Atspėk skaičių

Žaidžiama dviese. Pirmasis žaidėjas sugalvoja keturženklį skaičių, kurio visi skaitmenys skirtingi. Antrasis spėja – pateikia kokį nors keturženklį skaičių. Tuomet pirmasis pasako, kiek skaitmenų atspėjo antrasis ir kiek iš jų yra savo vietose.

Pavyzdžiui, sugalvotas skaičius 7280.

Antrasis žaidėjas pateikia skaičių 8630.

Pirmasis žaidėjas atsako: atspėti du skaitmenys, iš jų vienas teisingoje vietoje.

Taip žaidžiama tol, kol antrasis žaidėjas atspėja skaičių.

Užduotis. Parašykite algoritmą, kuris atspėtų skaičių, t. y. kad būtų žaidžiama už antrąjį žaidėją. Spėjimų skaičius turi būti minimalus.

Biblioteka. Pirmojo žaidėjo ėjimus atliks kompiuteris. Abu bendrauja modulio `ZAIDIMAS` procedūros `ejimas` pagalba. Ši procedūra turi vieną pradinį duomenį – sveikąjį keturženklį skaičių ir du rezultatus: atspėtų skaitmenų skaičių ir atspėtų savo vietose esančių skaitmenų skaičių.

Pastaba. Moksleiviams buvo pateikta viena modulio `ZAIDIMAS` versija, testavimui buvo naudojama kita.

Modulio `ZAIDIMAS` pavyzdys:

```
unit žaidimas;
interface
  procedure ėjimas (sk: integer; var atsp, vietoj: integer);
    {gavusi skaičių, procedūra palygina jį su sugalvotu }
    { skaičiumi ir pasako kiek skaitmenų atspėta ir kiek iš jų }
    { yra savo vietose }
implementation
  var ska: string;          { sugalvotas skaičius }
      aibė: set of char;   { sugalvoto skaičiaus skaitmenų aibė }
      skt: longint;        { spėjimų skaičius }
  procedure ėjimas (sk: integer; var atsp, vietoj: integer);
    var s: string; { spėjamas skaičius, paverstas eilute }
        i: integer;
  begin
    skt := skt + 1;
    if skt = 100     { jei viršijamas spėjimų limitas }
       then begin
              writeln ('Viršytas spėjimų limitas - 100'); halt
            end;
    if (sk < 1000) or (sk > 9999)
       then begin
              write ('Bandyta spėti ne keturženklį skaičių: ');
              writeln (sk); halt
```

```
            end;
      str (sk, s);  { bus paprasčiau lyginti dvi eilutes }
      if s = ska    { jei skaičius atspėtas }
         then begin
              writeln ('Atspėta per ', skt, ' kartų.');  halt
              end
      else begin  { rasime kiek skaitmenų savo vietoje ir kiek atspėta }
             vietoj := 0;
             for i := 1 to 4 do
                 if s[i] = ska[i]
                    then vietoj := vietoj + 1;
             atsp := 0;
             for i := 1 to 4 do
                 if s[i] in aibė
                    then atsp := atsp + 1
          end;
   end; { ėjimai }
   var reikšmė, kl_kodas, elem_sk: integer;
       c: char;
begin
   writeln ('Tai yra demonstracinis modulis.');
   writeln ('Sugalvokite keturženklį skaičių patys: ');
   readln (ska);
   reikšmė := 0; { įvestas skaičius verčiamas simbolių eilute }
   val (ska, reikšmė, kl_kodas);
   if (kl_kodas <> 0)  or                  { jei skaičius įvestas klaidingai }
      (reikšmė < 1000) or (reikšmė > 9999)
      then begin
           writeln ('Sugalvojote ne keturženklį skaičių.');
           halt
         end;
   aibė := [ska[1]] + [ska[2]] + [ska[3]] + [ska[4]];
   elem_sk := 0;
   for c := '0' to '9' do
       if c in aibė
          then elem_sk := elem_sk + 1;
   if elem_sk <> 4
      then  begin
          writeln ('Skaičiaus skaitmenys turi būti skirtingi.');
           halt
       end;
end.
```

Pradiniai duomenys. Jokių duomenų įvesti ar išvesti nereikia. Modulis sustabdys programą, kai bus atspėtas teisingas skaičius. Modulis `ZAIDIMAS` pateiks rezultatą – kiek kartų buvo kreiptasi į procedūrą `ejimas` – tai bus spėjimų skaičius.