

Untitled-1

```

1 namespace AP8_PROGRAMME_VALLEE_ESTEBAN
2 {
3     public partial class Form1 : Form
4     {
5         public Form1()
6         {
7             InitializeComponent();
8         }
9
10        private void textBox1_TextChanged(object sender, EventArgs e)
11        {
12
13        }
14
15        private void BTN_TRADBIN_Click(object sender, EventArgs e)
16        {
17            {
18                string adresseip = TB_ENTADR.Text; //adresse ip est prise de la tb, renseignée
par l'utilisateur
19                Global.IPV4 = TB_ENTADR.Text;
20                Global.CIDR = int.Parse(TB_CIDR.Text);
21                String[] IP = adresseip.Split("."); //On split l'adresse selon les points
22                if (IP.Length == 4 && IP[0] != "" && IP[1] != "" && IP[2] != "" && IP[3] != "")
23                {
24                    Int32 p1 = int.Parse(IP[0]); //l'adresse splitée en 4 va prendre 4 valeurs
int
25                    Global.IPnb1 = p1; //On enregistre la partie 1 pour la question sur la
classe de l'IP
26                    Int32 p2 = int.Parse(IP[1]);
27                    Int32 p3 = int.Parse(IP[2]);
28                    Int32 p4 = int.Parse(IP[3]);
29
30                    if (p1 >= 0 && p1 <= 255 // on vérifie que chaque partie de l'ip est valide
31                        && p2 >= 0 && p2 <= 255
32                        && p3 >= 0 && p3 <= 255
33                        && p4 >= 0 && p4 <= 255)
34                    {
35                        if (Global.CIDR > 0 && Global.CIDR < 32) //on vérifie le CIDR et sa
validité
36                        {
37                            MessageBox.Show("Votre adresse IP et votre CIDR ont été
enregistrés !");
38                            Int16 numBase = 2; //base binaire
39                            string number1 = Convert.ToString(p1, numBase).PadLeft(8, '0');
//conversion base binaire
40                            string number2 = Convert.ToString(p2, numBase).PadLeft(8, '0');
41                            string number3 = Convert.ToString(p3, numBase).PadLeft(8, '0');
42                            string number4 = Convert.ToString(p4, numBase).PadLeft(8, '0');
43
44                            MessageBox.Show("Conversion " + p1 + "." + p2 + "." + p3 + "." + p4
+ "." + " en base " + numBase + " = " + number1 + " " + number2 + " " + number3 + " " + number4
+ " ");

```

```
45 Global.IPV4binaire = number1 + "." + number2 + "." + number3 + "."
+ number4;
46
47 int cidr = Global.CIDR;
48 int[] masque = { 0, 0, 0, 0 };
49 // valeurs possibles pour octet masque
50 int[] masqueoctet = { 0, 128, 192, 224, 240, 248, 252, 254, 255 };
51
52 for (int i = 0; i < 4; i++)
53 {
54     int bitsRestants = cidr - (i * 8);
55
56     if (bitsRestants >= 8)
57     {
58         masque[i] = 255; // octet plein
59     }
60     else if (bitsRestants > 0)
61     {
62         masque[i] = masqueoctet[bitsRestants]; // masque partiel
63     }
64     else
65     {
66         masque[i] = 0; // que des 0
67     }
68 }
69
70 int[] ip = { p1, p2, p3, p4 };
71 int[] reseau = { 0, 0, 0, 0 };
72
73 for (int i = 0; i < 4; i++)
74 {
75     if (masque[i] == 255)
76     {
77         reseau[i] = ip[i];
78     }
79     else if (masque[i] == 0)
80     {
81         reseau[i] = 0;
82     }
83     else
84     {
85         reseau[i] = ip[i] - (ip[i] % (256 - masque[i]));
86     }
87 }
88
89 string adresseReseau =
90     reseau[0].ToString() + "." +
91     reseau[1].ToString() + "." +
92     reseau[2].ToString() + "." +
93     reseau[3].ToString();
94
95 MessageBox.Show("Adresse réseau principale : " + adresseReseau);
96
97 resultatIPB page2 = new resultatIPB();
98 this.Hide();
```

```
98         page2.Show();
99
100     }
101
102     else
103     {
104         MessageBox.Show("Votre CIDR est incorrect, il doit être compris
entre 1 et 32 !");
105     }
106 }
107
108
109
110
111     else
112     {
113         MessageBox.Show("Votre adresse IP est incorrecte !");
114     }
115 }
116
117
118
119     else
120     {
121         MessageBox.Show("Votre adresse IP est incorrecte !");
122     }
123 }
124 }
125
126
127
128
129 private void TB_ENTADR_KeyPress(object sender, KeyPressEventArgs e)
130 {
131
132 }
133
134 private void TB_ENTADR_TextChanged(object sender, EventArgs e)
135 {
136     if (TB_CIDR.Text != "" && TB_ENTADR.Text != "")
137     {
138         BTN_SUIV.Enabled = true;
139         BTN_TRADBIN.Enabled = true;
140     }
141
142 }
143 private void Form1_Load(object sender, EventArgs e)
144 {
145     if (TB_CIDR.Text == "" && TB_ENTADR.Text == "")
146     {
147         BTN_SUIV.Enabled = false;
148         BTN_TRADBIN.Enabled = false;
149     }
150 }
```

```
151     }
152
153     private void TB_CIDR_TextChanged(object sender, EventArgs e)
154     {
155         if (TB_CIDR.Text != "" && TB_ENTADR.Text != "")
156         {
157             BTN_SUIV.Enabled = true;
158             BTN_TRADBIN.Enabled = true;
159         }
160     }
161
162     private void BTN_QuitAccueil_Click(object sender, EventArgs e)
163     {
164         this.Close();
165     }
166
167     private void BTN_SUIV_Click(object sender, EventArgs e)
168     {
169         if (Global.IPV4binaire == null)
170         {
171             BTN_SUIV.Enabled = false;
172             MessageBox.Show("Aucune IP ou masque n'est renseigné.");
173         }
174
175         else
176         {
177             BTN_SUIV.Enabled = true; //le bouton suivant est enable seulement si
l'utilisateur a déjà rentré par le passé des infos
178             resultatIPB page2 = new resultatIPB();
179             this.Hide();
180             page2.Show();
181
182         }
183     }
184 }
185 }
186
187
188
189
190
191
192
193
194
195
```