```
{
projeto numbers - 2024 (fragmento)
Projeto de estudo de algoritmos
Esta obra está licenciado sob Licença Creative Commons Atribuição-CompartilhaIgual 4.0
Internacional.
Veja os termos desta licença na página https://creativecommons.org/licenses/by-sa/4.0/
deed.pt_BR
Copyleft 2024-1 - Evandro Guglielmeli
versão 0.0.0
}

program numbers;

{
    Constantes para ajuste da capacidade do programa conforme o número de dígitos
    recomenda-se limitar o número de dígitos à faixa entre 4 e 9
    4      24      '000'
    5     120     '0000'
    6     720     '00000'
    7    5040    '000000'
    8   40320   '0000000'
    9  362880  '00000000'
   10 3628800 '000000000'
}
const MAXDIG = 4;
      MAXARR = 24;
      ZPREFIX = '000';

// declarações prévias da MV do programa
type placa = array [1..MAXDIG] of integer;

var arrplaca : array [1..MAXARR] of string;
    linha : integer;

{
    guarda uma série no vetor de séries (arrplaca)
    método insertion sort
}
procedure guarda(serie : string);

var n, l : integer;

begin
    l := linha;
    n := l - 1;    // aponta para a posição antes da primeira posição vazia
    if l > 1 then // procura o lugar se já tem alguma série no vetor
    begin
        repeat
            if arrplaca[n] > serie then // está fora de ordem?
            begin
                arrplaca[l] := arrplaca[n]; // "puxa" a série para "cima"
                l := n; // troca ponteiros para "baixo"
                n := l - 1;
            end
            else
                break; // sai do laço, "n" aponta uma posição abaixo de
                        // onde a serie deve ser colocada no vetor }
        until n = 0;
    end;

```

```
n := n + 1; // corrige posição de n
arrplaca[n] := serie; // guarda a serie na posição apontada por n
linha := linha + 1; // incrementa contador de linhas
end;

// rotina para embaralhar os dígitos dados pelo operador
procedure mistura(input : placa; index : integer); // parâmetro
// input tem a série de dígitos da placa
// index indica a posição que inicia o arranjo

var output : placa;
p, i : LongInt;
keep, c : string;

begin
  if index=MAXDIG then
    begin
      keep := ''; c := '';
      p := 0;
      for i := 1 to MAXDIG do
        p := p * 10 + input[i];
      str(p,keep);
      keep := ZPREFIX + keep;
      p := length(keep)-(MAXDIG-1);
      c := copy(keep,p,MAXDIG);
      {
        substitui estas duas linhas pela chamada à procedure guarda()
      }
      //arrplaca[linha] := c;
      //linha := linha + 1;
      guarda(c);
    end
  else
    begin
      output := input;
      for p := index to MAXDIG do
        begin
          output[index] := input[p];
          for i := index to p - 1 do
            output[i + 1] := input[i];
          for i := p + 1 to MAXDIG do
            output[i] := input[i];
          mistura(output,index + 1); // argumento
        end;
    end;
  end;

// o programa principal
// versão 0.0.0

var entrada : string;
input : placa;
v, w, //para uso no procedimento val()
p, i : integer; //para controle de laços

begin
  // iniciar a iteração do algoritmo
  repeat
    write('Entre o número da placa, na forma de uma sequência de ',MAXDIG,' dígitos: ');
```

```
readln(entrada);
if entrada='fim' then
  break;
{
  se compr(entrada) = MAXDIG então
  executa o abaixo
}
if length(entrada) = MAXDIG then
begin
  for i := 1 to MAXDIG do
    begin
      val(entrada[i],v,w);
      if (w=0) then
        input[i] := v
      else
        break;
    end;
  if (w<>0) then
    begin
      writeln;
      writeln('Apenas dígitos de 0 a 9 são aceitos...');
    end
  else
    begin
      linha := 1;
      mistura(input,1); // argumentos
      for p := 1 to MAXARR do
        begin
          writeln(arrplaca[p]);
        end;
    end;
  end;
until FALSE;
end.
```