

```

Begin
    (* Ecriture du texte de l'ecran predefini *)
    Error:=0;
    T.nom:='';
    T.prenom:='';
    T.rue:='';
    T.ville:='';
    T.cp:=0;
    T.pays:='';
    T.telephone:='';
    ClrScr;
    LowVideo;
    WriteLn('-----');
    WriteLn(' SAISIE CARNET D'ADRESSE          Entrez les informations demandees');
    WriteLn('-----');
    WriteLn('');
    WriteLn(' Nom   : [          ]');
    WriteLn('-----');
    WriteLn('');
    WriteLn(' Prenom : [          ]');
    WriteLn('');
    WriteLn(' Rue   : [          ]');
    WriteLn('');
    WriteLn(' Ville : [          ]');
    WriteLn('');
    WriteLn(' Code postal : [    ]');
    WriteLn('');
    WriteLn(' Pays   : [          ]');
    WriteLn('');
    WriteLn(' Telephone : [      ]');
    WriteLn('-----');
    WriteLn(' RETURN: change de zone  BACKSPACE: correction  ESC: finir saisie');
    WriteLn('-----');

    (* gestion des zones ecran *)
    NrVar:=1; Buf1:=''; Ion:=0; Flag:=false;
    Repeat
        Ch_Zon;
        if Error=0 then begin
            NrVar:=NrVar+1;
            gotoxy(1,25);write(' ');
        end else begin
            gotoxy(1,25);write(' SAISIE INVALIDE RECOMMENCEZ!'); end;
        if NrVar=8 then NrVar:=1;
        if NrVar=0 then NrVar:=8;
        Ch_Zon; Buf1:=Buf2; Cur:=1; Ion:=0;
        gotoxy(X,Y);for i:=1 to L do write(' ');
        Repeat
            LowVideo; gotoxy(X,Y); write(Buf1); gotoxy(Cur,Y);
            Repeat Until keypressed; Read(Kbd,Car);
            if (Car=#27) and (not(keypressed)) then Flag:=true
            else if Car=#13 then Ion:=1
            else if Car=#8 then begin
                Cur:=Cur-1; if X>Cur then Cur:=1; end
            else
                if (Car<chr(31)) and (Car<chr(123)) then begin
                    Buf2:=Buf1; Buf1:=copy(Buf2,1,(Cur-1))+Car+copy(Buf2,(Cur-X+2),L);
                    Cur:=Cur+1;
                end;
                if (Cur-X)>L then Ion:=1;
            until Flag or (Ion<0);
        until Flag;
        Ch_Zon;
        ClrScr;
    End;
    (* Genere par programme TURBO_MAKER L.Dumont *)
    (* Definition du type employe par la procedure: Exemple *)
    Type VExemple=Record
        cp : Integer;
        nom : String[23];
        prenom : String[21];
        rue : String[35];
        ville : String[23];
        pays : String[19];
        telephone : String[13];
    End;
    Procedure Exemple(Var T: VExemple);
        (* Masque d'ecran cree par generateur TURBO_MAKER *)
        Var Ion,X,Y,Cur,L,NrVar,Error,i : Integer;
            Buf1,Buf2 : String[79];
                Flag : Boolean;
                    Car : Char;
        Procedure Ch_Zon;
            (* gestion des zones en entree sur ecran *)
            Begin
                Case NrVar of
                    1 : begin
                            Buf2:=T.nom;
                            T.nom:= copy(Buf1,1,23);
                            X :=27;
                            Y :=5;
                            L :=23;
                        end;
                    2 : begin
                            Buf2:=T.prenom;
                            T.prenom:= copy(Buf1,1,21);
                            X :=30;
                            Y :=10;
                            L :=21;
                        end;
                    3 : begin
                            Buf2:=T.rue;
                            T.rue:= copy(Buf1,1,35);
                            X :=30;
                            Y :=12;
                            L :=35;
                        end;
                    4 : begin
                            Buf2:=T.ville;
                            T.ville:= copy(Buf1,1,23);
                            X :=30;
                            Y :=14;
                            L :=23;
                        end;
                    5 : begin
                            Str(T.cp,Buf2);
                            Val(Buf1,T.cp,Error);
                            X :=30;
                            Y :=16;
                            L :=5;
                        end;
                    6 : begin
                            Buf2:=T.pays;
                            T.pays:= copy(Buf1,1,19);
                            X :=30;
                            Y :=18;
                            L :=19;
                        end;
                    7 : begin
                            Buf2:=T.telephone;
                            T.telephone:= copy(Buf1,1,13);
                            X :=50;
                            Y :=20;
                            L :=13;
                        end;
                end;
            end;
        End;
    End;

```

Fig. 2. - Liste Pascal du programme generé par Turbo-Maker, avec le masque de la figure 1.

```

Program TURBO_MAKER;      (* générateur de masques de saisie en et pour Turbo-Pascal *)

(* Dernière mise à jour : 27.09.86 *)
(* Matériel : MS-DOS,CPMB6,CPM/80 avec Turbo Pascal *)
(* Auteur : Laurent Dumont *)

Type
  Messa = String[30];      (* enregistrement des information de chaque variable *)
  TypTVar = Record
    Px,Py,Lg,Typ : Integer;
    InOut : Boolean;
    Nvar : String[15];
  End;

Var
  ATVar : Array[1..20] of TypTVar;      (* Tableau d'informations sur les variables *)
  TEnt,TTex : Array[1..25] of String[80]; (* contenu des textes a afficher *)
  TVar : TypTVar;
  CarLu : Char;
  NomEcr,NomFichier : String[30];
  Error,i,j,Lo,NbVar,
  NbTex,CurX,CurY : Integer;
  OK,Fanal : Boolean;
  Fi,Fo : Text;      (* fichiers texte entrée:Fi, sortie:Fo *)

Procedure anal_ecran;      (* ----- analyse du contenu de l'écran saisi ----- *)
Var
  FlagTyp,FVar,FText,FTex2 : Boolean;
  TChain,Vide,Chein,Chaine : String[80];
  Lon,NbBlan : integer;

procedure init_anal_var;      (* ----- initialise l'analyse d'une variable ----- *)
begin
  fvar:=true;      (* drapeau de détection d'une variable *)
  ftext:=false;      (* interdit reconnaissance comme texte *)
  FlagTyp:=false;      (* pas de reconnaissance de chgmt de type de variable *)
  TVar.px:=curx+1;
  TVar.py:=cury;
  TVar.Type:=1;      (* initialise variable type=caractere *)
  TVar.InOut:=true;      (* initialise variable en entrée *)
  NbVar:=NbVar+1;
  If TVar.lg<>0 then error:=3;      (* erreur de saisie variable sur 2 lignes *)
  TVar.Nvar:='';
  Chaine:='';
end;      (* ----- endproc init_anal_var ----- *)

procedure anal_var;      (* ----- reconnait et enregistre une var ----- *)
begin
  TVar.lg:=TVar.lg+1;
  if CarLu <> chr(32) then
  begin
    if FlagTyp=True then
    begin
      if (Carlu='r') or (Carlu='R') then TVar.Type:=2;      (* reconnait un reel *)
      if (Carlu='i') or (Carlu='I') then TVar.Type:=3;      (* reconnait un entier *)
      FlagTyp:=false;
    end
    else
      if Carlu = '/' then FlagTyp:=true      (* détecte chgmt de type *)
      else
        if Carlu = ']' then      (* détecte fin de zone variable *)
        begin
          if Chaine='' then error:=2;      (* erreur si pas de nom *)
          TVar.lg:=TVar.lg-1;
          TVar.nvar:=copy(chaine,1,15);
          if (TVar.typ=2) and (TVar.lg<8) then Error:=4;      (* erreur si reel de long.<8 car *)
          ATVar[NbVar]:=TVar;
        end
      end
    end
  end
end

```

Fig. 3. - Listing du programme.

```

fvar:=false;
ftext:=true;
end
else
  Chaine:=chaine+CarLus;
end;
end; (* ----- endproc anal_var ----- *)

procedure anal_text; (* ----- reconnait et enregistre le texte lu ----- *)

procedure sau_v_chain; (* ----- reconnait ligne de texte ----- *)
begin
  TText(NbText):=Chaine;
  NbText:=NbText+1;
  Ion:=0;
  Chaine:='';
end; (* ----- endproc Sauv_Chain ----- *)

begin (* ----- copie le texte ----- *)
if TVar.Lg <> 0 then
begin
  chaine:=chaine+C;
  chaine:=chaine+copy(vide,1,TVar.lg);
  Ion:=Ion+TVar.lg+1;
  TVar.lg:=0;
  fvar:=false;
end;
chaine:=chaine+carlus;
Ion:=Ion+1;

if Ion=80 then sau_v_chain;
end; (* ----- endproc anal_text ----- *)

begin (* ----- prog principal anal_ecran ----- *)
Vide:=
curx:=1; cury:=1;
error:=1; fvar:=false; (* initialisation de l'analyse *)
ftext:=true; NVar:=0; (* du fichier texte en entrée *)
NbText:=1; TVar.lg:=0;
Ion:=0; Chaine:='';
repeat
  Tchaine:=vide;
  Tchaine:=TEnt(curx);
  carlus:=Tchain(curx);
  if fvar then (* passage en mode analyse variable *)
    anal_var;
  if carlus = '[' then (* detection de début de variable *)
    init_anal_var;
  if ftext then (* reconnaissance du texte *)
    anal_text;
  curx:=curx+1;
  if curx=81 then
    if fvar then
      error:=1 (* erreur si pas de ']' après '[' *)
    else
      begin
        cury:=cury+1;
        curx:=1;
      end;
    if cury=25 then error:=0; (* sortie si analyse ok *)
  until error=0;
end; (* ----- endproc anal_ecran ----- *)

procedure make_turbo; (* ----- génération du source turbo ----- *)
var BufLin,
    BufLi2 : string[120];
    Lid : string[80];

procedure make_var; (* ----- génération des variables ----- *)
begin
  writeln(Fo, '(* Généré par programme TURBO MAKER L.Dumont *)');
  writeln(Fo, '(* Définition du type employé par la procedure: ', NomEcr, ' *)');
  writeln(Fo, ' ');
  writeln(Fo, 'Type V', NomEcr, '=Record';
  BufLin:=
  BufLi2:=
  for i:=1 to NbVar do
    begin
      TVar:=ATVar[i];
      if TVar.Type=3 then
        BufLin:=BufLin+TVar.Nvar+', ' ;
      if TVar.Type=2 then
        BufLi2:=BufLi2+TVar.Nvar+', ' ;
    end;
  if BufLin<>
    then (* regroupe les variables de type 'integer' *)
    begin
      BufLin:=copy(BufLin,1,(length(BufLin)-1));
      BufLin:=BufLin+ ' : Integer';
      writeln(Fo,BufLin);
    end;
  if BufLi2<>
    then (* regroupe les variables de type 'Real' *)
    begin
      BufLi2:=copy(BufLi2,1,(length(BufLi2)-1));
      BufLi2:=BufLi2+ ' : Real';
      writeln(Fo,BufLi2);
    end;
  for i:=1 to NbVar do
    (* définition des types 'string' *)
    begin

```

```

TVar:=ATVar[i];
if TVar.Type=1 then
begin
  BufLin:=
  str(TVar.lg,Lid);
  BufLin:=BufLin+TVar.Nvar+ ' : String'+Lid+'';
  writeln(Fo,BufLin);
end;
end;
writeln(Fo, End);
writeln(Fo, ');
writeln(Fo, ');
writeln(Fo, 'Procedure ', NomEcr, '(Var T: V', NomEcr, ');');
writeln(Fo, ' (* Masque d'écran crée par générateur TURBO MAKER *)');
writeln(Fo, ');
writeln(Fo, ' Var Ion,X,Y,Car,L,NrVar,Error,1 : Integer; (* variables statiques du générateur *)');
writeln(Fo, ' Buf1,Buf2 : String[79];');
writeln(Fo, ' Flag : Boolean;');
writeln(Fo, ' Car : Char;');
writeln(Fo, ');
end; (* ----- endproc make_var ----- *)

procedure make_gestion; (* ----- genere la gestion des zones ----- *)
begin
  writeln(Fo, ' Procedure Ch_Zon;');
  writeln(Fo, ' (* gestion des zones en entrée sur écran *)');
  writeln(Fo, ' Begin');
  writeln(Fo, ' Case NrVar of');
  for i:=1 to NbVar do
    begin
      str(i,Lid);
      writeln(Fo, ' '+Lid+' : begin');
      TVar:=ATVar[i];
      str(TVar.lg,Lid);
      if TVar.Type=1 then
        begin
          if TVar.Type=2 then
            writeln(Fo, ' Str(T, '+TVar.nvar+' :3, Buf2);')
          else
            writeln(Fo, ' Str(T, '+TVar.nvar+', Buf2);');
            writeln(Fo, ' Val(Buf1,T, '+TVar.nvar+', Error);');
          end
        else
          begin
            writeln(Fo, ' Buf2:=T, '+TVar.nvar+';');
            writeln(Fo, ' T, '+TVar.nvar+':= copy(Buf1,1, '+Lid+' );');
          end;
        Str(TVar.px,Lid);
        writeln(Fo, ' X := '+Lid+';');
        str(TVar.py,Lid);
        writeln(Fo, ' Y := '+Lid+';');
        str(TVar.lg,Lid);
        writeln(Fo, ' L := '+Lid+';');
        writeln(Fo, ' end;');
      end;
      writeln(Fo, ' end;');
      writeln(Fo, ' end;');
      writeln(Fo, ');
    end; (* ----- endproc make_gestion ----- *)

procedure make_text; (* ----- fabrique des writeln avec le texte ----- *)
begin
  writeln(Fo, ' Begin');
  writeln(Fo, ' (* Ecriture du texte de l'écran prédéfini *)');
  writeln(Fo, ' Error:=0;');
  for i:=1 to NbVar do
    begin (* Initialisation des variables *)
      (* à la valeur blanc ou zéro, *)
      (* cette zone peut être enlevée *)
      if TVar.Type=1 then
        writeln(Fo, ' T, '+TVar.nvar+':='';') (* si vous préférez passer une *)
      else (* valeur initiale aux variables *)
        writeln(Fo, ' T, '+TVar.nvar+':=0;') (* lors de l'appel de procédure *)
      end;
      (* >> SPECIAL AU TURBO: *)
      writeln(Fo, ' ClrScr;') (* effacement d'écran *)
      writeln(Fo, ' LowVideo;') (* passage en sous intensité *)
      for i:=1 to 24 do
        begin
          Lid:='';
          Io:=79;
          repeat
            carlus:=copy(TText(i),Io,1);
            Io:=Io-1;
          until (carlus<chr(32)) or (Io=1); (* Elimination des blancs en fin *)
          Lid:=copy(TText(i),1,Io+1); (* de ligne de dessin d'écran *)
          if Lid=
            then
              BufLin:= ' writeln(' ;
            else
              BufLin:= ' writeln(''+Lid+'');';
              writeln(Fo,BufLin);
            end;
          writeln(Fo, ');
          (* gestion des zones écran *)');
          writeln(Fo, ');
          writeln(Fo, ' NrVar:=1; Buf1:=''; Zon:=0; Flag:=false;');
          writeln(Fo, ' Repeat');
          writeln(Fo, ' Ch_Zon;');

```

Listing du programme (suite).

```

writeln(Fo, ' if Eror=0 then begin ');
writeln(Fo, '   NrVar:=NrVar+Zon; ');
writeln(Fo, '   gotoxy(1,25);write('' ');');
writeln(Fo, ' end else begin ');
writeln(Fo, '   gotoxy(1,25);write(''SAISIE INVALIDE RECOMMENCEZ!!''); end;');
strlnbVar+1,Lid;
writeln(Fo, ' if NrVar='+Lid+' then NrVar:=1;');
writeln(Fo, ' if NrVar=0 then NrVar:='+Lid+';');
writeln(Fo, ' Ch_Zon; Buf1:=Buf2; Cur:=X; Zon:=0;');
writeln(Fo, ' gotoxy(X,Y);for i:=1 to L do write('' ');');
writeln(Fo, ' Repeat ');
writeln(Fo, '   NormVideo ; gotoxy(X,Y) ; write(Buf1); gotoxy(Cur,Y); '); (* SPECIFIQUES TURBO: *)
writeln(Fo, ' Repeat Until KeyPressed; Read(Kbd,Car); '); (* NormVideo et Lowvideo: *)
writeln(Fo, ' if (Car=#27) and (not(KeyPressed)) then Flag:=true; '); (* surintensité de l'écran *)
writeln(Fo, ' else if Car=#13 then Zon:=1; '); (* KeyPressed: attend qu'un *)
writeln(Fo, ' else if Car=#8 then begin '); (* caractère soit tapé au *)
writeln(Fo, '   Cur:=Cur-1; if X>Cur then Cur:=X; end; '); (* clavier, iu par KBD *)
writeln(Fo, ' else ');
writeln(Fo, '   if (Car>chr(31)) and (Car<chr(123)) then begin ');
writeln(Fo, '     Buf2:=Buf1 ; Buf1:=copy(Buf2,1,(Cur-X))+Car+copy(Buf2,(Cur-X+2),L); ');
writeln(Fo, '     Cur:=Cur+1; ');
writeln(Fo, '   end; ');
writeln(Fo, '   if (Cur-X)>L then Zon:=1;');
writeln(Fo, ' Until Flag or (Zon<0);');
writeln(Fo, ' Until Flag;');
writeln(Fo, ' Ch_Zon;');
writeln(Fo, ' ClrScr;');
writeln(Fo, ' End; ');
end; (* ----- endproc make_text ----- *)

begin (* ----- progrprincipal make_turbo ----- *)
  assign(Fo,NomFichier);
  NomEcr:=copy(NomFichier,pos(':',NomFichier)+1,length(NomFichier));
  NomEcr:=copy(NomEcr,1,pos('.',NomFichier)-1);
  rewrite(Fo);
  make_var;
  make_gestion;
  make_text;
  close(Fo);
end; (* ----- endproc make_turbo ----- *)

begin (* ----- programme principal GEN_ECRAN ----- *)
  ClrScr; (* spécial TURBO: effacement d'écran *)
  Repeat
    writeln('--- Entrez le nom du fichier texte : ');
    Readln(NomFichier);
    assign(Fi,NomFichier); (* >> I: directive de compilation *)
    (**I*) reset(Fi); (**I+*) (* permet de recuperer erreur *)
    ok:=(IOResult=0); (* si le fichier n'existe pas *)
    writeln;
    if not ok then
      writeln('Impossible de trouver le fichier "',Nomfichier,"");
      writeln;
    Until OK;
    for i:=1 to 25 do
      TEnt[i]:='';
    i:=1;
    Repeat
      readln(Fi,TEnt[i]);
      i:=i+1;
    until (i=25) or Eof(Fi);
    close(Fi);
    writeln;
    writeln(' Analyse de l'écran - Traitement en cours, patientez...');
    writeln;
    anal_ecran;
    if Eror>0 then
      begin
        writeln('--- ERREUR ',Eror,' --- LORS DE L'ANALYSE DES VARIABLES!');
        writeln;
        writeln('Numero de la variable concernée: ',NbVar);
        if Eror=1 then writeln(' -> Ne se termine pas sur une ligne');
        if Eror=2 then writeln(' -> Il faut donner un nom à la variable ');
        if Eror=3 then writeln(' -> Variable non fermée; il manque "1" ');
        if Eror=4 then writeln(' -> Variable réelle de longueur < 8 caractères ');
        writeln;
        WriteLn(' Reprenez votre fichier source, ('',NomFichier,'') et modifiez le. ');
      end
    else
      begin
        writeln;
        writeln(' ----- Entrez le nom du fichier de sortie (va contenir le source Pascal): ');
        readln(NomFichier);
        if (pos('.pas',NomFichier)=0) and (pos('.PAS',NomFichier)=0) then NomFichier:=NomFichier+'.PAS';
        writeln;
        writeln(' CREATION DU SOURCE PASCAL ... en cours ');
        writeln;
        make_turbo;
        writeln(' Le fichier contenant la procedure pascal est crée. ');
        writeln(' il s'appelle : ',NomFichier);
        writeln(' la procedure s'appelle : ',NomEcr);
        writeln(' la variable totale : V',NomEcr);
        writeln;
        writeln(' Fin de Traitement MAK_TURBO ');
        end;
end. (* ----- fin du programme principal GEN_ECRAN ----- *)

```