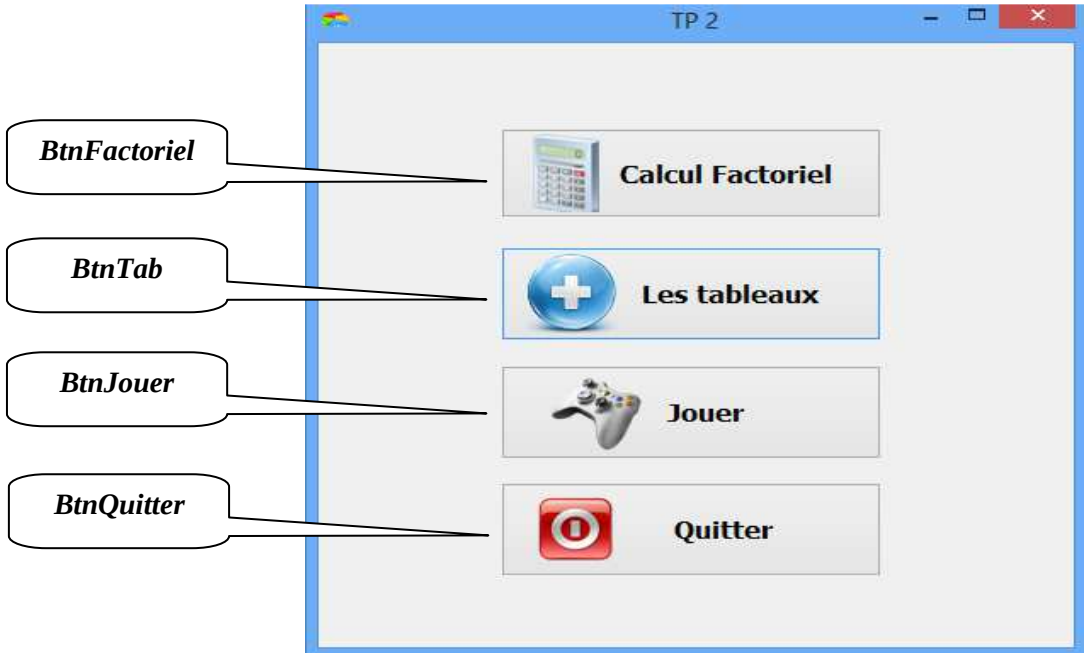
TP 2 : STRUCTURES DE CONTROLE

OBJECTIFS :

Manipulation des tableaux et structures de contrôle avec C# et découverte d'autres contrôles et évènements.

I-Interface d'accueil:

Placer dans l'interface ***frmAccueil*** suivante:



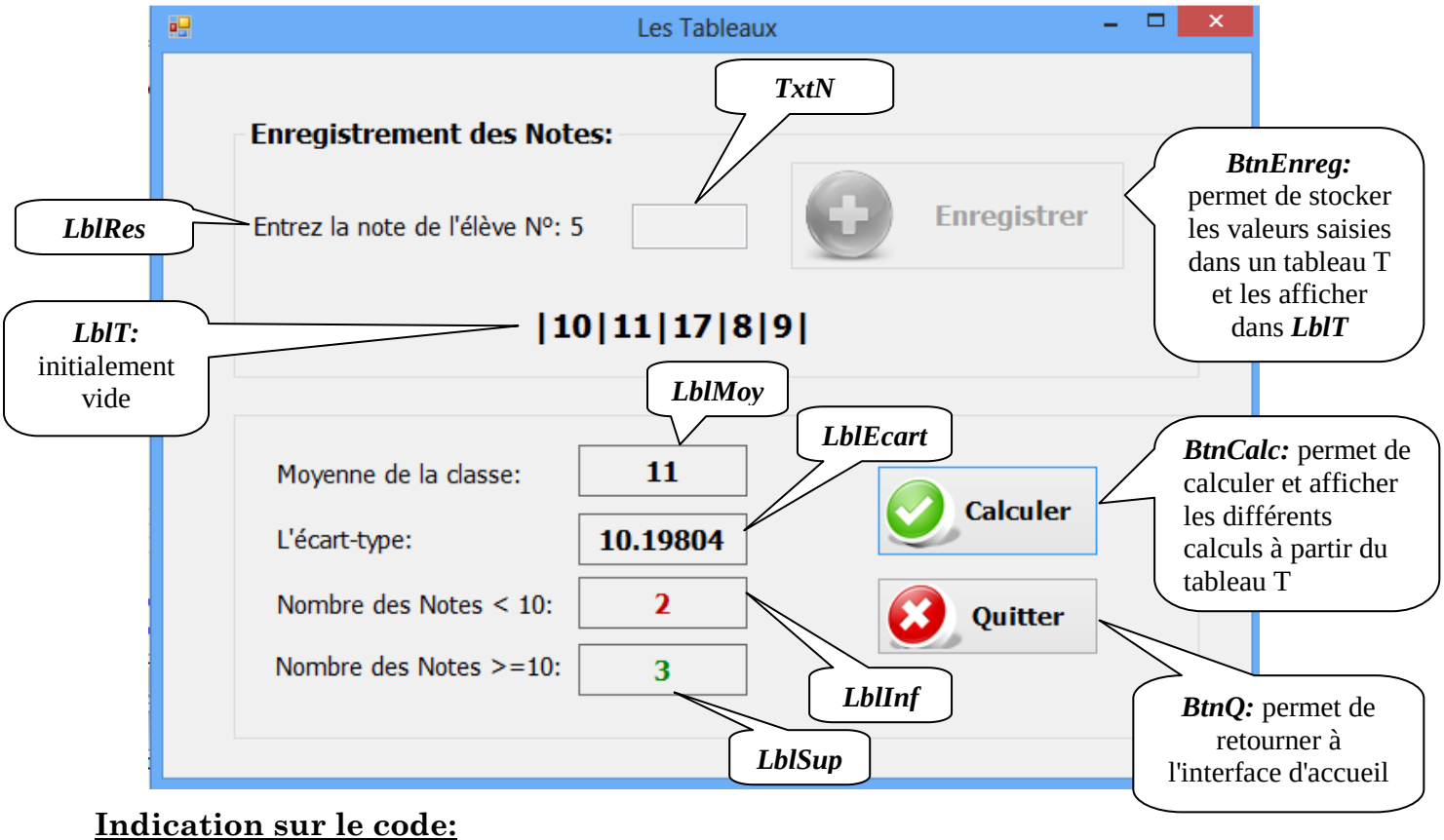
II-Factoriel:

Placer dans l'interface ***frmFactoriel*** suivante:

III-Les tableaux:

Développez une application permettant la saisie et le stockage des notes d'un ensemble de 5 élèves. A partir de ces notes, on calcule la moyenne de la classe, son écart-type, le nombre des notes < 10 et le nombre des notes >= 10.

Le calcul de la moyenne: $\overline{x} = \sum_{i=1}^{N} x_i / N$. Le calcul de l'écart-type est : $\sigma_x = \sqrt{\sum_{i=1}^{N} \left(x_i - \overline{x}\right)^2}$



Indication sur le code:

Déclaration globale:

```
const int n = 5;
float[] T = new float[n];
int i = 0;
```

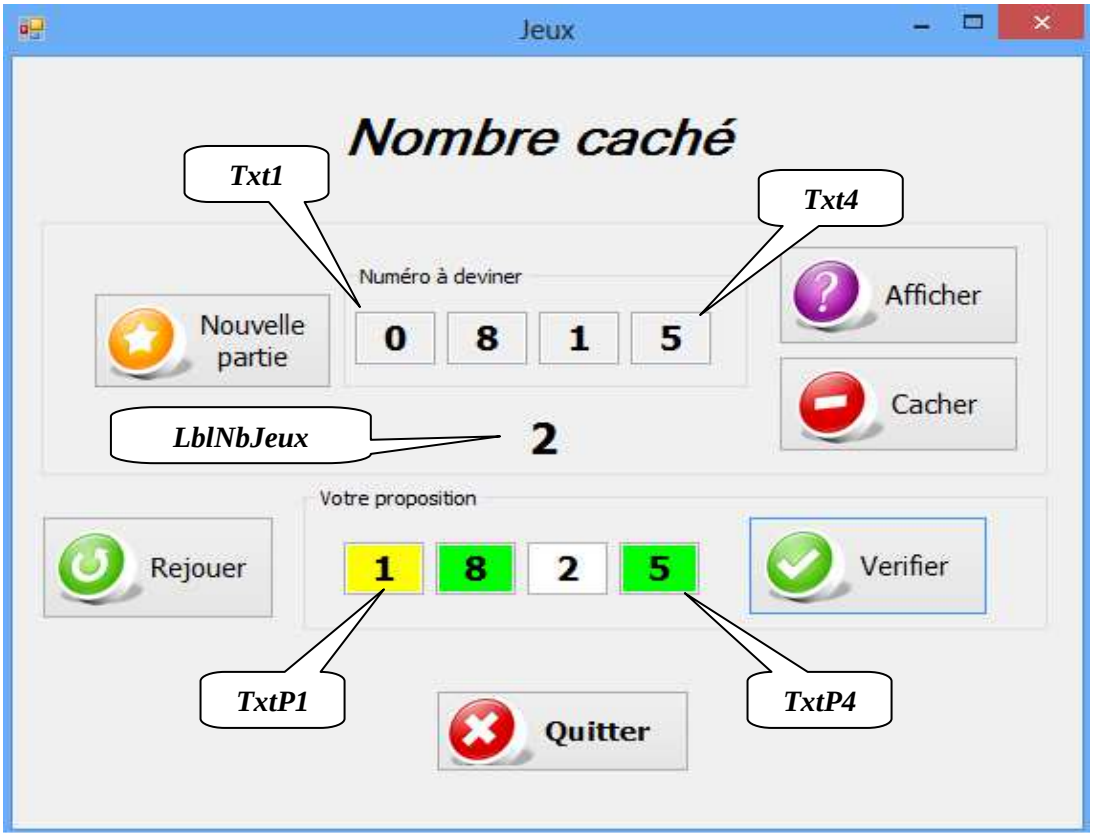
Bouton enrgistrer

```
float val;
if (float.TryParse(.....) && val >= 0 && ....)
{
   lblRes.Text = "Entrez la note de l'élève N°: " + (i + 2).ToString();
   lblT.Text = lblT.Text + txtn.Text + "|";
   T[i]=val;
   i++;
   TxtN.Text = "";
   TxtN.Focus();
   if (i == n)
   {
        BtnCalc.Enabled = true;
        BtnEnreg.Enabled = false;
        txtn.Enabled = false;
        BtnCalc.Focus();
   }
} else .....
```

IV-Jeux: nombre caché:

L'objectif et de générer aléatoirement un numéro composé par 4 chiffres et de demander à l'utilisateur de le trouver dans au maximum 3 propositions.

Le bouton vérifier permet de comparer la proposition de l'utilisateur avec le numéro caché et de colorer en vert les chiffres bien placé et en jaune les chiffres qui existent mais mal placé.



Indication sur le code:

Déclaration globale:

```
List<TextBox> LTxtC = new List<TextBox>();
List<TextBox> LTxtP = new List<TextBox>();
int nbJeux=0;  bool animation = false;
```

Chargement de l'interface:

```
int n1,n2,n3,n4;
Random random = new Random();
n1 = random.Next(0, 10);
txt1.Text = n1.ToString();
do
     n2 = random.Next(0, 10);
while (n2 == n1);
txt2.Text = n2.ToString();
do
     n3 = random.Next(0, 10);
while (n3 == n2||n3==n1);
txt3.Text = n3.ToString();
......
LTxtC.Add(Txt1);
.....
LTxtP.Add(TxtP1);
.....
```

Le bouton Cacher:

```
for (int i = 0; i < 4; i++)
       LTxtC[i].ForeColor = Color.White;
```

Le bouton Rejouer:

```
for (int i = 0; i < 4; i++)
  {
       LTxtP[i].BackColor = Color.White;
       LTxtP[i].Text = "";
  }
LTxtP[0].Focus();
```

Le bouton Verifier:

```
int i,j,correct = 0;
for(i=0;i<4;i++)
{
    if (LTxtP[i].Text == LTxtC[i].Text)
    {
         LTxtP[i].BackColor = Color.Lime;
         correct++;
    }
    else
    for (j = 0; j < 4; j++)
    {
         if (LTxtP[i].Text == LTxtC[j].Text)
            {
                    LTxtP[i].BackColor = Color.Yellow;
                    break;
            }
}   }
if (correct == 4)
{
      timer1.Start();
      for (i = 0; i < 4; i++)
             LTxtC[i].ForeColor = Color.Black;
}
else
{    nbJeux++;
     LblNbJeux.Text = (nbJeux+1).ToString();
     if (nbJeux == 3)
     {
          GrB1.Enabled = false;
          for (i = 0; i < 4; i++)
              LTxtC[i].ForeColor = Color.Black;
              MessageBox.Show("Vous avez perdu", "Game Over",
                        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
     }
}
```

Le timer:

```
if (animation == true)
{
   for (int i = 0; i < 4; i++)
       LTxtC[i].BackColor = Color.Yellow;
   animation = false;
}
else
{
   for (int i = 0; i < 4; i++)
       LTxtC[i].BackColor = Color.Red;
   animation = true;
}
```