



Du žirgai

Sprendimas gana paprastas. Reikia tik kruopščiai surasti baltojo žirgo kiekvieno ėjimo (o jų gali būti 8) koordinates ir įsitikinti, ar jos tinka: ar žirgas nenušoka nuo lentos ir ar jo nekerta juodasis.

Visas algoritmas turėtų būti aiškus iš pateiktos programos.

```
program du_žirgai;
  type pozicija = record { padėtis šachmatų lentoje }
                    x, y: char
                    end;

  var b, j,      { baltojo ir juodojo žirgų pradinės koordinatės }
      n: pozicija; { galimas baltojo žirgo ėjimas }
      i: integer;
      rasta: boolean;

  procedure ėjimas (i: integer; { ėjimo numeris }
                    b: pozicija; var n: pozicija);
  { baltojo žirgo ėjimo koordinačių nustatymas }
  begin
    { apskaičiuojama koordinatė x }
    case i of
      1, 4: n.x := succ (b.x);
      2, 3: n.x := succ (succ (b.x));
      5, 8: n.x := pred (b.x);
      6, 7: n.x := pred (pred (b.x))
    end;
    { apskaičiuojama koordinatė y }
    case i of
      2, 7: n.y := succ (b.y);
      1, 8: n.y := succ (succ (b.y));
      3, 6: n.y := pred (b.y);
      4, 5: n.y := pred (pred (b.y))
    end
  end;

  function gerai (p: pozicija): boolean;
  { ar žirgas nenušoka nuo lentos }
  begin
    gerai := (p.x in ['a'..'h']) and
              (p.y in ['1'..'8'])
  end;
```



```
function kerta (j, b: pozicija): boolean;  
{ ar juodasis žirgas kerta baltąjį }  
  var dx, dy: integer;  
begin  
  dx := abs (ord(j.x) - ord(b.x));  
  dy := abs (ord(j.y) - ord(b.y));  
  kerta := (dx = 1) and (dy = 2) or  
           (dx = 2) and (dy = 1)  
end;  
  
begin  
  readln (b.x, b.y, j.x, j.y);  
  rasta := false;  
  for i := 1 to 8 do { išbandomi visi galimi baltojo žirgo ėjimai }  
    begin  
      ėjimas (i, b, n);  
      if gerai (n) and not kerta (j, n)  
        then begin  
          write (n.x, n.y, ' ');  
          rasta := true  
        end  
    end;  
  if not rasta  
    then writeln ('BALTASIS STОВI')  
end.
```