

初の情報 I 共通テストをふまえた上での プログラミングの授業デザイン

京都産業大学附属中学校・高等学校 教諭
森本 岳

1. はじめに

2025年1月、初の情報Iの共通テストが実施された。本稿では、プログラミング分野が問われた第3問をふりかえりながら、今後の情報Iの授業、特にプログラミング分野の授業のポイントを整理したい。

2. 情報 I 共通テスト雑感

情報Iには「問題解決」「プログラミング」「データの活用」「情報デザイン」と大きく4領域があるが、それぞれ並列ではなく、「問題解決」という大きな目的に対する手段として他の3領域が存在するという構成になっている。今回の共通テストの問題もこの構成に忠実に則っており、各領域の分量も含めて、実にバランスの良い出題であったと思う。

第1問は、身近な所で使われている情報システムがどのような情報技術や情報デザインに基づいて、既存の問題を解決しながら創られたか、作り手側の立場で追体験させるような問題になっている。その後、データの活用が実社会の問題解決(ビジネス)にどれほど有用なのかを再認識させるような第2問、部活動における作業工程を題材にしてモデル化とシミュレーションおよびプログラミングの基礎を段階的に押さえていく構成の第3問、観光をテーマに探究をしている生徒を主役として会話文形式で尺度をふまえた適切なグラフ選択や散布図・箱ひげ図などのグラフの読み取りを問う第4問と続く。第2問と第4問は少し難しく感じられたかもしれないが、第1問と第3問は比較的易しく、平均点も69.26点と高いことから全体的には「簡単だ」と感じた受験生が多かったと思われる。

2022年の試作問題と大きく方向性は変わらなかったが、知識をそのまま問うような問題、教科書の例題にありがちなデータ量や画像サイズを計算によって求めるような問題よりも、実社会に近いテーマで与えられた様々な条件や情報から判断し

ていくような論理的思考力が試される問題が多く、共通テストが目指す方向性をより濃く反映したものになっていたように感じる。この方向性は、間違いなく次年度入試にも引き継がれてゆくであろう。

3. 第3問をふりかえってみる

プログラミングが問われた第3問は、工芸部の部員たちが文化祭に向けて工芸品を分担して、製作する作業工程を題材にした問題であった。

問1の文章の穴埋めは、示されている製作日数の表やガントチャートの内容が把握できていれば答えられる非常に易しい問題であった。難易度を下げる方策とも受け取れるが、そもそも文章と図から必要な情報が得られていなければ得点できなかったと思われるため、問題文や図の読解力が試された。

問2は、問1の読み取りの上で展開される問題で、前半は配列を使ってどのようにプログラムを組んでいくかという過程を追体験するスタイルであった。最初にいくつかの条件が示され、各部員の手が空く日付を管理する配列としてAkibiを用意すること、その添字は1から始まり「部員番号」と一致すること、要素はその部員の手が空く日付であることが明記されている。添字は0から始まるものだと思っただけ練習を積んできた受験生は少し戸惑ったかもしれないが、結果的に部員番号と一致したことによってプログラミング経験が浅い受験生にとっては正解を導き出しやすかったかもしれない。「共通テスト用プログラム表記の例示¹⁾」では「特に説明がない場合、配列の要素を指定する添字は0から始まる」とあるため、問題文の説明には注意が必要である。

部員1が工芸品1(製作日数4)を、部員2が工芸品2(製作日数1)を、部員3が工芸品3(製作日数3)を作った後にいち早く手が空いている部員2が工芸品4(製作日数1)に取り掛かったという状況が図1で示されているため、作り終えて手が空いた日を示

す配列 Akibi は部員 1, 2, 3 の順 (= 添字の順) で 5, 3, 4 となる (問 3 の「各部員が空きになる日付を管理する配列 Akibi」の記載がヒントになる)。そして、この考え方にさらに変数 buinsu を用いてプログラムで表すとどうなるか、というのが後半で、今回は部員数は 3 人なので、buinsu には 3 が代入されている。図 4 で示されたプログラムは、話の続きにある工芸品 5 の担当部員を表示させるものだが、現在製作しているものを終えて一番早く空きになる部員を割り出す必要がある。そのためには配列 Akibi に格納されている日数を比較し、その中で一番少ないものの添字を突き止めれば、その添字 = 部員番号ということである。図 4 のプログラムでは、3 行目で暫定的に tantou を部員 1 とし、比較する相手を 4 行目で部員 2 から 1 ずつ増やしながら変えている。問題になっている if 文で部員 1 と他の部員との比較を行い、真の場合は 6 行目で担当者を部員 1 から他の部員に変えるということなので、if 文の空欄の部分では、部員 1 と他の部員を比較させて部員 1 よりも小さい (つまり部員 1 より早く手が空く) 場合、その部員を担当者としたらよいことになる (問 3 の「次に割り当てる工芸品の担当部員の番号を変数 tantou に代入する処理を行う」の記載がヒントになる)。したがって空欄キは、Akibi[buin] < Akibi[tantou] となる。最小値を見つけるプログラムであることに気づけた受験生は解きやすかったと思われる。

問 3 では、工芸品ごとに製作日数を並べた配列 Nissu (こちらも添字は 1 から)、工芸品数 9 が代入された変数 kougeihinsu が登場する。あとは問 2 と同様に、図 5 の 5 行目の外側のループの穴埋めは「kougeihin を 1 から kougeihinsu まで 1 ずつ増やしながら繰り返し」とし、その中で先ほどの問 2 のループを回せばよい。最後の画面表示の際の担当の終了日は、問題文の「どの部員も合宿 1 日目で空きであるため」という条件から Akibi[tantou] + Nissu[kougeihin] - 1 で求めることになり、マイナス 1 をしなければならぬところが少し難しかったかもしれない。こうすることで、例えば 1 日目から製作日数 3 日の工芸品を担当する場合、 $1+3-1=3$ で作業終了日は 3 日目であるため、「1 日目～3 日目」と表示できる。一方、最後の問題は終了日ではなく、手が空く日を求めるのでマイナス 1 をする必要はなく Akibi[tantou] + Nissu[kougeihin] となる。

4. おすすめしたい授業のデザイン

第 3 問に限った話ではないが、問題をきっちり読めば何とかなった問題がいくつかあったことから、まず問題文や与えられた表・グラフからの確に情報をとらえる力が必要である。それをひとつひとつ整理しながら何を言っているのかを正確に読み取ること。また、今回のように各問が連続している場合に、後ろの問題の文章からヒントを得られることもあるため、まずは大問全体を読んで全体像を把握しておくことも意識しておきたい。

授業では、「プログラムの前提条件となる情報を箇条書きで整理をした上で、プログラムも 1 行ずつどういう処理をしているかを自分で解説する」という練習を日々繰り返すことが重要になると思う。教員が解説すると、生徒は何となくわかったつもりにはなるが、当たり前の話ではあるが試験当日は自力で解く必要があるため、まずは簡単なものからでいいので自力で解説ができるように特訓する練習をしておく必要があると思う。その時にチームを組んで協働で問題の解説作りをしたり、チーム全員が異なる問題の解説を相互にするのも効果がある。

本校に進学してくる京都・大阪・滋賀の中学生でも、1/3 の生徒がプログラミングの経験が無いとアンケートで答えている (2024 年春調査)。まずはビジュアルプログラミングからでいいので、プログラムを自分の口で解説する習慣を身に付けておきたい。画面上のプログラムではイメージしにくい生徒には、micro:bit の傾きセンサー等を用いてフィジカル・コンピューティングで条件分岐や繰り返しを学ばせると、試行錯誤しながら体験でもって学べ、五感を通じて得た概念は記憶としても残りやすい。

このように、講義形式ではなく、試行錯誤を通じて自分の頭で考え、仲間と協働した経験とそれに伴う自信が、未知の問題に対して粘り強く立ち向かう姿勢や答えを導き出す力に繋がるのではないかと。限られた授業時間ではあるが、そのような経験は学校の授業の場でしか得られないため、その機会や学びを通して得られる概念や経験を大切にできるかが問われているのかもしれない。

参考文献

- 1) 大学入試センター、「共通テスト用プログラム表記の例示」、試作問題「情報」の概要、2022 年 11 月