Abėcėlė

Tai trumpas ir paprastas uždavinio sprendimas, kuris turėtų būti aiškus iš paties algoritmo. Ši programa labai tiktų tiems, kas pradeda mokytis rekursiją ir nori ją geriau perprasti. Rekursinė procedūra skirta visiems galimiems žodžiams formuluoti.

```
program abėcėlė;
  const max = 1000;   { maksimalus sudaromų žodžių skaičius }
        MILG = 25;    { maksimalus sudaromų žodžių ilgis }
  type žodis = array [1..MILG] of char;
  var abc : string; { abėcėlės simboliai }
      žod: žodis;
      f : text;

  procedure sudaryti_žodžius (n, k, poz: integer);
  { rekursinė procedūra, formuojanti žodžius nuo poz pozicijos }
  { abc bei žod – globalieji }
    var i: integer;
  begin
    if poz = k + 1     { jei žodis jau sudarytas }
       then begin { spausdiname jį }
              for i := 1 to k do
                write (f, žod[i]);
              writeln (f);
            end
       else for i := 1 to n do   { formuojame žodį }
              begin
                žod[poz] := abc[i];
                sudaryti_žodžius (n, k, poz+1, kiek)
              end
  end; { sudaryti_žodžius }

  var n, k, i : integer;
begin
  assign(f, 'ABC.DAT');
  reset(f);
  readln(f, n);
  for i: = 1 to n do
    readln(f, abc[i]);   { skaitoma abėcėlė }
  readln(f, k);
  close(f);
  assign(f, 'ABC.REZ');
  rewrite(f);
```

```
  žodis(n, k, 1);
  close(f);
end.
```

Jei pradiniai duomenys būtų gerokai didesni ir visi surasti žodžiai netilptų į rezultatų masyvą, tuomet būtų galima radus žodį iš karto jį rašyti į bylą, o ne į rezultatų masyvą.