2022.4.6

「micro:bit で学ぶプログラミング」
～入力用拡張ブロック～

2022 年度より始まる高等学校の情報科科目「情報 I」のプログラミング指導において，様々な言語（スクラッチ，VBA，JavaScript，Python など）が利用される。プログラミング言語として micro:bit を利用する際に，これらの言語，特にスクラッチや VBA 言語との互換性，及びデータの入力方法を統一的にするために，入力用の拡張ブロックを作成した。

ここでは，「micro:bit で学ぶプログラミング」（コロナ社）の例題プログラムで，ボタンを利用した箇所などを拡張ブロックに置き換えた。

＜拡張ブロックのプログラム例（JavaScript）（一部）＞

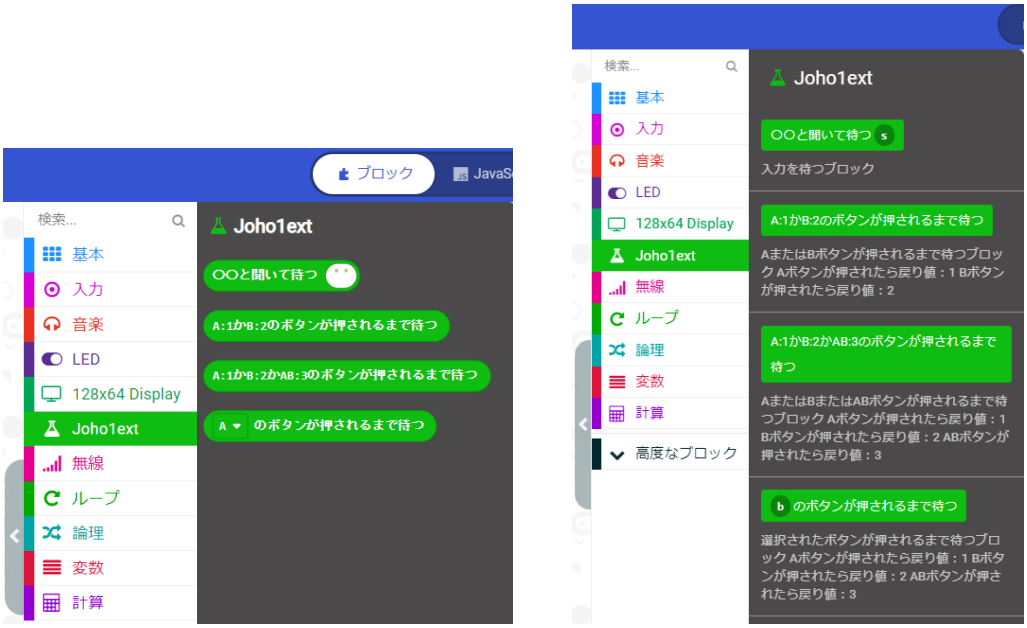
```
/**
 * 情報 I　拡張ブロック
 */
//% weight=100 color=#0fbc11 icon="¥uf0c3"
namespace Joho1ext {
    /**
         * A または B または AB ボタンが押されるまで待つブロック
         * A ボタンが押されたら戻り値：1
         * B ボタンが押されたら戻り値：2
         * AB ボタンが押されたら戻り値：3
         */
    //% block="A:1 か B:2 か AB:3 のボタンが押されるまで待つ"
    export function A か B か AB のボタンが押されるまで待つ(): number {
        while (!(input.buttonIsPressed(Button.A)) && !(input.buttonIsPressed(Button.B))
&& !(input.buttonIsPressed(Button.AB))) {

        }
        if (input.buttonIsPressed(Button.AB)) {
            return 3
        }
        else if (input.buttonIsPressed(Button.A)) {
            return 1
        }
        else if (input.buttonIsPressed(Button.B)) {
            return 2
        } else {
            return -1
        }
    }
    /**
        * A または B ボタンが押されるまで待つブロック
        * A ボタンが押されたら戻り値：1
        * B ボタンが押されたら戻り値：2
        */
    //% block="A:1 か B:2 のボタンが押されるまで待つ"
    export function A か B のボタンが押されるまで待つ(): number {
        while (!(input.buttonIsPressed(Button.A)) && !(input.buttonIsPressed(Button.B))) {

        }
        if (input.buttonIsPressed(Button.A)) {
            return 1
        } else if (input.buttonIsPressed(Button.B)) {
            return 2
        } else {
            return -1
        }
    }
```

＜拡張ブロックのプログラム例（JavaScript）の表示（一部）＞

＜拡張ブロック（Joho1ext の内容＞



作成した拡張ブロック（Joho1ext）は，上記のように，具体的なブロック(3 つ)，汎用的なブロック（１つ）であるが，ここでは，「〇〇と聞いて待つ」「A:1 か B:2 のボタンが押されるまで待つ」などを利用して例題等のプログラム変更を行った。

なお，元のプログラムや実行結果は，教科書を参照してください。また，JavaScript，Python のプログラムは，自動変換で確認してください。

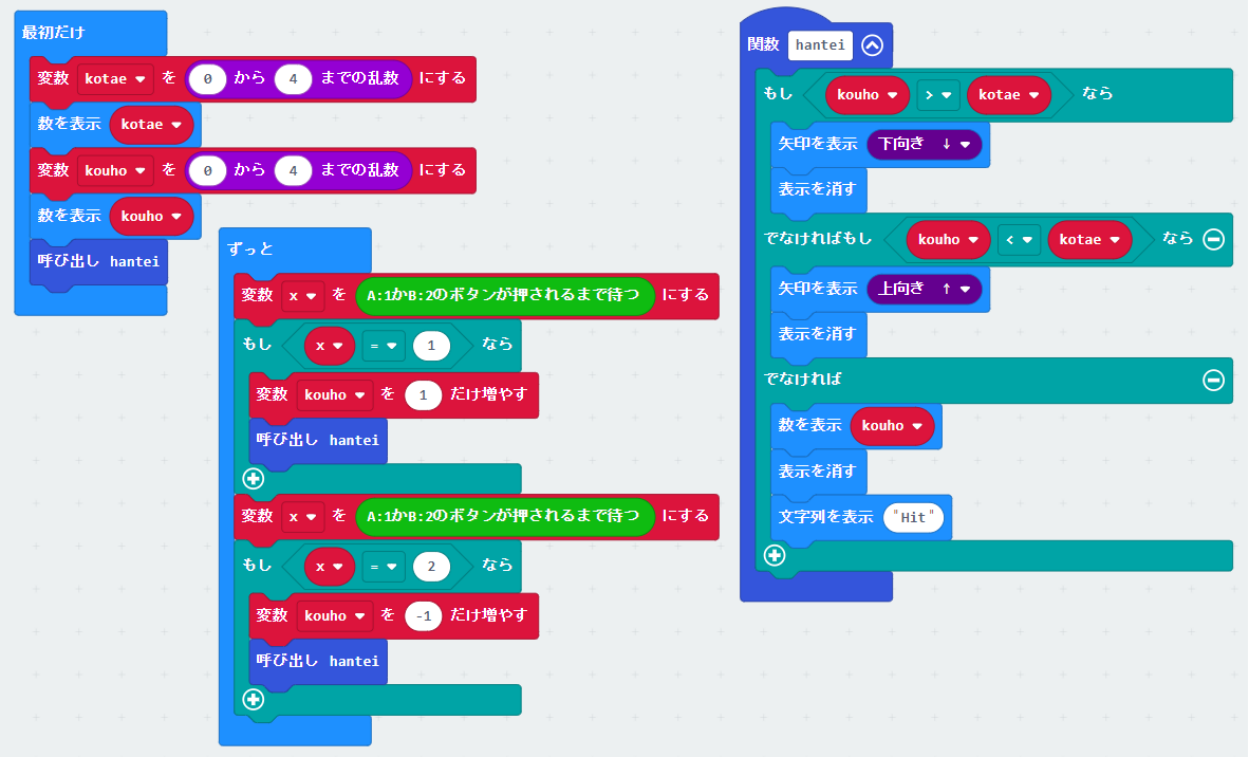
1.　2 章 2.2　数あてゲーム　(p.16～p.17)

【例題 2-3】　簡単な数あてゲームのプログラムを作成してみよう。ボタン A を押したとき，候補の数値を「0～2」の乱数で発生させる。ボタン B を押したとき，あっていれば「♥」のアイコン，間違っていれば，「×」のアイコンを表示する。

＜入力例＞

A ボタンを押すと，「1」，B ボタンを押すと，「2」が返される。

【例題 2-4】　つぎのようなプログラムを作成してみよう。「0～4」の乱数を発生させて，答えおよび候補の数値を作成し，候補の数値が答えより大きければ「↓」，候補の数値が答えより小さければ「↑」を表示する。答えと一致していれば，答えを表示した後，「Hit」と表示する。プログラムは，関数 hantei とする。なお，ボタン A を押すと，候補の数値を一つ増し，ボタン B を押すと，候補の数値を一つ減らす。

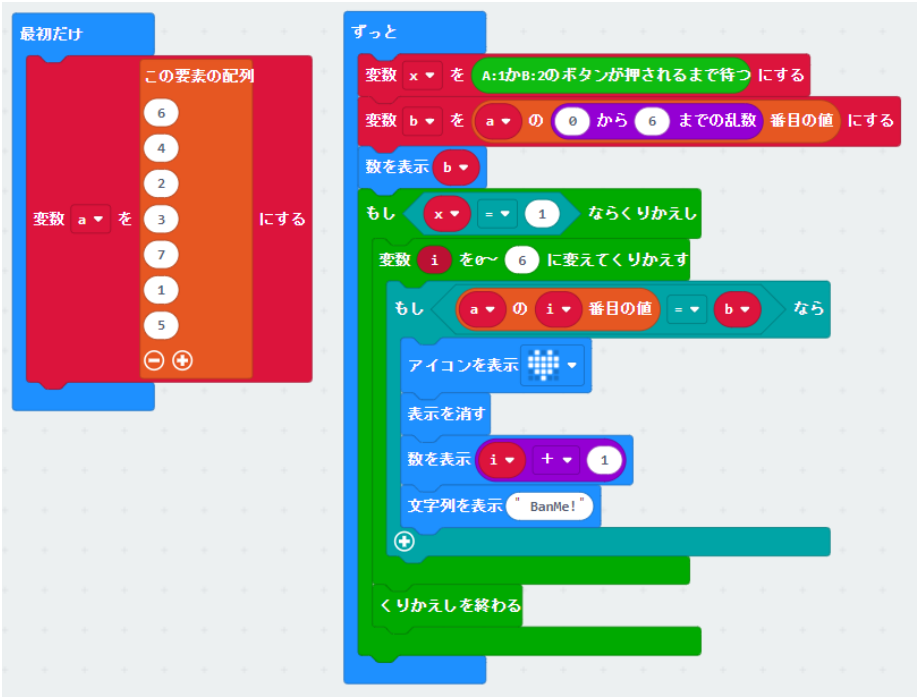


2.　5 章 5.4　探索（逐次探索）　(p.54～p.56)

【例題 5-1】　探索の中で最も簡単なアルゴリズムである逐次探索のプログラムを作成しよう。

数値データは，(6，4，2，3，7，1，5) とし，A ボタンを押すと，数値データの中からランダムに値を選び，選んだ値を探索する。探索のデータがあれば，「♥」のアイコンとデータの位置を表示しよう。

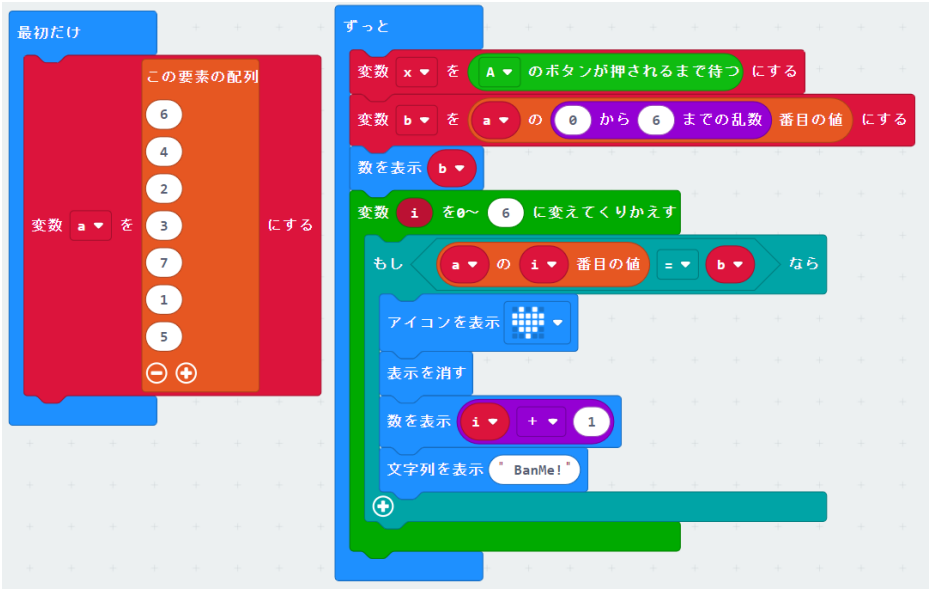
＜配列データ：固定＞



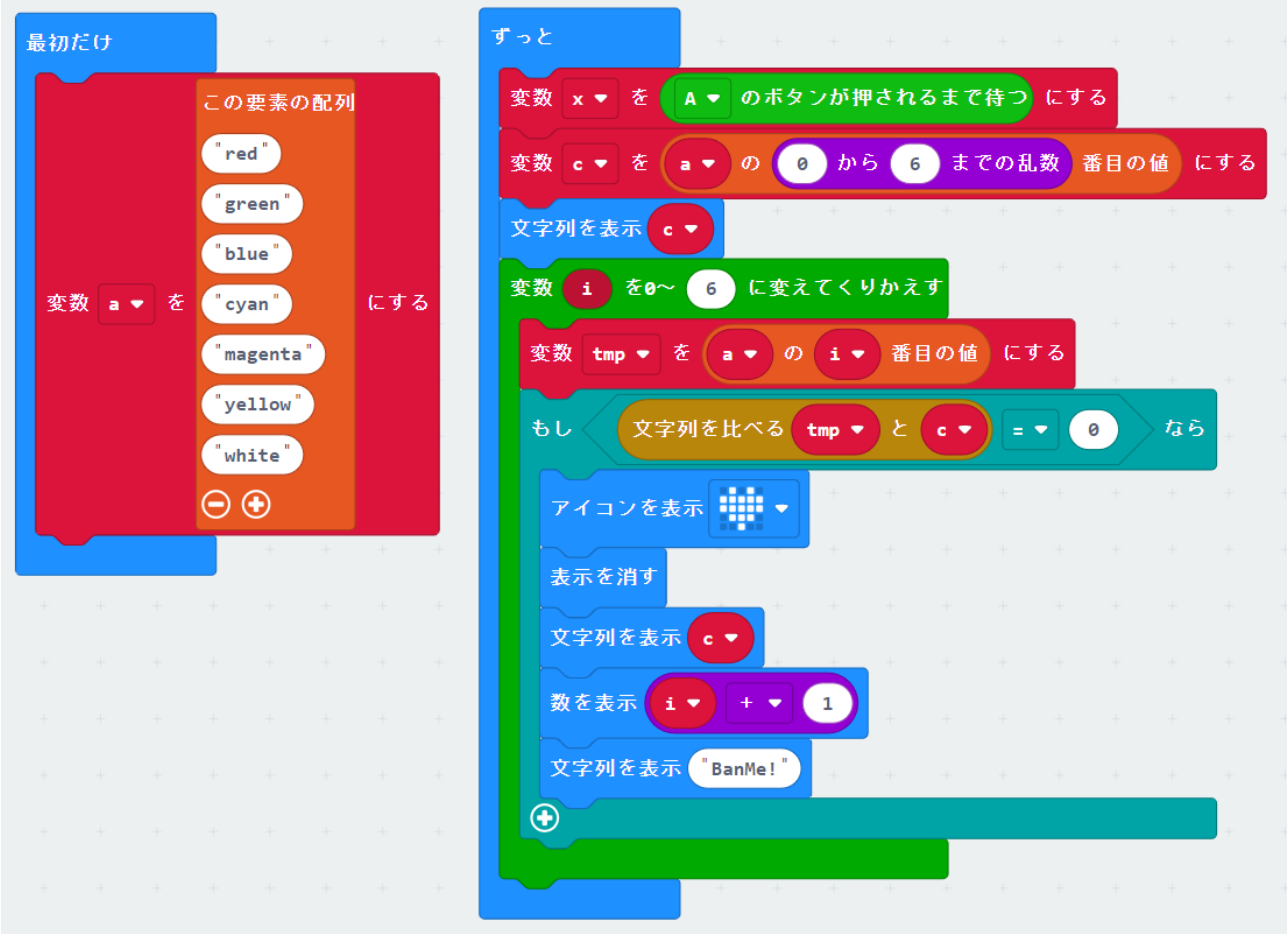
【練習 5-1】　例題 5-1 のプログラムを変更し，A ボタンを押すと探索する数値 (0～9) が順番に選択できるようにしよう。選択されている数値は LED に表示しよう。

B ボタンを押すと探索を開始し，数値データがあれば，「♥」のアイコンとデータの位置を表示し，なければ「×」を表示しよう。

＜配列データ：固定＞



【例題 5-2】　文字列の逐次探索プログラムを作成してみよう。文字列データは，(red，green，blue，cyan，magenta，yellow，white) とし，A ボタンを押すと，探索する文字列を文字データの中からランダムに選び，その文字列を探索する。探索データがあれば，「♥」のアイコンとデータの位置を表示しよう。

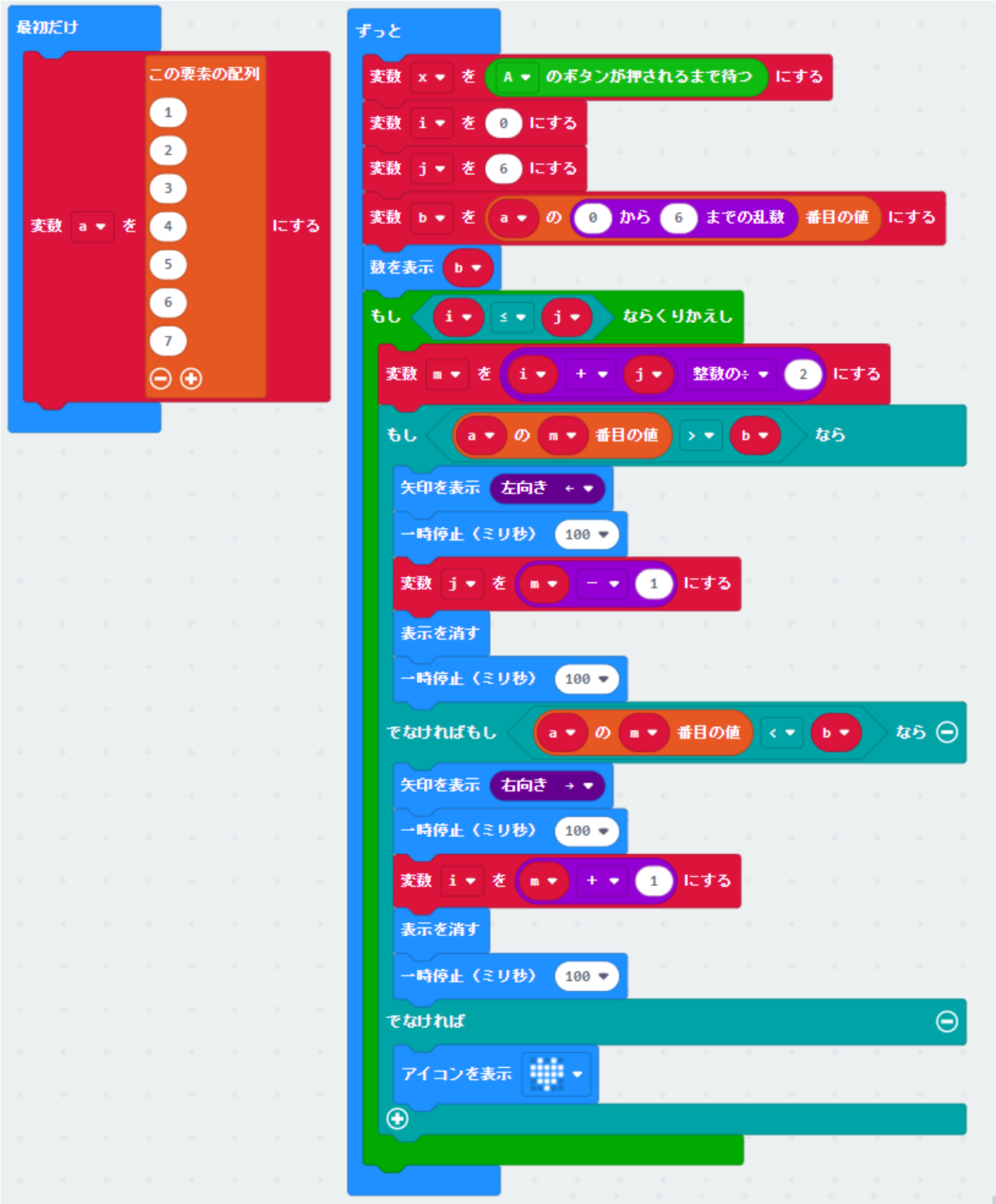


3.　5 章 5.4　探索（二分探索）　(p.58～p.59)

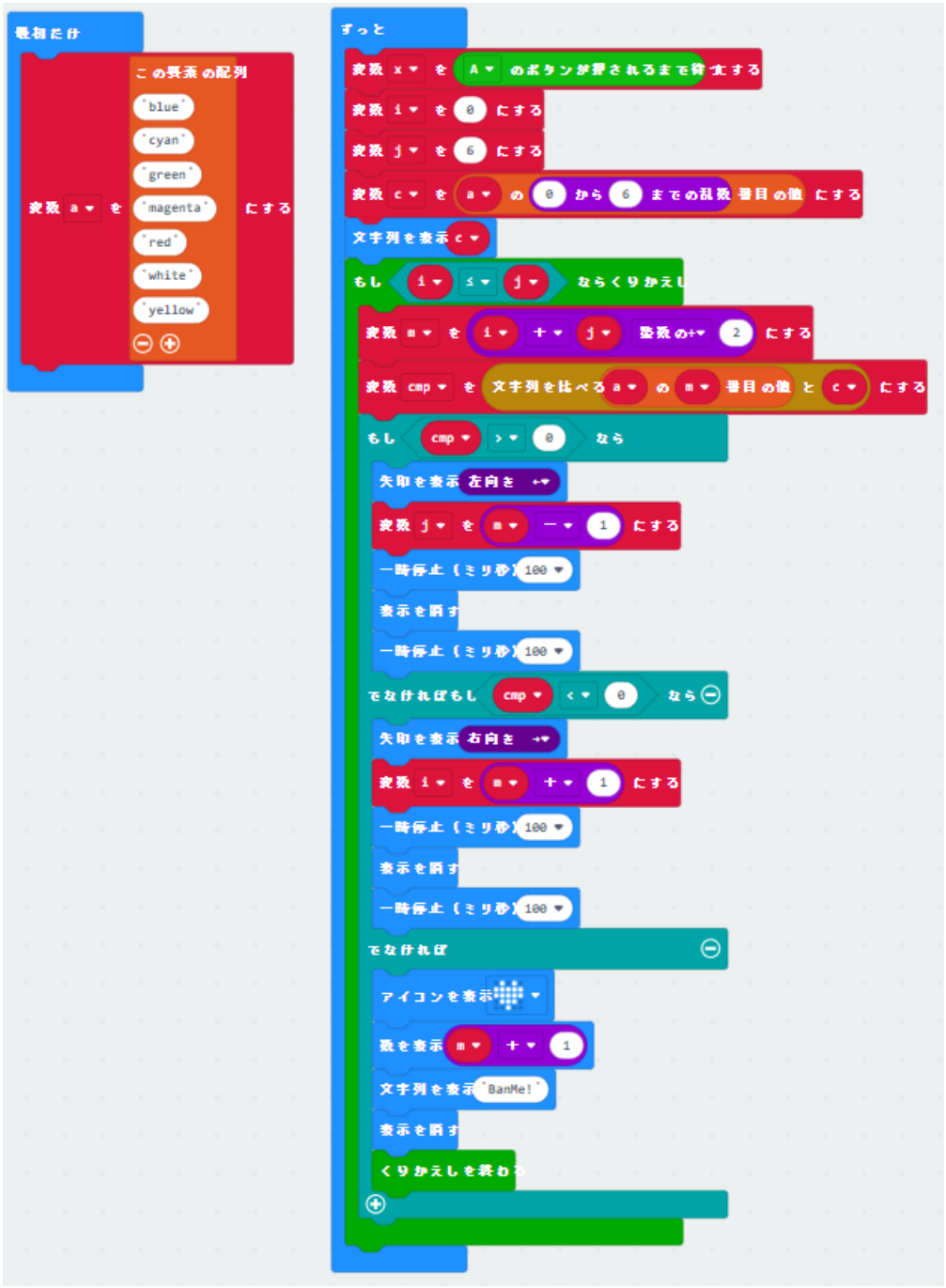
【例題 5-3】　二分探索のプログラムを作成してみよう。整列された数値データは，(1，2，3，4，5，6，7) とし，A ボタンを押すと，数値データ内の値をランダムに選び，その数値を探索する。探査する数値がいまの探索点から左側なら（小さいなら）「←」のアイコン，右側なら（大きいなら）「→」のアイコン，見つかれば「♥」のアイコンを表示しよう。

＜配列データ：固定＞

注）データは昇順



【例題 5-4】　例題 5-2，5-3 のプログラムを参考にして，文字列の二分探索プログラムを作成してみよう。なお，整列された文字列データは，(blue，cyan，green，magenta，red，white，yellow) とし，A ボタンを押すと探索する文字列を文字データ内からランダムに選び，その文字列を探索する。探索する文字列がいまの探索点から左側なら（小さいなら）「←」のアイコン，右側なら（大きいなら）「→」のアイコン，見つかれば「♥」のアイコンを表示しよう。

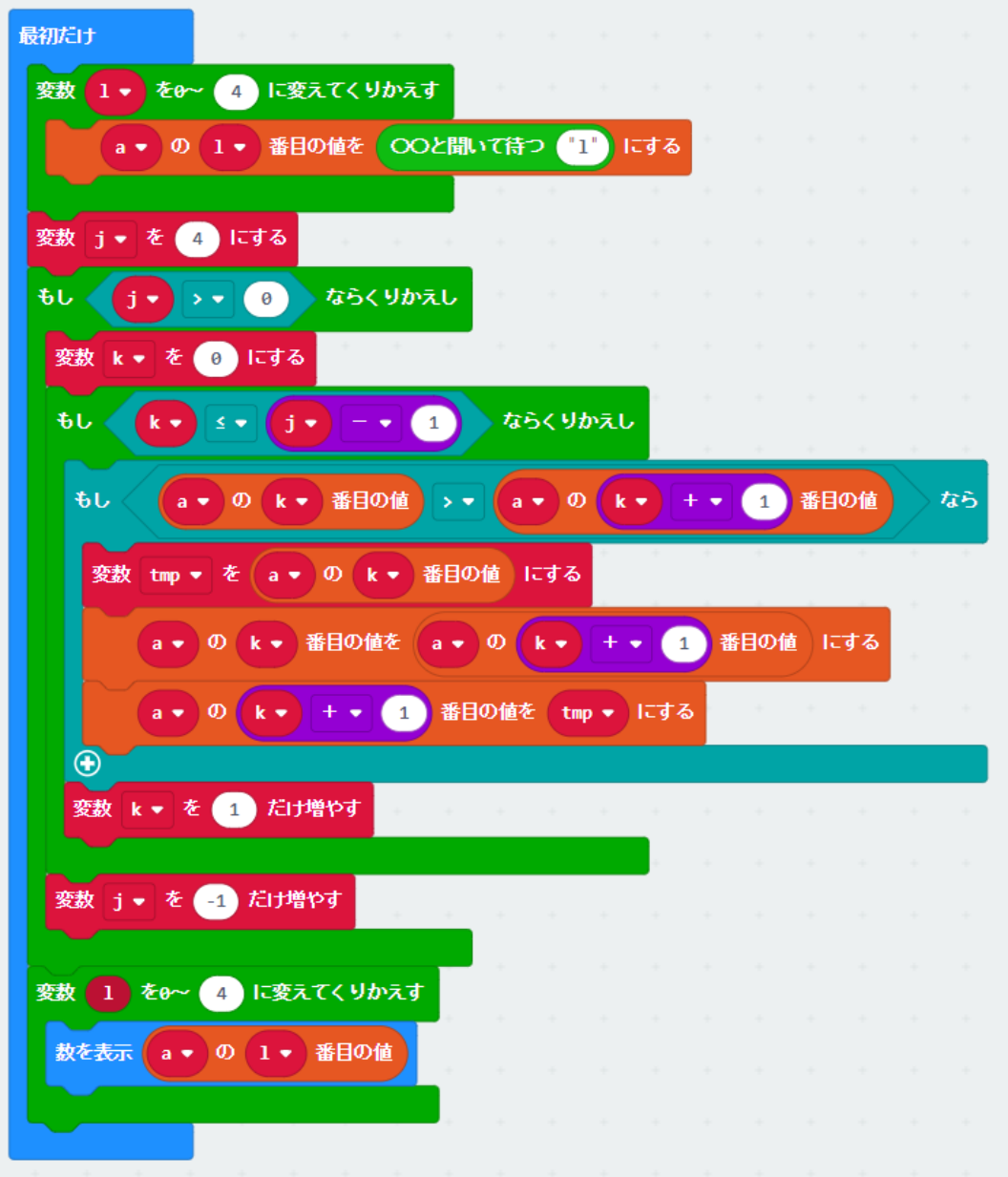


4. 5 章 5.2 整列（交換法，直接選択法） （p.61～p.63）（p.71）

【例題 5-5】 交換法で整列するプログラムを作成してみよう。数値データは，（3，2，1，5，4）とし，整列は，昇順（小さい順）とする。

＜配列データ：可変で数値は 1 桁＞

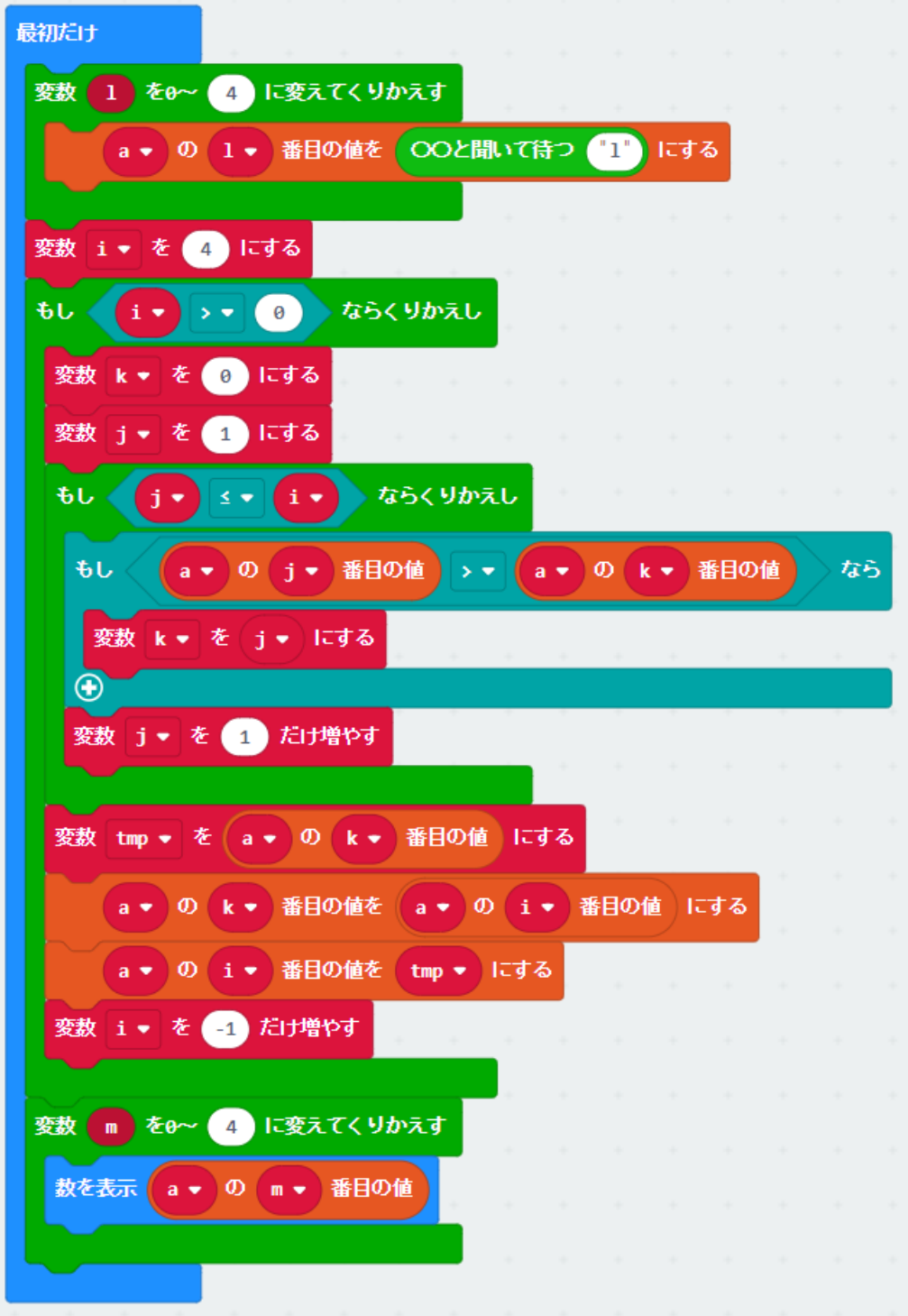
初期値は 0 が表示されるので，A ボタンで数値を変更（1 回押すと，1 つずつ増える）し，B ボタンで確定する。



【演習 5-1】　交換法で整列するプログラムを参考にして，直接選択法のプログラムを作成してみよう。

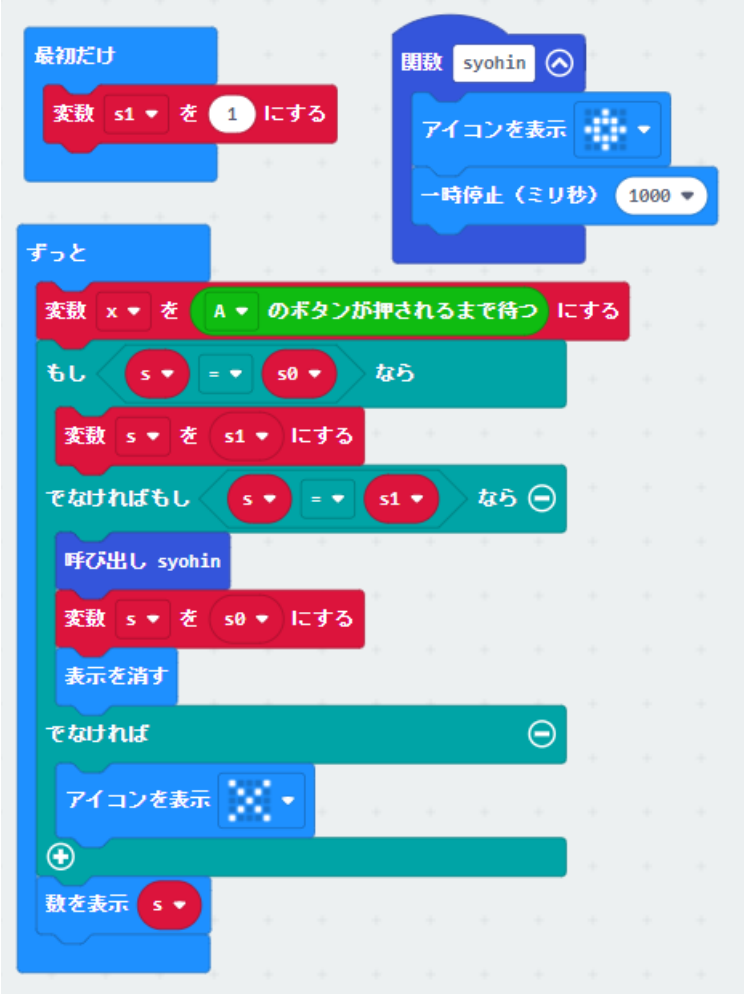
＜配列データ：可変で数値は 1 桁＞

初期値は 0 が表示されるので，A ボタンで数値を変更（1 回押すと，1 つずつ増える）し，B ボタンで確定する。



5. 5 章 5.4 自動販売機 （p.68～p.69)(p.72)

【例題 5-7】 200 円の商品を売っている自動販売機がある。投入する硬貨は 100 円だけとし，商品を購入する場合の状態遷移図は，図 5.14(省略)のようになる。スイッチ A を押すと 100 円硬貨を投入したときのシミュレーションを行うプログラムを作成してみよう。なお，商品のアイコンは，「まと」にする。

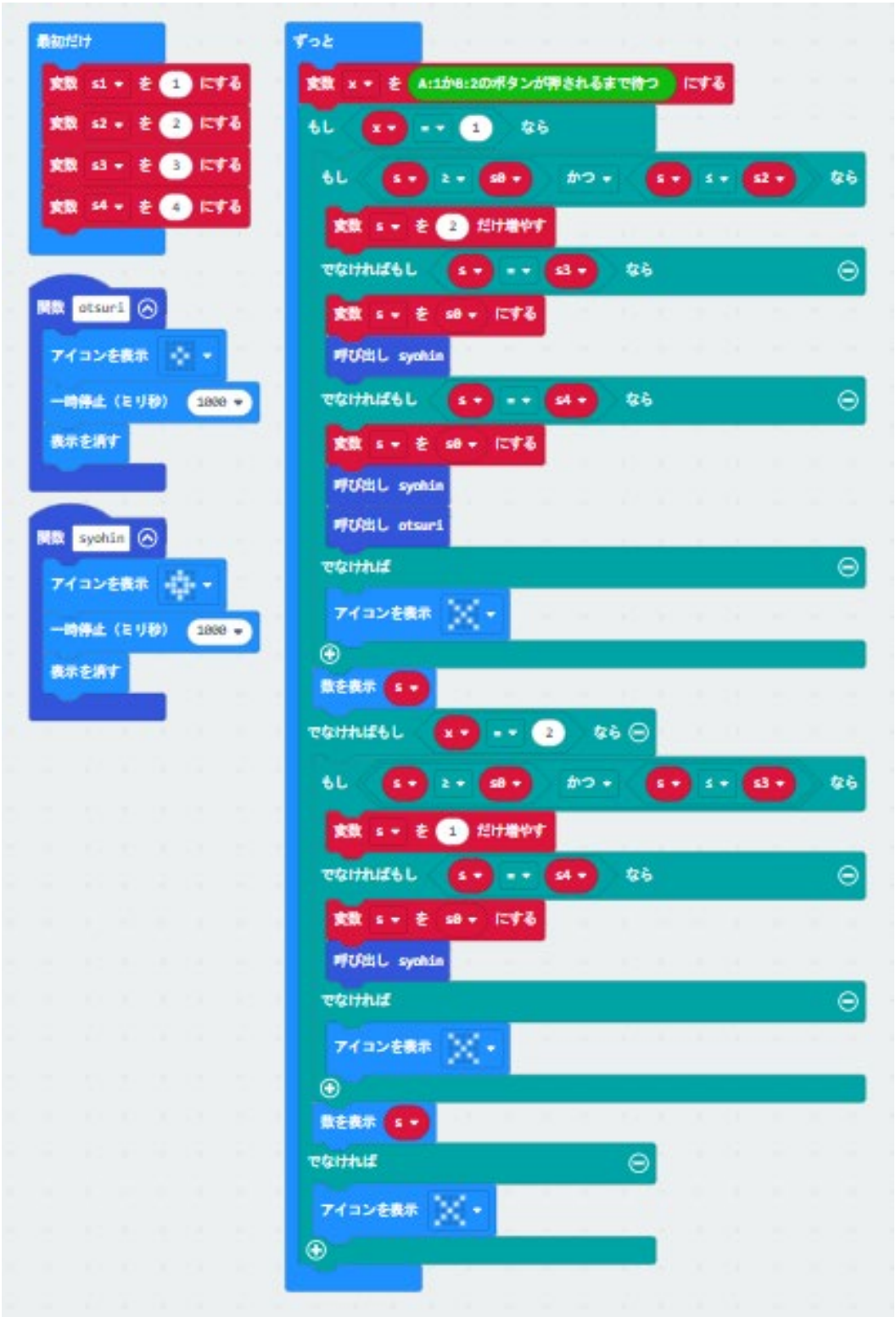


【例題 5-8】 自動販売機に 50 円と 100 円を投入して，150 円の商品を購入する場合の状態遷移図は，図 5.16(省略)のようになる。スイッチ A を押すと 100 円硬貨を投入，スイッチ B を押すと 50 円硬貨を投入したときのシミュレーションを行うプログラムを作成してみよう。なお，商品のアイコンは，「まと」，おつりのアイコンは，「小さなダイアモンド」にする。


最初だけ
変数 s1 を 1 にする
変数 s2 を 2 にする
関数 syohin
アイコンを表示
一時停止（ミリ秒） 1000
関数 otsuri
アイコンを表示
一時停止（ミリ秒） 1000
ずっと
変数 x を A:1か8:2のボタンが押されるまで待つ にする
もし x = 1 なら
もし s = s0 なら
変数 s を s2 にする
でなければもし s = s1 なら
呼び出し syohin
変数 s を s0 にする
一時停止（ミリ秒） 1000
表示を消す
でなければもし s = s2 なら
呼び出し syohin
呼び出し otsuri
変数 s を s0 にする
一時停止（ミリ秒） 1000
表示を消す
でなければ
アイコンを表示
数を表示 s
でなければもし x = 2 なら
もし s = s0 なら
変数 s を s1 にする
でなければもし s = s1 なら
変数 s を s2 にする
でなければもし s = s2 なら
呼び出し syohin
変数 s を s0 にする
一時停止（ミリ秒） 1000
表示を消す
でなければ
アイコンを表示
数を表示 s
でなければ
アイコンを表示

【演習 5-2】

（a）例題 5-8 の商品の値段を 300 円にする。自動販売機の状態遷移表は，表 5.2(省略)のとおりである。



（b）例題 5-8 の商品の値段を 250 円にする。自動販売機の状態遷移表は，表 5.3(省略)のとおりである。