

Musteraufgabe:

Reiterhofprobleme

Leider sind nicht alle Pferde auf dem Reiterhof so brav, daß sie von jedem Anfänger geritten werden können. Einige verlangen gute oder sogar sehr gute Reiter. Nicht genug damit. Die Reiterinnen und Reiter haben auch noch spezielle Wünsche.

Reiterinnen und Reiter:

Anja	Anfängerin, mag Nicki, Pucki, Witthen und Anex
Bertram	guter Reiter ohne Wunsch
Christa	sehr gute Reiterin, bevorzugt Hurrikan
Doris	Anfängerin, mag Anex, Witthen und Pucki
Emil	Anfänger, mag alle Pferde
Fritz	sehr guter Reiter, mag Sturmwind und Hurrikan
Gabi	Ist gut, mag Hurrikan, Anex, Tornado, Sturmwind

Pferde:

Anex	braves Anfängerpferd etwas schlitzhörnig, nur für gute Reiter
Nicki	temperamentvolles Pferd, nur für gute Reiter
Hurrikan	liebes Schmuspferd, für Anfänger geeignet
Pucki	schwierig und ungestum, verlangt sehr gute Reiter
Tornado	liebes Anfängerpferd bockig aber leistungsstark, nur für gute Reiter
Witthen	müde gewordenes altes Pferd, für Anfänger eigenwillig und schwierig, nur für sehr gute Reiter
Sturmwind	
Zausel	
Zickzack	



Lösungsidee:

Es wird versucht, jeden Reiter auf ein Pferd zu setzen. Dabei wird für jedes Paar (Reiter/Pferd) geprüft, ob diese Kombination möglich ist. Aus zwei Gründen kann das Besetzen eines Pferdes j mit einem Reiter i nicht möglich sein:

1. das Pferd j ist schon von einem anderen Reiter besetzt, oder
2. das Können von Reiter i reicht für Pferd j nicht aus.

In beiden Fällen muß versucht werden, ein anderes Pferd für Reiter i zu finden. Hat man eine Möglichkeit gefunden, jeden Reiter auf ein Pferd zu setzen, wird gezählt, wieviele Wünsche bei dieser Zuordnung erfüllt werden. Eine Zuordnung wird dann ausgegeben, wenn sie wenigstens so viele Wünsche erfüllt wie jede der bisher gefundenen Zuordnungen.

Wenn noch nicht alle Kombinationen (Reiter/Pferd) ausprobiert wurden, wird weitergesucht, ob es eine weitere, wenigstens genauso gute Zuordnung wie die zuletzt ausgegebene gibt.

Programm-Text

program Reiterhof:

```

const
  maxPferde = 10;
  maxReiter = 10;

type
  tier = record
    name: string;
    grad: integer;
  end;

  person = record
    name: string;
    koennen: integer;
    wunsch: array [1..maxPferde] of boolean;
  end;

```



```

var
  pferd: array [1..maxPferde] of tier;
  reiter: array [1..maxReiter] of person;
  Reitpferd: array [1..maxReiter] of integer;
  AnzReiter, AnzPferde, MaxWunsch: integer;
  ausgegeben: integer;

```

procedure Eingabe:

```

var
  i, j: integer;
  jaNein: char;

begin
  write('Wieviele Reiter? (max. 10) '); readln(AnzReiter);
  write('Wieviele Pferde? (max. 10) '); readln(AnzPferde);
  for i := 1 to AnzPferde do with pferd[i] do begin
    write('Name von Pferd Nr. ', i, ': '); readln(Name);
    writeln('1 für Anfängerpferd, 2 normales, 3 schwieriges');
    write('Schwierigkeitsgrad: '); readln(Grad);
  end;
  for i := 1 to AnzReiter do with reiter[i] do begin
    write('Name von Reiter Nr. ', i, ': '); readln(Name);
    writeln('1 für Anfänger, 2 guter Reiter, 3 sehr guter');
    write('Können: '); readln(koennen);
    write('Wünsche: ');
    for j := 1 to AnzPferde do begin
      write('Ist ', pferd[j].Name, ' Wunschpferd? (j/n)'); readln(jaNein);
      reiter[i].Wunsch[j] := (jaNein='j');
    end;
  end;
end;
writeln;
for i := 1 to AnzReiter do Reitpferd[i] := 0;
MaxWunsch := 0;
end; (Eingabe)

```



procedure Ausgabe:

```

var i, Wuensche: integer;

begin
  Wuensche := 0;
  for i := 1 to AnzReiter do
    if reiter[i].Wunsch[Reitpferd[i]] then
      Wuensche := Wuensche + 1;
  if Wuensche >= MaxWunsch then begin
    ausgegeben := ausgegeben + 1;
    MaxWunsch := Wuensche;
    for i := 1 to AnzReiter do
      writeln(reiter[i].Name, ' sitzt auf ',
        pferd[Reitpferd[i]].Name);
      writeln('Erfüllte Wünsche: ', Wuensche);
    end;
  end; (Ausgabe)
end; (Ausgabe)

procedure Besetze(i: integer);
var j: integer;

function Besetzbar(i, j: integer): boolean;
var
  moeglich: boolean;
  k: integer;

begin
  moeglich := true;
  for k := 1 to i-1 do
    if Reitpferd[k] = j then moeglich := false;
  if reiter[i].Koennen < pferd[j].Grad then moeglich := false;
  Besetzbar := moeglich;
end; (Besetzbar)

begin (Besetze)
  if i > AnzReiter then Ausgabe
  else begin
    for j := 1 to AnzPferde do
      if Besetzbar(i, j) then begin
        Reitpferd[i] := j;
        Besetze(i-1);
      end;
    end; (Besetze)
  end;
end; (Besetze)

begin
  Eingabe;
  ausgegeben := 0;
  Besetze(1);
  if ausgegeben = 0 then writeln('Schade, keine Lösung!')
  else writeln('ausgegeben, ', ausgegebene Lösungen.);
end.

```

