

## Musteraufgabe:

# Das Haus des Nikolaus

Für das Zeichnen des Haus-des-Nikolaus gelten die bekannten Regeln:

► Der Streckenzug muß ohne Absetzen gezeichnet werden.

► Jede Strecke wird genau einmal durchlaufen.

Nun steht das Haus-des-Nikolaus am Ufer eines Sees, in dem es gespiegelt wird. Es entsteht das folgende Bild:

## Aufgabe:

Schreibe ein Programm, das alle möglichen Wege ausgibt, die nach den oben angegebenen Regeln dieses Bild entstehen lassen! Gib dabei die Reihenfolge der durchlaufenen Punkte mit der oben angegebenen Bezeichnung aus!

## Lösungsidee:

Alle möglichen Wege müssen am Punkt D beginnen und an E enden bzw. umgekehrt an E beginnen und an D enden. Dies liegt daran, daß von D und E jeweils 5 Kanten ausgehen, also eine ungerade Anzahl, während von allen anderen Punkten eine gerade Anzahl Kanten ausgeht. Bei gerader Anzahl kann man einen Punkt jeweils über eine Kante erreichen und über eine andere wieder verlassen; die Punkte mit ungerader Anzahl Kanten muß man dagegen einmal öfter verlassen als erreichen (also Anfangspunkt eines Weges) oder einmal öfter erreichen als verlassen (Endpunkt). Außerdem ergibt sich jeder Weg von E nach D als Umkehrung eines Weges von D nach E, so daß es genügt, nur D als Startpunkt zu betrachten und jeden gefundenen Weg auch in umgekehrter Richtung auszugeben. Zum Finden der Wege bietet sich eine rekursive Vorgehensweise an. Ausgehend von D gelangt man über eine noch unbenutzte Kante zu einem Punkt, der nun als zweiter Punkt auf dem Weg Ausgangspunkt für die weitere Suche nach einer unbenutzten Kante wird. Wird die Suchprozedur für einen Punkt an Position 16 auf dem Weg aufgerufen, hat man eine Lösung gefunden, die ausgegeben werden kann (und ihre Umkehrung ebenfalls).

## Programmtext:

```

program Nikolaus;

const con : array[1..15] of string[2] =
  ('AB','AC','BC','BD','BE','CD','CE','DE',
  'DF','DG','EF','EG','FG','FH','GH');

var used : array[1..15] of boolean;
    chrs : array[1..16] of char;
    num : integer;

procedure scan(c : char;pos : integer);

var i : integer;

begin
  chrs[pos] := c;
  if pos = 16 then
    begin
      write(num+1,'te Lösung: ');
      for i := 1 to 16 do
        write(chrs[i]);
      write(' ');
      write(num+2,'te Lösung: ');
      for i := 16 downto 1 do
        write(chrs[i]);
      writeln;
      num := num + 2;
    end
  else
    for i := 1 to 15 do
      if not used[i] then
        if con[i,1] = c then
          begin
            used[i] := true;
            scan(con[i,2],pos+1);
            used[i] := false;
          end
        else
          if con[i,2] = c then
            begin
              used[i] := true;
              scan(con[i,1],pos+1);
              used[i] := false;
            end
          end;
        end;
    end;
end;

```

## Programm-Dokumentation:

Im Array „con“ werden die 15 Kanten als Konstante abgespeichert. Welche dieser Kanten auf dem aktuellen Weg bereits benutzt wurden, gibt das Array „used“ an. Die Abfolge der Punkte auf dem aktuellen Weg wird im Array „chrs“ gespeichert:

## Halbformale Programmbeschreibung:

Initialisiere notwendige Variable; suche von Punkt D aus, wobei D an erster Position auf dem Weg auftritt; gib Gesamtzahl der Lösungen aus.

Suche von Punkt X an Position k:  
wenn k = 16:  
gib Folge der Punkte in chrs aus;  
gib Folge der Punkte in chrs in umgekehrter Reihenfolge aus;  
sonst:  
betrachte alle 15 Kanten der Reihe nach:  
wenn Kante noch nicht benutzt:  
der Kante = X:  
suche vom anderen Punkt der Kante an Position k + 1;

## Bildschirmausgabe:

1. Lösung: DBACBEDEFDFGHGE
  2. Lösung: EGHFGDFEDCEBCABD
  3. Lösung: DBACBEDEFDFGHGE
  4. Lösung: EGFHGFEDCEBCABD
- ...
- 13757.te Lösung: DGHGFEDCEDBACBE  
 13758.te Lösung: EBCABDCEDFEGFHGD  
 13759.te Lösung: DGHGFEDCEDBACBE  
 13760.te Lösung: EBACBDCEDFEGFHGD

Anzahl der Lösungen: 13760

