22　読解問題

あるプロシージャが自分自身を呼び出して使うことを**再帰呼び出し**といい，そのようなプロシージャを**再帰手続き (recursive procedure)** といいます。

22.1　例

次のプログラムを実行すると，下のような実行結果になります。

プログラム

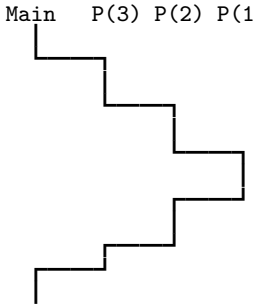
```
 1  program Saiki1;
 2  {$APPTYPE CONSOLE}
 3  uses SysUtils;
 4
 5  procedure P(K : Integer);
 6    begin
 7      WriteLn('P(', K, ') 開始');
 8      if K > 1
 9        then begin
10               P(K-1);
11               // P(K-1) が終了すると，P(K) のここに戻ってくる
12             end;
13      WriteLn('P(', K, ') 終了');
14    end;
15
16  var
17    N : Integer;
18  begin {Main}
19    N := 3;
20    P(N);
21    // P(N) が終了するとここに戻ってくる
22    ReadLn;
23  end.
```

実行結果

```
P(3) 開始
P(2) 開始
P(1) 開始
P(1) 終了
P(2) 終了
P(3) 終了
```

実行の推移

P(K) を呼び出したとき，“P(K) の先頭にジャンプする” と考えないで，“新しく P{K) のコピーを作って，それを実行する” と考えるとわかりやすいでしょう。



22.2　問題

次のプログラムを実行したときの実行結果を書きなさい。

22.2.1

```
 1  program Saiki2;
 2  {$APPTYPE CONSOLE}
 3  uses SysUtils;
 4
 5  var
 6    Answer : Integer;
 7
 8  procedure P(K : Integer);
 9    begin
10      WriteLn('P(', K, ') 開始');
11      Answer := Answer*10+K;               // P(K) に入った時に計算する
12      WriteLn(''':20, Answer);
13      if K > 1
14        then begin
15               P(K-1);
16               // P(K-1) が終了すると，P(K) のここに戻ってくる
17             end;
18      WriteLn('P(', K, ') 終了');
19    end; {P}
20
21  var
22    N : Integer;
23  begin
24    N := 3;
25    Answer := 0;
26    P(N);
27    // P(N) が終了するとここに戻ってくる
28    WriteLn;
29    WriteLn('[答] ', Answer);
30    ReadLn;
31  end.
```

22.2.2

```
 1  program Saiki3;
 2  {$APPTYPE CONSOLE}
 3  uses SysUtils;
 4
 5  var
 6    Answer : Integer;
 7
 8  procedure P(K : Integer);
 9    begin
10      WriteLn('P(', K, ') 開始');
11      if K > 1
12        then begin
13               P(K-1);
14               // P(K-1) が終了すると，P(K) のここに戻ってくる
15             end;
16      Answer := Answer*10+K;               // P(K) から戻る時に計算する
17      WriteLn(''':20, Answer);
18      WriteLn('P(', K, ') 終了');
19    end; {P}
20
21  var
22    N : Integer;
23  begin
24    N := 3;
25    Answer := 0;
26    P(N);
27    // P(N) が終了するとここに戻ってくる
28    WriteLn;
29    WriteLn('[答] ', Answer);
30    ReadLn;
31  end.
```

22.2.3

```
 1  program Saiki4;
 2  {$APPTYPE CONSOLE}
 3  uses SysUtils;
 4
 5  var
 6    Answer : Integer;
 7
 8  procedure P(K : Integer);
 9    begin
10      WriteLn('P(', K, ') 開始');
11      Answer := Answer*10+K;               // P(K) に入った時にも計算する
12      WriteLn('':20, Answer);
13      if K > 1
14        then begin
15               P(K-1);
16               // P(K-1) が終了すると，P(K) のここに戻ってくる
17             end;
18      Answer := Answer*10+K;               // P(K) から戻る時にも計算する
19      WriteLn('':20, Answer);
20      WriteLn('P(', K, ') 終了');
21    end; {P}
22
23  var
24    N : Integer;
25  begin
26    N := 3;
27    Answer := 0;
28    P(N);
29    // P(N) が終了するとここに戻ってくる
30    WriteLn;
31    WriteLn('[答] ', Answer);
32    ReadLn;
33  end.
```

22.2.4

```
 1  program Saiki5;
 2  {$APPTYPE CONSOLE}
 3  uses SysUtils;
 4
 5  procedure P(K : Integer);
 6    begin
 7      WriteLn('P(', K, ') 開始');
 8      if K > 1
 9        then begin
10               P(K-1);
11               // P(K-1) が終了すると，P(K) のここに戻ってくる
12             end;
13      if K > 1
14        then begin
15               P(K-1);
16               // P(K-1) が終了すると，P(K) のここに戻ってくる
17             end;
18      WriteLn('P(', K, ') 終了');
19    end; {P}
20
21  var
22    N : Integer;
23  begin
24    N := 3;
25    P(N);
26    // P(N) が終了するとここに戻ってくる
27    ReadLn;
28  end.
```

22.2.5

```
 1  program Saiki6;
 2  {$APPTYPE CONSOLE}
 3  uses SysUtils;
 4
 5  var
 6    Answer : Integer;
 7
 8  procedure P(K : Integer);
 9    begin
10      WriteLn('P(', K, ') 開始');
11      if K > 1
12        then begin
13               P(K-1);
14               // P(K-1) が終了すると, P(K) のここに戻ってくる
15             end;
16      Answer := Answer*10+K;      // 途中で計算する
17      WriteLn(''':20, Answer);
18      if K > 1
19        then begin
20               P(K-1);
21               // P(K-1) が終了すると, P(K) のここに戻ってくる
22             end;
23      WriteLn('P(', K, ') 終了');
24    end; {P}
25
26  var
27    N : Integer;
28  begin
29    N := 3;
30    Answer := 0;
31    P(N);
32    // P(N) が終了するとここに戻ってくる
33    WriteLn;
34    WriteLn('[答] ', Answer);
35    ReadLn;
36  end.
```

22.2.6

```
 1  program Saiki7;
 2  {$APPTYPE CONSOLE}
 3  uses SysUtils;
 4
 5  var
 6    Answer : Integer;
 7
 8  procedure P(K : Integer);
 9    var
10      I : Integer;
11    begin
12      //WriteLn('P(', K, ') 開始');
13      for I := 1 to 3 do
14        begin
15          Answer := Answer*10+I;
16          WriteLn(Answer);
17          if Answer mod 7 = 0
18            then WriteLn('[答] ':20, Answer)
19            else if K > 1
20                   then begin
21                          P(K-1);
22                          // P(K-1) が終了すると, P(K) のここに戻ってくる
23                        end;
24          Answer := Answer div 10;      // 元に戻す
25        end;
26      //WriteLn('P(', K, ') 終了');
27    end; {P}
28
29  var
30    N : Integer;
31  begin
32    N := 3;
33    Answer := 0;
34    P(N);
35    // P(N) が終了するとここに戻ってくる
36    ReadLn;
37  end.
```

22.3　オセロ馬鹿詰めを解くプログラム

●○協力して，なるべく早く●だけまたは○だけにする手順を求めるプログラムです。

```
  end; {DokonimoOkenai}  // これより下を消して，追加する

procedure WriteAnswer(Kai : String);
          { 答を書く }
  begin
    WriteBan;
    WriteLn(Kai);
    ReadLn;
  end; {WriteAnswer}

procedure Search(Tesuu : Integer; Kai : String);
          { 10 手以内で終了する手順を探す }
  var
    BanHozon : TBan;
    KosuuHozon : TKosuu;
    OkeruMasuHozon : TMasus;
    OkeruSuuHozon : Integer;
    K : Integer;
  begin
    if (Kosuu[Kuro] = 0) or (Kosuu[Siro] = 0)
      then WriteAnswer(Kai)
      else if Tesuu > 0
             then begin
                    BanHozon := Ban;
                    KosuuHozon := Kosuu;
                    ListUpOkeruMasu;
                    OkeruMasuHozon := OkeruMasu;
                    OkeruSuuHozon := OkeruSuu;
                    for K := 1 to OkeruSuuHozon do
                      begin
                        Oku(OkeruMasuHozon[K]);
                        Hanten(Teban);
                        Hanten(Aite);
                        Search(Tesuu-1, Kai+' '+IntToStr(OkeruMasuHozon[K]));
                        Hanten(Teban);
                        Hanten(Aite);
                        Ban := BanHozon;
                        Kosuu := KosuuHozon;
                      end;
                  end;
  end; {Search}

begin {Main}
  SyokiSettei;
  Search(10,'');
  WriteLn(' サーチ終了');
  ReadLn;
end.
```

これだと解が多すぎるので，初期設定で初めの３手を置いた状態にして，それから７手以下で終了する手順を求めるのがいいかもしれません。