Solutions écrites en Pascal (utilisant Delphi avec "Console Application")

I. Superdiv — 20 points

```
program SuperDiv;
{$APPTYPE CONSOLE}
uses
  SysUtils;

var i1,i2,i3,i4,i6,i7,i8,i9 : integer;

function found(a,b,c,d,e,f,g,h : integer) : boolean;
var TEMP : boolean;
begin
   TEMP := true;
   TEMP := TEMP and ((100*a + 10*b + c) mod 3 = 0);
   TEMP := TEMP and ((1000*a + 100*b + 10*c + d) mod 4 = 0);
   TEMP := TEMP and ((100000*a + 10000*b + 1000*c + 100*d + 50 + e) mod 6 = 0);
   TEMP := TEMP and ((1000000*a + 100000*b + 10000*c + 1000*d + 500 +
                                                        10*e + f) mod 7 = 0);
   TEMP := TEMP and ((10000000*a + 1000000*b + 100000*c + 10000*d +
                                             5000 + 100*e + 10*f + g) mod 8 = 0);
   TEMP := TEMP and (a <> b) and (a <> c) and (b <> c);
   TEMP := TEMP and (a <> d) and (b <> d) and (c <> d);
   TEMP := TEMP and (a <> e) and (b <> e) and (c <> e) and (d <> e);
   TEMP := TEMP and (a <> f) and (b <> f) and (c <> f) and (d <> f) and
                    (e <> f);
   TEMP := TEMP and (a <> h) and (b <> h) and (c <> h) and (d <> h) and
                    (e <> h) and (g <> h) and (f <> h);
   TEMP := TEMP and (a <> g) and (b <> g) and (c <> g) and (d <> g) and
                    (e <> g) and (g <> f);
   TEMP := TEMP and (a <> 5) and (b <> 5) and (c <> 5) and (d <> 5);
   TEMP := TEMP and (e <> 5) and (f <> 5) and (g <> 5) and (h <> 5);
   found := TEMP
end;

begin
  i1 := 1;
  while i1 <= 9 do
     begin
        i2 := 2;
        while i2 < 9 do
           begin
              i3 := 1;
              while i3 <= 9 do
                  begin
                     i4 := 2;
                     while i4 < 9 do
                        begin
                           i6 := 2;
                           while i6 < 9 do
```

```
                                begin
                                   i7 := 1;
                                   while i7 <= 9 do
                                      begin
                                         i8 := 2;
                                         while i8 < 9 do
                                            begin
                                               i9 := 1;
                                               while i9 <= 9 do
                                                  begin
                                                    if
                                                    found(i1,i2,i3,i4,i6,i7,i8,i9)
                                                    then writeln
                                                      (i1,i2,i3,i4,5,i6,i7,i8,i9);
                                                    i9 := i9 + 1
                                                  end;
                                               i8 := i8 + 2
                                            end;
                                         i7 := i7 + 1
                                      end;
                                   i6 := i6 + 2
                                end;
                             i4 := i4 + 2
                          end;
                       i3 := i3 + 1
                    end;
                 i2 := i2 + 2
              end;
           i1 := i1 + 1
        end;
     readln
end.
```

SuperDiv = 381654729

Pour satisfaire la contrainte des 5 secondes accordées, le chiffre du milieu doit être le 5 et les chiffres à des emplacements pairs doivent être pairs !

II. LookAndSay — 20 points

```
program LookAndSay;
{$APPTYPE CONSOLE}
uses SysUtils;

var STR,STR2,CH : string;
    I,J,Z,N : integer;

begin
  readln(N);
  STR := '1';
  writeln(STR);
  for J := 2 to N do
     begin
        STR2 := '';
        while length(STR) > 0 do
           begin
              CH := copy(STR,1,1);
              I := 1;
              Z := 0;
              while copy(STR,I,1) = CH do
                 begin
                    Z := Z + 1;
                    I := I + 1
                 end;
```

```
              STR2 := STR2 + inttostr(Z) + CH;
              delete(STR,1,Z);
            end;
         writeln(STR2);
         STR := STR2
      end;
  readln
end.
```

III. Sudoku 60 points

```
program Sudoku;            (* ce n'est que du: backtracking ! *)
{$APPTYPE CONSOLE}
uses SysUtils;

type t_su = array [1..9,1..9] of
               record
                  zahl : integer;
                  fix : boolean
               end;

var su : t_su;
    f,ff : text;

procedure LESE (var su : t_su);
var i,j : integer;
begin
   assign(f,'SUDO_IN.TXT');
   reset(f);
   for i := 1 to 9 do
      for j := 1 to 9 do
         begin
            read(f,su[i,j].zahl);
            if su[i,j].zahl = 0
            then su[i,j].fix := false
            else su[i,j].fix := true
         end;
   close(f)
end;

procedure AUSGABE (su : t_su);
var i,j : integer;
begin
   for i := 1 to 9 do
      begin
         for j := 1 to 9 do
            write(ff,su[i,j].zahl,' ');
         writeln(ff)
      end;
   writeln(ff)
end;

function OK (r,c,i : integer) : boolean;
var TEMP : boolean;
    x,y,boxx,boxy : integer;
begin
   TEMP := true;
   (* row *)
   for x := 1 to 9 do
```

```
      if x <> r
      then if su[x,c].zahl = i
           then TEMP := false;
   (* col *)
   for x := 1 to 9 do
      if x <> c
      then if su[r,x].zahl = i
           then TEMP := false;
   (* 3x3 box *)
   if r <= 3
   then boxx := 1
   else if r <= 6
        then boxx := 4
        else boxx := 7;
   if c <= 3
   then boxy := 1
   else if c <= 6
        then boxy := 4
        else boxy := 7;
   for x := boxx to boxx + 2 do
      for y := boxy to boxy + 2 do
         if (x <> r) and (y <> c)
         then if su[x,y].zahl = i
              then TEMP := false;
   OK := TEMP and (c <= 9) and (c > 0)
end;

function SUDOOK : boolean;
var TEMP : boolean;
    r,c : integer;
begin
   TEMP := true;
   for c := 1 to 9 do
      for r := 1 to 9 do
         begin
            if su[r,c].zahl <> 0
            then begin
                    if not OK(r,c,su[r,c].zahl)
                    then TEMP := false
                 end
         end;
   SUDOOK := TEMP
end;

function DONE : boolean;
var i,j : integer;
begin
   DONE := true;
   for i := 1 to 9 do
      for j := 1 to 9 do
         if su[i,j].zahl = 0
         then DONE := false
end;

procedure TRYOUT (r,c : integer);
var i : integer;
begin
   for i := 1 to 9 do
      if OK (r,c,i)
      then begin
              if not su[r,c].fix
              then su[r,c].zahl := i;
              if DONE
              then begin
```

on place un chiffre

```
                    AUSGABE (su);
                    exit
                 end;
              repeat
                 r := r + 1;
                 if r > 9
                 then begin
                         c := c + 1;
                         r := 1
                      end;
              until not su[r,c].fix;
              TRYOUT (r,c);
              repeat
                 r := r - 1;
                 if r < 1
                 then begin
                         c := c - 1;
                         r := 9
                      end;
              until not su[r,c].fix;
              su[r,c].zahl := 0
           end
end;

begin
   LESE(su);
   assign(ff,'SUDO_OUT.TXT');
   rewrite(ff);
   if SUDOOK
   then TRYOUT(1,1)
   else writeln(ff,'No solution');
   close(ff)
end.
```

appel récursif

le chiffre placé nous a conduit dans cul-de-sac, on l'enlève et on essaye le chiffre suivant

Tests et résultats:

	SUDO_IN	SUDO_OUT
Test No 1 (exemple donné)	0 0 0 0 0 5 4 9 0 3 4 6 0 0 0 0 8 0 9 0 0 0 0 6 0 3 0 7 0 8 0 3 0 0 0 0 0 0 0 5 0 1 0 0 0 0 0 0 0 6 0 3 0 4 0 6 0 2 0 0 0 0 5 0 8 0 0 0 0 1 6 3 0 1 3 6 0 0 0 0 0	8 2 1 3 7 5 4 9 6 3 4 6 1 2 9 5 8 7 9 7 5 8 4 6 2 3 1 7 5 8 4 3 2 6 1 9 6 3 4 5 9 1 7 2 8 1 9 2 7 6 8 3 5 4 4 6 9 2 1 3 8 7 5 2 8 7 9 5 4 1 6 3 5 1 3 6 8 7 9 4 2
Test No 2 (impossible)	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	Pas de solution

Test No 3 (que des zéros)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 4 7 2 3 8 5 6 9 2 5 8 1 6 9 3 4 7 3 6 9 4 5 7 1 2 8 4 7 1 3 8 2 6 9 5 5 8 2 6 9 1 4 7 3 6 9 3 5 7 4 2 8 1 7 1 4 8 2 3 9 5 6 8 2 5 9 1 6 7 3 4 9 3 6 7 4 5 8 1 2
Test No 4 (exemple facile)	0 7 0 6 0 2 0 4 0 0 0 0 0 5 0 0 0 0 3 0 1 0 0 0 6 0 5 9 6 0 0 1 0 0 3 7 0 5 0 7 0 9 0 1 0 1 2 0 0 4 0 0 8 9 6 0 8 0 0 0 3 0 2 0 0 0 0 7 0 0 0 0 0 9 0 8 0 3 0 5 0	5 7 9 6 3 2 8 4 1 2 8 6 1 5 4 7 9 3 3 4 1 9 8 7 6 2 5 9 6 4 5 1 8 2 3 7 8 5 3 7 2 9 4 1 6 1 2 7 3 4 6 5 8 9 6 1 8 4 9 5 3 7 2 4 3 5 2 7 1 9 6 8 7 9 2 8 6 3 1 5 4
Test No 5 (exemple moyen)	0 0 0 0 0 0 0 9 0 5 0 0 0 0 0 0 6 0 0 9 0 0 0 0 0 0 2 2 3 0 0 1 0 0 0 6 1 0 0 0 8 0 0 0 3 4 0 0 0 0 7 0 0 5 0 0 0 0 0 9 0 0 0 0 8 1 2 7 0 0 0 0 0 2 7 0 0 6 0 5 0	8 4 2 7 6 3 5 9 1 5 1 3 9 2 8 4 6 7 7 9 6 4 5 1 3 8 2 2 3 9 5 1 4 8 7 6 1 7 5 6 8 2 9 4 3 4 6 8 3 9 7 2 1 5 6 5 4 1 3 9 7 2 8 9 8 1 2 7 5 6 3 4 3 2 7 8 4 6 1 5 9
Test No 6 (exemple difficile)	0 0 0 0 0 3 0 0 0 0 0 0 4 0 0 0 0 0 8 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 3 4 0 9 0 0 7 1 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 1 8 0 2 0 0 2 5 0 0 6 0 0 0 0	4 1 9 8 7 3 5 6 2 5 6 2 4 9 1 7 8 3 8 7 3 2 5 6 4 9 1 6 2 7 3 4 8 1 5 9 1 8 5 6 2 9 3 4 7 9 3 4 7 1 5 6 2 8 7 4 8 5 3 2 9 1 6 3 9 6 1 8 4 2 7 5 2 5 1 9 6 7 8 3 4