

```

                                Unit1
{=====
=====      STEPHANE BRESSANI      =====
=====
=====      INFO P2A      =====
=====
=====      EXERCICE 3.19      =====
=====      FIFO      =====
=====
=====      VERSION 1.0      =====
=====      12 mars 2003      =====
=====
=====      Exercice qui crée un fifo      =====
=====
}

unit Unit1;

interface

uses
  windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Grids, StdCtrls;

type
  TFIFO = class(TForm)
    StringGrid_FIFO: TStringGrid;
    Button_In: TButton;
    Button_OUT: TButton;
    Edit_In: TEdit;
    Edit_Out: TEdit;
    Edit_FIFO: TEdit;
    procedure Button_InClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button_OUTClick(Sender: TObject);
  private
    procedure Affiche();
    { Déclarations privées }
  public
    { Déclarations publiques }
  end;

const
  BUFSIZE = 15 ;    // on définit la constante de la taille du buffer :)
var
  FIFO: TFIFO;
  BufferFifo : array [0..BUFSIZE] of char;
  // on fait un tableau avec comme taille la taille de la constante BUFSIZE
  G_head : integer = 0; //on déclare la variable globale de
                        //la tete et de la queue
  g_queue : integer = 0;

implementation

{$R *.DFM}

// porcedure ki sert a afficher dans l'edit_Fifo.text :)
procedure TFIFO.Affiche();
var TempStr : string;
    CopieQueue : integer; //On déclare en local deux petite variable temporaire
begin
  TempStr := ''; // on met la variable temporaire ki est en string a vide
  CopieQueue := G_Queue; // on fias une copie de la que dans la 2ème
                        //variable temporaire qui est en integer
  while CopieQueue <> (G_Head) do
    begin

```



```

Unit1
// tant ke la queu (variable Copie_queue) <> a la tete
tempStr:= tempStr + BufferFifo[CopieQueue]; // on rajoute a tempstr se
// qui se trouve dans le tbl
// du buffer :)
inc(CopieQueue); // et en incérmente
// copie queue
if CopieQueue > BUFSIZE then // Si Copiequeue > ke la taille du buffer
  CopieQueue := 0; // CopieQueue
end;
Edit_FIFO.Text := tempStr; // On Affiche dans l'edit la valeur de tempstr
end;
// procedure qui se lance quand en appuie sur le bouton IN
procedure TFIFO.Button_InClick(Sender: TObject);
var MonChar : char;
    temp: string;
begin
  BUTTON_Out.Enabled := True; // Le bouton Out est sur TRUE
  MonChar := chr(random(26) + integer('A')); // On ajoute un charater aléatoire
// dans la variable monchar
  BufferFifo[G_Head] := MonChar; // on rajoute dans le tableau a la
// place de la tete le cahracètre
// aléatoire crée en dessus :)
  Edit_IN.Text := MonChar; // Au passage en le rentre
// dans l'edit_in :)
  StringGrid_FIFO.Cells[G_Head,0] := MonChar ;
// et dans le stringgrid par la
// meme ocasion
  temp := ''; // on vide la variable local temp
  inc(G_Head); // et on incrémente la tête
  if G_Head > BUFSIZE then // si la tête > que la taille
// du buffer
// on met la tete a 0
  G_Head := 0; // si la tete égale la que
  if G_Head = (G_queue) then
  begin
    showmessage('Fifo Plein'); // on affiche que le fifo ets plein
    Button_In.Enabled := False; // et on verouille bouton In
// (on le met sur False)
  end;
  Affiche(); // on lance la procédure affiche
end;

procedure TFIFO.FormCreate(Sender: TObject);
begin
  randomize; // a la cération de la form on randomize ^^
end;

// procedure qui se lance quand en appuie sur le bouton OUT
procedure TFIFO.Button_OUTClick(Sender: TObject);
begin
  BUTTON_In.Enabled := True; // on enclance le bouton in (on le passe a true)
  Edit_OUT.Text := StringGrid_FIFO.Cells[G_queue+1,0]; // l'edit_out vaudra la
// valeur ki est a la place dans le string grid (tete + 1)
  StringGrid_FIFO.Cells[G_queue,0] := ''; // on vide la position qui vaut
// la queue du string grid
  inc(G_queue); // on incrémente la queue
  if G_queue > BUFSIZE then // si la queue > que le buffer
// elle passe alros a 0
  G_queue := 0; // si elle egale la queu alors
  if g_queue = g_Head then // on passe le bouton_Out a false
  begin
    BUTTON_OUT.Enabled := False; // et on affiche que le fifo
// est vide
    showmessage('fifo vide');
  end;
  Affiche(); // on apelle la procedure affiche
end;
end.

```