

- ▼ N-王妃(or 8-Queen)問題(暇なら以下の問題を考えよ)
 - NxNマスの盤にN個のQueenをお互いが縦, 横, 斜めにあたらないように置く, すべての可能な配置を表示する問題を考える. N=8の8-王妃問題が有名.
- ▼ 初期化
 - Queen:=Array(1..N);
ととる. ここでQueen[1]:=2とすると, 1行目のQueenが2にあるとする.
- ▼ printQ:=proc(Queen)
 - ▼ Queenの配置を表示する関数を考えよ. 例えば,
 - ▶ N:=5:


```
Queen:=Array(1..N);
for i from 1 to N do Queen[i]:=i; end do;
printQ(Queen);
Q . . . .
. Q . . .
. . Q . .
. . . Q .
. . . . Q
```

 とQueenの配置を表示してくれる.
- ▼ "あたり"を確認するcheck関数
 - "あたり"を確認するcheck関数を作る.
 - ▶ ヒント:この関数の動作は複雑であるが, 4, 5行で書ける. i行目のQueenをj列目に入れたとして, i, jを引数として受け取る. p行に入れたQueenがq列目にあるとして, 縦, 前斜め, 後ろ斜めに当たってないか調べる. これをi-1行まですべてについて調べる. 一つでもあたりがあればfalseを返し, 当たってない時にはtrueを返すようにすればよい.
- ▼ 再帰関数try1:=proc(i)
 - N-王妃の配置を考えるには, 再帰関数を使うと便利である. あるi行での場所が


```
Queen[i]:=k
```

 となると仮に決める. これ以前の行に置いたQueenと縦・横・斜めに当たってないかをcheck関数で確認して当たってなければ次の"i+1行"へ行く. あたれば次の列に置いてみてcheckする. これを繰り返し, i=N+1となれば成功. ここまで来たQueen配置を出力すればよい.
 - 再帰関数は,


```
try1:=proc(i)
global Queen, N;
...
end proc;
```

 としてi行目の配置を作っていく. "...の部分を考える.