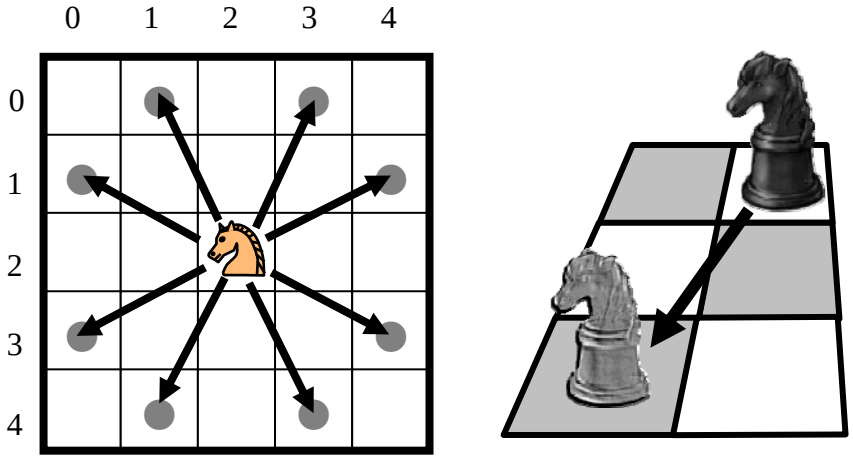
5.4　騎士巡歴問題

(1)騎士巡歴問題とは

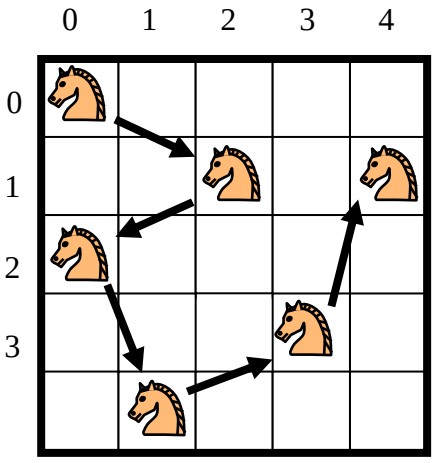
騎士巡歴(Knight's Tour)問題とは，N×N のマス目上のすべてをチェスのナイトが訪れる順序を求める問題です。

チェスのナイトは，進行方向に対して+2，直交方向に+1 だけジャンプします。いわば，将棋の桂馬と似た働きをします。正確にいうと，もし，5×5 のマス目の中央にナイトがいたとすると，以下の灰色丸印の 8 方向にジャンプできます。



(2)解を求める方法

最初(0, 0)の位置にいるものとします。ナイトがジャンプできる方向に，順次動いていくものとします。



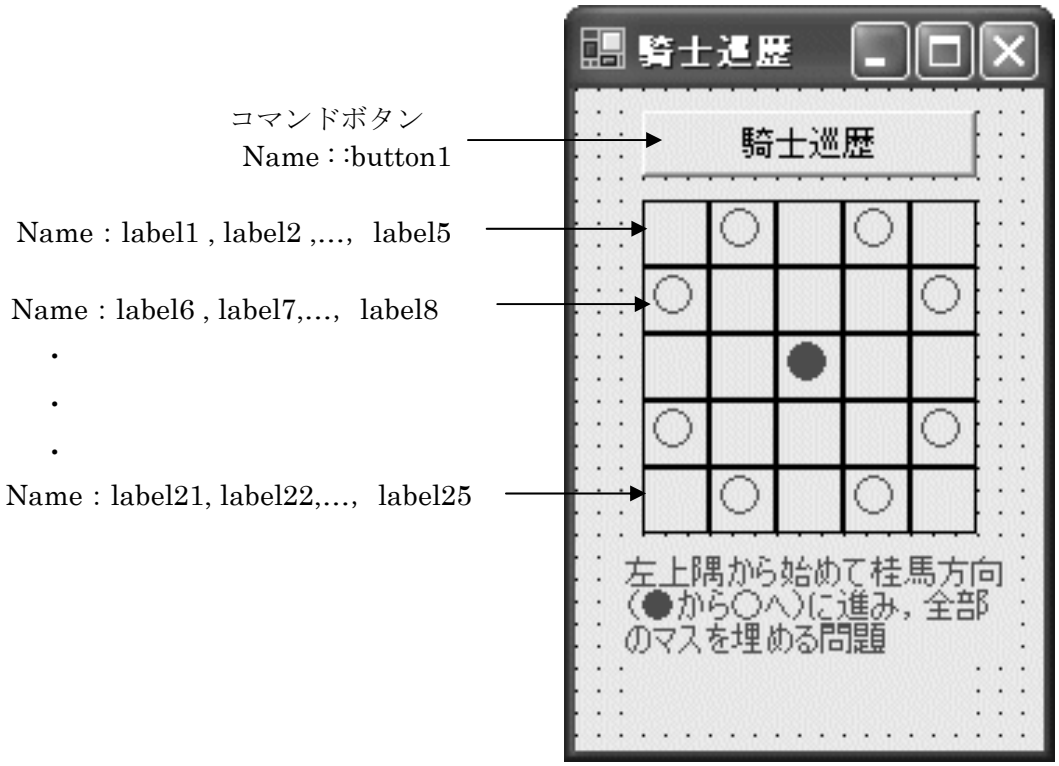
ジャンプできる方向は，８通りありますから，以下のような表を作成しておきます。

```
ｉ）行先表[0].X =  2; 行先表[0].Y =  1;
ii）行先表[1].X =  1; 行先表[1].Y =  2;
iii）行先表[2].X = -1; 行先表[2].Y =  2;
iv）行先表[3].X = -2; 行先表[3].Y =  1;
ｖ）行先表[4].X = -2; 行先表[4].Y = -1;
vi）行先表[5].X = -1; 行先表[5].Y = -2;
vii）行先表[6].X =  1; 行先表[6].Y = -2;
viii）行先表[7].X =  2; 行先表[7].Y = -1;
```

これらの行き先に，すでにコマがあれば，失敗とみなし，最大到着数(25 コマ)に達すれば成功とします。

[Program 5-3] 騎士巡歴問題

ラベルの位置で，各コマを表現します。



【実行例】

右のように，騎士が訪れたマスに到着順序を示す番号が表示されます。

プログラムリスト１

```
public const int NSIZE  = 5;
public Label[,] LabelDT = new Label[NSIZE+1,NSIZE+1];
public int[,]   盤面    = new int[NSIZE+1,NSIZE+1];
public int 到達最大数   = NSIZE*NSIZE;
public int 解番号       = 0;
public int 到達数       = 0;
public struct 行先
{ public int X; public int Y;
}
public 行先[] 行先表 = new 行先[8];
private void 初期設定()
{ int i,j, ID; string labelName; int X=-5;
  for(i = 1;i < 6; i++)    // 盤面用ラベルの 2 次元配列化
  { X += 5;
    for(j = 1; j < 6; j++)
    { ID = X +j;
      labelName="label" + ID.ToString();
      foreach( Control myControl in Controls)
          if(myControl.Name == labelName)
              { LabelDT[i,j] = (Label) myControl; break;}
    }
  }
```

プログラムリスト2

```
    行先表[0].X= 2; 行先表[0].Y= 1;  // 行き先表の設定
    行先表[1].X= 1; 行先表[1].Y= 2;
    行先表[2].X=-1; 行先表[2].Y= 2;
    行先表[3].X=-2; 行先表[3].Y= 1;
    行先表[4].X=-2; 行先表[4].Y=-1;
    行先表[5].X=-1; 行先表[5].Y=-2;
    行先表[6].X= 1; 行先表[6].Y=-2;
    行先表[7].X= 2; 行先表[7].Y=-1;
}
private void 表示()
{ int i,j;
  for(i = 1; i < 6; i++)
     for(j = 1; j < 6; j++)LabelDT[i,j].Text = 盤面[i,j].ToString();
  解番号++;
  MessageBox.Show(解番号.ToString() +"番目の解が見つかりました");
}
private void Try(int X,int Y)
{ int i;
  if(X < 1 || X > NSIZE || Y < 1 || Y > NSIZE) return;
  if(盤面[X,Y] > 0) return;
  到達数++; 盤面[X,Y] = 到達数;
  if(到達数 >= 到達最大数) 表示();
  else for(i = 0; i < 8; i++) Try(X + 行先表[i].X,Y + 行先表[i].Y);
  盤面[X,Y] = 0; 到達数--;
}
private void Form1_Load(object sender, System.EventArgs e)
{ 初期設定();
}

private void button1_Click(object sender, System.EventArgs e)
{ int i,j;
  for(i = 1; i <= NSIZE; i++)for(j = 1; j <= NSIZE; j++)盤面[i,j] = 0;
  解番号 = 0; 到達数 = 0;
  Try(1,1);
  lblMessage.Text = "解の数=" + 解番号.ToString() + "通り";
  MessageBox.Show(解番号.ToString() +"個の解が見つかりました");
}
```