

## 17 プログラムの読解 2

### 17.1 模擬試験

```
1 program MagicSquare; // 学生証番号 氏名
2 {$APPTYPE CONSOLE}
3 uses SysUtils;
4
5 const
6   NMax = 11;
7
8 type
9   TBangou = 1..NMax;
10  TSquare = array [TBangou, TBangou] of Integer; // NMax 行 × NMax 列の方陣
11  TWaHyou = array [TBangou] of Integer; // 和の表
12
13 var
14   Square : TSquare;
15   WaGyou, WaRetu : TWaHyou;
16   WaNaname1, WaNaname2 : Integer;
17
18 procedure MakeMagicSquare(Size : TBangou);
19   { Size × Size の魔方陣を作る }
20   { ただし, Size は 3 以上の奇数 }
21   var
22     N : Integer;
23     Gyou, Retu : TBangou;
24     GyouKouho, RetuKouho : TBangou;
25   begin
26     for Gyou := 1 to Size do
27       begin
28         for Retu := 1 to Size do
29           begin
30             Square[Gyou, Retu] := 0;
31           end;
32         end;
33     Gyou := (Size+1) div 2;
34     Retu := 1;
35     Square[Gyou, Retu] := 1;
36     for N := 2 to Sqr(Size) do
37       begin
38         if Gyou <= Size-1
39           then GyouKouho := Gyou+1
40           else GyouKouho := Gyou+1-Size;
41         if Retu <= Size-2
42           then RetuKouho := Retu+2
43           else RetuKouho := Retu+2-Size;
44         if Square[GyouKouho, RetuKouho] = 0
45           then begin
46             Gyou := GyouKouho;
47             Retu := RetuKouho;
48           end
49         else begin
50             Gyou := Gyou;
```

```
51         Retu := Retu+1;
52     end;
53     Square[Gyou,Retu] := N;
54 end;
55 end; {MakeMagicSquare}
56
57 procedure WaKeisan(Size : TBangou);
58     { 行, 列, 対角線の和を計算する }
59     var
60     I,K : TBangou;
61     begin
62     for K := 1 to Size do
63     begin
64     WaGyou[K] := 0;
65     WaRetu[K] := 0;
66     end;
67     WaNaname1 := 0;
68     WaNaname2 := 0;
69     for I := 1 to Size do
70     begin
71     for K := 1 to Size do
72     begin
73     WaGyou[I] := WaGyou[I]+Square[I,K];
74     WaRetu[K] := WaRetu[K]+Square[I,K];
75     end;
76     WaNaname1 := WaNaname1+Square[I,I];
77     WaNaname2 := WaNaname2+Square[I,Size+1-I];
78     end;
79     end; {WaKeisan}
80
81 procedure WriteSquare(Size : TBangou);
82     { Square を書く }
83     var
84     Gyou,Retu : TBangou;
85     begin
86     for Gyou := 1 to Size do
87     begin
88     Write('':8);
89     for Retu := 1 to Size do
90     begin
91     Write(Square[Gyou,Retu]:4);
92     end;
93     WriteLn(' ':4, WaGyou[Gyou]:4);
94     end;
95     Write(' / ':8);
96     for Retu := 1 to Size do
97     begin
98     Write(' ':4);
99     end;
100    WriteLn(' \ ':4);
101    Write(WaNaname1:4, '':4);
102    for Retu := 1 to Size do
103    begin
104    Write(WaRetu[Retu]:4);
105    end;
106    WriteLn(WaNaname2:8);
107 end; {WriteSquare}
108
```

```
109 var
110   Ji : TBangou;
111 begin {Main}
112   WriteLn(' n 次の魔方陣を 1 つ作ります ( n は 3 以上の奇数 ) ');
113   repeat
114     Write(' 何次 ? ');
115     ReadLn(Ji);
116   until (Ji mod 2 = 1) and (3 <= Ji) and (Ji <= NMax);
117   MakeMagicSquare(Ji);
118   WaKeisan(Ji);
119   WriteSquare(Ji);
120   ReadLn;
121 end.
```

問 17.1 プログラム MagicSquare.dpr を実行し、次数として 5 を入力したとします。

- (1) プロシージャ MakeMagicSquare に従って、解答欄の Square の表を完成させなさい。初期設定の 0 は書かなくていいです。
- (2) プロシージャ WaKeisan に従って、解答欄の WaGyou, WaRetu, WaNaname1, WaNaname2 が変化する過程を書きなさい。
- (3) プロシージャ WriteSquare に従って、出力を解答欄に書きなさい。

