# АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ПЕРМУТАЦИИ

Нека числата на търсената пермутация са в масива nom[]. Въвеждаме променлива mx=0. При срещане на знака > в променлива p1 запомняме позицията i на първия от тях, ако са последователни. Когато на позиция i срещнем знака <: тогава ако p1=0 увеличаваме mx и nom[i]=mx, иначе от позиция i до p1 увеличаваме mx и nom[j]=mx, където j намалява от i до p1.

```
for (i=1;i<N;i++)
 if (c[i]=='<') {
  if (p1==0)                              // няма последователност от > преди i
   nom[i]=++mx;                           // увеличаваме mx
   else {                                 // има последователност от > преди i
    mx++;                                 // увеличаваме mx
    nom[i]=mx;                            // и го присвояваме на текущото nom[i]
    for (j=i-1;j>=p1;j--) nom[j]=++mx;    // за всички j от i-1 до p1 правим същото
    p1=0;                                 // нулираме p1
   }
 }
 else
 if (p1==0)                  // запомняме първото появяване на последователност от >
  p1=i;
```

Остава да се съобразим с последния знак.

Ако той е <, последно число в пермутацията става равно на N:

```
if (c[N-1]=='<')
 nom[N]=N;
```

Когато последният знак е >, последния елемент nom[N] става mx+1 и ако p1>0 отново образуваме цикъл отзад-напред за числата между последния индекс N-1 и p1:

```
nom[N]=++mx;
if (p1>0)
  for (j=N-1;j>=p1;j--) nom[j]=++mx;
```

*Автор: Павел Петров*