ループ	12	0.8/10	8/100
問 4：より複雑な 2 重ループ	20	0.1/ 7	1.4/100
第 4 回：お絵描き			
問 4：1 重ループ	9	3.2/10	32/100
問 4：2 重ループ	9	1.1/ 8	14/100

100 点換算で見ると低得点であるが、回を追う毎に上昇している。行数が多い問題は得点も低くなる。1 重ループより 2 重ループの方が得点が低い。

7.　追加実験

情報専門学科のプログラミングの授業にて、1 クラス (90 人程度) を 2 グループに分け、片方は短冊型で、もう片方は自由記述で問題を解かせる小テストを 2 回行った。

1 回目は 3 つの引数から中央値を返す問題である。

短冊型の問題の解答群は次の通りである。これで、以下の関数の定義部分を埋めるというものである。

```
int median(int x, int y, int z){
 ... ここに短冊を埋める ...
}
```

[*1] 1 文字の挿入・削除・置換によって、一方の文字列をもう一方の文字列に変形するのに必要な手順の最小回数

ア	`if(x<=y){`
イ	`}`
ウ	`if(x<=z)`
エ	`if(y<=z)`
オ	`if(x<=y)`
カ	`if(z<=y)`
キ	`if(z<=x)`
ク	`if(y<=x)`
ケ	`return x;`
コ	`return y;`
サ	`return z;`

対応する自由記述型もまったくのフリーではなく、以下の制限をつけた。

> 使って良い文は、else のない if 文、return 文のみとする。変数宣言も、関数呼出しも、配列も使ってはダメ。

短冊型の問題との違いは、条件部分を自由に書けるという点である。

小テストの結果は、どちらのグループもほぼ同じ正解率 (ただし低い) であった。

2 回目は、再帰を使って数を 2 進法で表現して書き出す関数 printbinary を作らせる問題である。結果は以下の通りであった。

	○	×	合計
自由記述	7	35	42
短冊型	12	27	39

短冊型だとプログラミングが限定されることにより、正解率があがる傾向にあるようだ。

短冊型の場合、不正解のもののうち、正解との距離が 1 のものはなかった。距離が 2 のものが 4 件あったが、どれも printbinary(0) が何も印刷しないという誤りであった。距離が近い場合は意味的にも正解に近いことが見てとれた。部分点を与える場合の根拠として使えるだろうか。

8. 考察

8.1 プログラミング能力を測れるか

(1) 部分点をうまくつけるには

長いプログラムの場合、意味的にどこまで合っていれば部分点を何点つけるかの判断が難しい。短いプログラム (10 行程度) なら、Levenshtein 距離を用いて、正解より距離 1 かせいぜい距離 2 までのずれで減点するば、部分点の役割を果たすだろう。

(2) どういう問題を作るとよいか？

長いプログラムを作らせる時は、いくつかのブロックで構成するように工夫し、ブロック (つまりより短いプログラム) ごとに、距離を用いた部分点をつけるようにすれば良いだろう。

また、採点の手間を抑えるには、正解数を減らす工夫が必要である (後述)。

8.2 正解数が爆発しないように：変数による束縛

自由な書き方を導入すると (たとえばあらゆる種類の代入文などを短冊にすると)、正解数が爆発的に増える。そこである程度自由な書き方を禁止する。ただし、それでプログラミング能力が測れなくなるようでは困る。変数の利用束縛をすると若干トリッキーになるが解答の場合数を減らすことができる。

例 1: 選択肢に `if(a>b){...}`しかなければ、`if(a<=b){...}`とは書きたくても書けない。

例 2: 選択肢が、

- `for(i=1;i<=n;i++){`
- `for(j=1;j<=k;j++){`

だと、変数が束縛される。i は n に、j は k に影響される。それによってどちらのループを使うかを作題者が制御できる。

自由に書けないようにするということは、見方を変えると、制限下でのプログラムを構成する力を問うているとも言えるので、プログラミング能力があれば解けるはずである。これによって、プログラミングをパターン記憶で解こうとするやり方を排除できる。

8.3 else 付き if 文は使えないか

短冊として日本語標記を使い、日本語として読んで意味が通じるように作ることにすると、else を導入するのが難しい。

```
if(i が偶数){ ... }
else       { ... }
```

に対応する日本語記述がすっきりとはできない。現状では以下のように else なしの if 文を並べることで対処している。

```
もし、i が偶数ならば
    …
ここまでが「もし」の範囲
もし、i が奇数ならば
    …
ここまでが「もし」の範囲
```

日本語標記をやめて、標準的な記号的な書き方を使えればよいのだが、実際の試験を行う状況では、それは厳しい。(プログラムは C 言語で書くことといった制約を課すことは難しい。)

8.4 ループ

「repeat n 回」といった固定回ループは易しいが、制御変数を使うループは難しい。さらに、2 重ループはもっと難しい。制御変数が多いと場合の数が増える。

9. まとめ

短冊型問題は工夫すれば識別力はつけられそうである。小問に分け、プログラミング課題を複数出すとか、部分点は正解との距離で測るとかすればよい。変数の利用束縛は解答がトリッキーになるが、制限下でプログラムを構成する力を問うことは悪いことではない。

質疑応答

Q(美馬): 難しい問題が配点が低いのはなぜか？

A: 得点を上げるためにそうなってしまった。もともとの得点がもっと高ければそうはならないが。

Q: 「もし…ならば」、の終わりは、「ここまでがもしの範囲」ではなく、「ここまでがならばの範囲」ではなかろうか？

A: そうともいえる。

Q(八木原): if(..) の条件部を縛られるのはいやだ。自由に書きたい。奇数はそのまま、偶数ならループにしたいと思ったが、そうはかけないでは？

A: 正解の組み合わせを増やさないために、制約を付けている。そのくらいの機転を効かすことはプログラミング能力の一部と考えている。

Q(八木原): 短冊に A;B;C; とかあってもいいのでは？

A: それは良いと思う。

Q(飯尾): 計算機を使って動かしてみて、正しければ 100 点、答えは合ってるが冗長なら 80 点、答えがたまに怪しいなら 60 点、動かないなら 40 点とかやるのはどうだ？

A: CBT も考慮中ではあるが、部分点を与えるのが難しい。

Q(山之上): CBT でやるのなら、コンパイルやデバッグもさせるようにしたらどうか？

A: プログラミング過程を記録し、それから評価することができれば良いが、それもまた難しい。そう簡単に実現できるとは思えない。
(後日追加説明) CBT でやって、コンパイル、デバッグとやらせるには、言語を既定しないといけない。回答言語を一つに決めることは、実際上はなかなか難しい。(たとえば C 言語に決めてよいのか？)

C(久野): 固定回ループと制御変数を使うループで難しさの違いがわかったので、初心者に C 言語で教えるのはまずいことだと言える。

Q(田中): 代入のカードに 0 を入れるといったゲーム仕立てにすれば、開発過程もわかる。配点もそのデータからうまく並ぶように決める手はある。

A: そういうことができれば良いが、なかなかその解決は難しい。

Q(井上): 繰り返しの始めは 2 種類あるのに、終りが 1 種類しかない。どこの繰返しに対する終りかを指定できるようにしたらどうか？

A: そのようにしたい。

参考文献

[1] 中野由章, 久野靖, 佐久間拓也, 谷聖一, 筧捷彦, 村井純, 植原啓介, 中山泰一, 伊藤一成, 角田博保, 鈴木貢, 辰己丈夫, 永松礼夫, 西田知博, 松永賢次, 山崎浩二:「大学情報入試の必要性と情報入試研究会の活動」第 57 回プログラミング・シンポジウム予稿集, 2016, 155-169 (2016-1-8).

[2] 谷聖一, 佐久間拓也, 筧捷彦, 村井純, 植原啓介, 中野由章, 中山泰一, 伊藤一成, 角田博保, 久野靖, 鈴木貢, 辰己丈夫, 永松礼夫, 西田知博, 松永賢次, 山崎浩二:「『第 3 回・第 4 回大学情報入試全国模擬試験』の実施と評価」情報教育シンポジウム 2016 論文集, 2016, 7-14 (2016-08-15).