CIL 2009 • Epreuve de sélection préliminaire - Solutions modèles

Problème I - TROIS CHINOIS — 15 points

```
program CHINOIS;

var CH : char;
    STRIN,STROUT : string;
    I : integer;

begin
   readln(CH);
   STRIN := 'Drei Chinesen mit dem Kontrabass ' +
            'sassen auf der Strasse und erzaehlten sich was';
   STROUT := '';
   for I := 1 to length(STRIN) do
      if STRIN[I] in ['a','e','i','o','u']
      then STROUT := STROUT + CH
      else STROUT := STROUT + STRIN[I];
   writeln(STROUT);
   readln
end.
```

Problème II - CALCULS — 10 points

```
program CALCULS;

var N,I : integer;

begin
   readln(N);
   writeln(' x    x^2     x^3');
   writeln('--    ---    ----');
   writeln;
   for I := 1 to N do
      writeln(I:2,' ',I*I:6,' ',I*I*I:7);
   readln
end.
```

Problème III - NOMBRES REVERSIBLES — 25 points

```
program NOMBRES;

var NOMBRE,M : integer;

function INV (N : integer) : integer;
var TEMP : integer;
begin
   TEMP := 0;
   while N > 0 do
      begin
         TEMP := TEMP * 10 + N mod 10;
         N := N div 10
      end;
   INV := TEMP
end;

function REV (NOMBRE,MULT : integer) : boolean;
begin
   REV := NOMBRE*MULT = INV(NOMBRE)
end;
```

```
begin
  readln(M);
  for NOMBRE := 10 to 100000000 do
     if REV(NOMBRE,M)
     then writeln(NOMBRE,' ',M);
  readln
end.
```

Problème IV - CONVERSION 30 points

```
program CONV;

var { GZ Ganze Zahl, NP nicht-periodisch, P periodisch }
    DEZ,GZSTR,NPSTR,PSTR : string;
    NENNERNPSTR,NENNERPSTR : string;
    GZ,NP,P : int64;
    ZAEHLERNP,NENNERNP,ZAEHLERP,NENNERP : int64;
    RESZAEHLER,RESNENNER : int64;
    I : integer;

procedure ZERLEGE(DEZ : string; var GZSTR,NPSTR,PSTR : string);
begin
   GZSTR := copy(DEZ,1,1);
   delete(DEZ,1,2);
   NPSTR := copy(DEZ,1,pos('p',DEZ)-1);
   delete(DEZ,1,pos('p',DEZ));
   PSTR := DEZ
end;

procedure KUERZE (var A,B : int64);
var P : int64;
   function PGCD (A,B : int64) : int64;
   var TEMP : int64;
   begin
      while B > 0 do
         begin
            TEMP := A;
            A := B;
            B := TEMP mod B
         end;
      PGCD := A
   end;
begin
   P := PGCD(A,B);
   A := A div P;
   B := B div P
end;

procedure ADDIERE(A,B,C,D : int64; var RZ,RN : int64);
begin
   RZ := A*D + B*C;
   RN := B*D;
   KUERZE(RZ,RN)
end;

begin
  readln(DEZ);
  ZERLEGE(DEZ,GZSTR,NPSTR,PSTR);
  GZ := strtoint(GZSTR);
  NP := strtoint(NPSTR);
  P := strtoint(PSTR);
  {writeln(GZ,' ',NP,' ',P);}
  ZAEHLERNP := NP;
  NENNERNPSTR := '1';
  for I := 1 to length(NPSTR) do
     NENNERNPSTR := NENNERNPSTR + '0';
  NENNERNP := strtoint(NENNERNPSTR);
```

```
  KUERZE(ZAEHLERNP,NENNERNP);
  {writeln(ZAEHLERNP,'/',NENNERNP);}
  ZAEHLERP := P;
  NENNERPSTR := '';
  for I := 1 to length(PSTR) do
     NENNERPSTR := NENNERPSTR + '9';
  for I := 1 to length(NPSTR) do
     NENNERPSTR := NENNERPSTR + '0';
  NENNERP := strtoint(NENNERPSTR);
  KUERZE(ZAEHLERP,NENNERP);
  {writeln(ZAEHLERP,'/',NENNERP);}
  ADDIERE(ZAEHLERNP,NENNERNP,ZAEHLERP,NENNERP,RESZAEHLER,RESNENNER);
  {writeln(RESZAEHLER,'/',RESNENNER);}
  ADDIERE(GZ,1,RESZAEHLER,RESNENNER,RESZAEHLER,RESNENNER);
  writeln(RESZAEHLER,'/',RESNENNER);
  readln
end.
```

Problème V - CALENDRIER — 20 points

```
program CAL;
//-----------------------------------------------------------------
// Calender display program
// Given a month (1-12) and a year (1901-2999), display the
// corresponding month
// Convention: Throughout the program each day has a specific code:
// 0=Sunday, 1=Monday, 2=Tuesday...6=Saturday
//-----------------------------------------------------------------

var YEAR, MONTH : integer;

const kJAN_FIRST_1901 = 2;  // Tuesday

procedure WRITE_HEADER;
begin
  writeln('Dim Lun Mar Mer Jeu Ven Sam')
end;

procedure WRITE_SPACES(COUNT : integer);
var J : integer;
begin
  for J := 1 to COUNT do
    write('    ')
end;

procedure WRITE_MONTH(DAY_COUNT, DAY_START : integer);
//-------------------------------------------------------------
// DAY_COUNT  : number of days for month
// DAY_START  : first day in month (0=sunday, 6=monday)
// Application globals: NONE
//-------------------------------------------------------------
var I, COL : integer;
begin
  WRITE_HEADER;
  WRITE_SPACES(DAY_START);
  COL:=DAY_START;
  for I := 1 to DAY_COUNT do
    begin
      write(I:3,' ');
      COL := COL + 1;
      if COL > 6
      then
        begin
          writeln;
          COL := 0
        end
    end;
```

```
  if COL <> 0
  then
    writeln
end;

function IS_LEAP_YEAR(WHICH_YEAR: integer): boolean;
//--------------------------------------------------------------
// Return TRUE  if WHICH_YEAR is     a leap year.
// Return FALSE if WHICH_YEAR is not a leap year
// WHICH_YEAR     : 1901-2999
// IS_LEAP_YEAR   : TRUE / FALSE
// Application globals: NONE
//--------------------------------------------------------------
var FLAG : boolean;
begin
  if WHICH_YEAR mod 4 = 0
  then
    if WHICH_YEAR mod 100 = 0
    then
      if WHICH_YEAR mod 400 = 0
      then
        FLAG := TRUE
      else
        FLAG:=FALSE
    else
      FLAG:=TRUE
  else
    FLAG:=FALSE;
  IS_LEAP_YEAR := FLAG   // return value to caller
end;

function DAYS_FOR_MONTH(WHICH_MONTH, WHICH_YEAR: integer): integer;
//--------------------------------------------------------------
// Return the number of days in month
// WHICH_MONTH    : 1-12
// WHICH_YEAR     : 1901-2999
// DAYS_FOR_MONTH : 28-31
// Application globals: NONE
//--------------------------------------------------------------
var DAYS : integer;
begin
  DAYS:= 0; // to prevent from compiler warning
  case WHICH_MONTH of
    4, 6, 9, 11           : DAYS := 30;
    1, 3, 5, 7, 8, 10, 12 : DAYS := 31;
    2                     : if IS_LEAP_YEAR(WHICH_YEAR)
                            then
                              DAYS := 29
                            else
                              DAYS := 28
  end; // case
  DAYS_FOR_MONTH := DAYS // return value to caller
end;

function NEWYEARS_DAY(WHICH_YEAR: integer): integer;
//--------------------------------------------------------------
// Determine the day for january 1st for given year. Return the
// corresponding day code (0=Sunday 6=Monday)
// WHICH_YEAR   : 1901-2999
// NEWYEARS_DAY : 0-6
// Application globals: NONE
//--------------------------------------------------------------
var NON_LEAPYEAR_COUNT,  // number of non-leap years from 1901 to WHICH_YEAR-1
    LEAPYEAR_COUNT,      // number of leap years
    I                 : integer;
begin
  NON_LEAPYEAR_COUNT  := 0;
  LEAPYEAR_COUNT      := 0;
  //-----------------------------------------------------------------------------
```

```
  // Walk throught the different years and count the number of
  // leap years and the number of non-leap years. We take not WHICH_YEAR into
  // consideration since we check for Jan. 1st and do not need to know
  // whether WHICH_YEAR is a leap year or not, so that's why we loop up to
  // WHICH_YEAR-1
  //---------------------------------------------------------------------------
  for I:= 1901 to WHICH_YEAR-1 do
     if IS_LEAP_YEAR(I)
     then
       LEAPYEAR_COUNT:=LEAPYEAR_COUNT + 1
     else
       NON_LEAPYEAR_COUNT := NON_LEAPYEAR_COUNT + 1;
  //---------------------------------------------------------------------------
  // Implementor's note:
  // -------------------
  // For non-leap years, if a given date xx.yy.zzzz is a Tuesday, the following
  // year, xx.yy.(zzzz+1) will become a Wednesday   (+1)
  // For leap years, if a given date xx.yy.zzzz is a Tuesday, the following will
  // year, that date will become a Thursay (+2)
  // This note should explain the next line
  //---------------------------------------------------------------------------
  NEWYEARS_DAY := (kJAN_FIRST_1901 + NON_LEAPYEAR_COUNT + 2*LEAPYEAR_COUNT) mod 7
end;

function FIRST_OF_MONTH(WHICH_MONTH, WHICH_YEAR: integer): integer;
//-------------------------------------------------------------
// Determine the day for the 1st for given year and month. Return the
// corresponding day code (0=Sunday 6=Monday)
// WHICH_MONTH    : 1-12
// WHICH_YEAR     : 1901-2999
// FIRST_OF_MONTH : 0-6
// Application globals: NONE
//-------------------------------------------------------------
var NUMBER_OF_DAYS,I : integer;
begin
  // Check the day code for jan 1 of given year
  NUMBER_OF_DAYS := NEWYEARS_DAY(WHICH_YEAR);
  // Count the number of days from jan 1st up to previous month
  for I := 1 to WHICH_MONTH-1 do
    NUMBER_OF_DAYS := NUMBER_OF_DAYS + DAYS_FOR_MONTH(I, WHICH_YEAR);
  // Get the day code
  FIRST_OF_MONTH := NUMBER_OF_DAYS mod 7
end;

begin  // Main program
  write('Enter month (1-12): ');
  readln(MONTH);
  write('Enter year (1901-2999): ');
  readln(YEAR);
  writeln;
  WRITE_MONTH(DAYS_FOR_MONTH(MONTH, YEAR),FIRST_OF_MONTH(MONTH, YEAR));
  readln // Wait for the user to press ENTER
end.
```