

5 数当てゲーム

5.1 ハイ・ロー

5.1.1 ゲームの説明

二人でするゲームで、一方が出題者、他方が解答者になります。出題者は3桁の数を1つ思いま
す。解答者はその数を当てようとします。解答者が推理した数を言って外れたとき、出題者は次の
ようにヒントを出します。正解が推理した数より大きいときは「High」、小さいときは「Low」と
言います。

5.1.2 プログラム

コンピュータが出題者、人が解答者になってゲームをするプログラムです。

```

1  program GameHighLow; // 学生証番号 氏名
2  {$APPTYPE CONSOLE}
3  uses
4    SysUtils;
5
6  var
7    MyNumber   : Integer;           // 私が選んだ数
8    YourNumber : Integer;           // あなたが推理した数
9    Kaisuu     : Integer;           // 推理した回数
10   YesNo      : String;            // 続けるか止めるか
11 begin
12
13   WriteLn('私がでたために選んだ3桁の数を当ててください');
14   WriteLn('あなたが推理した数に対して');
15   WriteLn('私の数があなたの数より大きいときは High');
16   WriteLn('                小さいときは Low');
17   WriteLn('とヒントを教えます');
18   WriteLn('そのヒントを参考にして当ててください');
19   WriteLn('10回以内に当てられるはず');
20   repeat                           // ゲームを繰り返す
21     MyNumber := Random(900)+100;     // 100~999の乱数
22     WriteLn('***':18);
23     Kaisuu := 0;                       // 回数を0に初期設定
24     repeat                             // 推理を繰り返す
25       Inc(Kaisuu);                     // 回数を+1する
26       Write(Kaisuu:2, '回目の推理 ? '); // 入力を促す
27       ReadLn(YourNumber);              // 推理した数を読む
28       if MyNumber = YourNumber         // 当たったか
29         then begin                     // 当たったとき
30           WriteLn('':19, '当たりです');
31         end
32       else begin                         // 外れたとき
33         if MyNumber > YourNumber       // もっと大きいか
34           then begin                   // 大きいとき
35             WriteLn('':19, 'High');    // ヒント High を書く
36           end
37         else begin                       // 小さいとき
38           WriteLn('':19, 'Low');      // ヒント Low を書く
39         end;
40     end;

```

```

41     until YourNumber = MyNumber;                // 当たるまで繰り返す
42     if Kaisuu <= 10                             // 10 回以内か
43         then WriteLn('おめでとう')
44         else WriteLn('10 回以内で当てよう');
45     WriteLn;
46     Write('もう1度しますか [y / n] ? ');        // 入力を促す
47     ReadLn(YesNo);                              // 返答を読む
48     until YesNo = 'n';                          // 'n' 以外は繰り返す
49 end.

```

注 5.1 25 行目について。

`Inc(Kaisuu);` は `Kaisuu := Kaisuu+1;` と同じで、`Kaisuu` の値を 1 増加 (increment) します。

注 5.2 42~44 行目について。

`if` 文の、`then` の後と、`else` の後の `begin~end` で囲まれた部分が、文 1 個だけのときは `begin` と `end` を省略してもかまいません。

`else` の前 (上の行の最後) には ; を書いてはいけけないので注意してください。

注 5.3 22 行目 `MyNumber := Random(900)+100;` について。

`Random(n)` (^{ランダム}random) は $0 \sim n - 1$ のでたらのめの整数 (乱数) を返す関数です。 `Random(900)` は $0 \sim 899$ の乱数、したがって `Random(900)+100` は 3 桁の数 $100 \sim 999$ の乱数になります。

注 5.4 12 行目 空行 について。

乱数なので、1 問目の数、2 問目の数、... は予想できないでたらのめの数が作られます。しかし、いったんプログラムを終了して再び実行すると、1 問目の数、2 問目の数、... はさっきと同じ数が作られます。これではゲームになりませんね。これは、プログラムがいつも同じ乱数系を使うためです。

12 行目を次のように変更してください。

```

12 Randomize;                                     // 新しい乱数系にする

```

`Randomize` (^{ランダム}randomize) は乱数系を新しいものに変更します。こうすると、プログラムを実行するたびに異なる乱数系になるので、毎回違った問題が楽しめます。

`Randomize` は何回も実行するものではありません。プログラムの先頭で 1 回だけ実行するのが正しい使い方です。

5.1.3 実行

うまく推理して進めると必ず 10 回以内で当てることができます。たまたま運良く当たることもあるので、連続して 3 ゲーム 10 回以内で当てられるようがんばりましょう。

5.2 ゲーム ハミング距離

5.2.1 ゲームの説明

二人でするゲームで、一方が出題者、他方が解答者になります。出題者は4桁の数を1つ思います。解答者はその数を当てようとします。解答者が推理した数を言って外れたとき、出題者は2つの数のハミング距離をヒントとして教えます。

ハミング距離とは、両者の各桁の数の違いを足し合わせた数のことです。すなわち、正解を $ABCD$ 、推理した数を $abcd$ とすると

$$|A - a| + |B - b| + |C - c| + |D - d|$$

という式で表されます。

例 5.1 3456 と 9742 のハミング距離は

$$|3 - 9| + |4 - 7| + |5 - 4| + |6 - 2| = 6 + 3 + 1 + 4 = 14$$

5.2.2 実行例

私が選んだ4桁の数 ABCD を当ててください
 あなたが推理した数 abcd に対して
 ハミング距離 $|A-a|+|B-b|+|C-c|+|D-d|$
 をヒントとして教えてあげます
 このヒントを参考にして推理してください
 5回以内で当てられるはずですよ

```

****
1 回目の推理 ? 1234          9
2 回目の推理 ? 3455          16
3 回目の推理 ? 4321          9
4 回目の推理 ? 1000          0

```

もう1度しますか [y / n] ?

5.2.3 プログラム (実行部を除く)

```

1 program GameHumming; // 学生証番号 氏名
2 {$APPTYPE CONSOLE}
3 uses
4   SysUtils;
5
6 var
7   MyNumber   : Integer; // 私が選んだ数
8   MyA        : Integer; // その千の位
9   MyB        : Integer; //   百の位
10  MyC        : Integer; //   十の位
11  MyD        : Integer; //   一の位
12  YourNumber : Integer; // あなたが推理した数
13  YourA     : Integer; // その千の位
14  YourB     : Integer; //   百の位

```

```
15 YourC      : Integer;           // 十の位
16 YourD      : Integer;           // 一の位
17 Humming    : Integer;           // 両者のハミング距離
18 Kaisuu     : Integer;           // 推理した回数
19 YesNo      : String;            // もう1回するかの返答
```

問 5.1 実行部を作成しなさい。

5.2.4 ヒント

- (1) 4桁の数 MyNumber を乱数で作ったら、その各位の数を計算します。
- (2) 4桁の数の各位の数を求めるのは、第2回の Taisyou3keta で3桁の数の各位の数を求めたのを思い出しましょう。
- (3) 推理した数 YourNumber を読んだら、やはり各位の数を計算し、さらに MyNumber と YourNumber のハミング距離 Humming も計算します。
- (4) 絶対値 $|n|$ を計算する関数は $\text{Abs}(n)$ (アブソリュート absolute) です。
- (5) プログラムが完成するまでは、Randomize はしない方がいいです。Randomize をしないと、1問目の正解は 1000、2問目は 1282 です。答を知っているとハミング距離を正しく計算しているかどうかチェックできますね。

5.2.5 実行

今度のゲームは、必ず5回以内で当てることができます。これも3ゲーム連続して5回以内で当てるまで挑戦してください。