

```

package Lecture_Sure is
  function Lire_Code (invite : string; Nb_Max: Integer := 3) return Integer;
  -- cette fonction lit sur le support d'entrée standard un nombre avec nb_max tentative;
  -- si échec au bout de ces tentative, le retour est le plus petit entier possible (négatif)

end Lecture_Sure ;

```

===== attention, ceci ci dessous est dans un autre fichier compile séparément

```

with Ada.Text_IO;
use Ada.Text_Io;
with ada.integer_text_IO; use ada.Integer_Text_IO;

```

```

package body Lecture_Sure is
  function Lire_Code (Invite : String; Nb_Max: Integer := 3) return Integer is

    Valeur : Integer;
    Texte : String (1..20);
    lg_texte : natural;
  begin
    Put_Line (Invite);
    for I in 1..Nb_Max loop
      -- tant que le nombre n'est pas lu correctement, on réessaie
      begin
        -- alternative : Get_line (texte, lg_texte); get (texte(1..lg_texte), valeur, lg_texte);
        Get (Valeur) ;
        return valeur; -- ici la lecture c'est bien passée : fin
      exception
        when Data_Error =>
          Skip_Line ; -- on purge la ligne erronée
          Put_Line ("Erreur, attention, entrer une valeur numérique");
          Put_Line (Invite);
        end;
      end loop;
      -- A ce stade, on abandonne
      Put_Line ("désolé, veuillez consulter votre conseiller");
      return Integer'First;
    end Lire_Code;

  end Lecture_Sure;

```

-- programme de test :

```

with Ada.Text_Io; use Ada.Text_Io;
with Ada.Integer_Text_Io; use Ada.Integer_Text_Io;
with lecture_sure; use lecture_sure;

procedure Test_Lecture is
  X : Integer;

begin
  X := Lire_Code("merci de donner votre numéro secret");
  Put_Line ("le code lu est "& Integer'Image (X));
end Test_Lecture;

```