AP n°7 : Calcul d'adresses IP et masque de sous-réseau

Description	
Descriptif de l'AP	A partir de la saisie d'une adresse IP et du CIDR : - Vérifier la conformité des données saisies - Convertir l'IP en binaire - Calculer le masque de sous-réseau et l'afficher en valeurs décimale et binaire - Calculer le nombre d'hôtes
Durée estimée	8 heures pour la compréhension du sujet et la programmation
Savoir-faire	Ecrire un programme en langage C# à partir d'un besoin utilisateur Adressage IP
Compétences	Bloc 1 - Support et mise à disposition des services informatiques B1.2 – Bases de la programmation B1.2 – Bases du réseau
Contexte	Contexte n°3 – Commune de MARUT
Ressources	Aides à la programmation C# : - Annexe 1 : méthode ToString pour la conversion de données - Annexe 2 : méthode ToByte pour la conversion de données - Annexe 3 : tableau de valeurs - Annexe 4 : méthode Split pour extraire des valeurs d'une chaine

Dans le cadre de l'Audit demandé par Monsieur Brillat, directeur de la structure MSAP de la commune de Marut, le prestataire HDesk'63 a notamment pour mission de réorganiser l'adressage IP du réseau et créer des sous-réseaux. Ces sous-réseaux permettront d'isoler la partie ouverte au public de la partie réservée au personnel administratif. Il faut aussi isoler la salle multimédia et les salles de réunion.

Dans cette optique, le prestataire vous demande de développer un outil logiciel qui aura plusieurs objectifs :

1. Saisie d'une adresse IP sous la forme de 4 valeurs séparées par un point (1 seul textBox)
2. Saisie du CIDR (Classless Inter-Domain Routing)
3. Ces informations seront vérifiées : les valeurs doivent correspondre à des valeurs autorisées pour une adresse IP et un CIDR

Si les informations sont correctes, il faudra alors procéder aux traitements suivants :

4. Convertir et afficher l'adresse IP en binaire
5. Calculer le masque de sous-réseau en valeur décimale et en valeur binaire et l'afficher
6. Déterminer et afficher l'adresse réseau principale
7. Calculer et afficher le nombre d'hôtes maximum du sous-réseau
8. Déterminer la classe de l'adresse IP
9. Calculer le nombre de sous-réseaux
10. Calculer le nouveau CIDR à partir d'un nombre de sous réseau souhaité

Etape 1 :

- Création d'une première interface permettant la saisie de l'adresse IP et du CIDR
- Un bouton permettra de calculer et d'afficher l'adresse IP binaire
- Un bouton accessible uniquement après calcul de l'adresse binaire permettra d'accéder à la deuxième interface

Etape 2 :

- Création d'une interface affichant les traitements 5,6,7 et 8
- Afficher avant les résultats des traitements 5,6,7,8 et 9 l'adresse IP et le CIDR saisis dans la première interface

Etape optionnelle (point 10) :

- Saisie du nombre de sous réseau souhaité
- Un bouton permettant de calculer le nouveau CIDR à partir du nombre de sous réseaux souhaité

L'utilisateur doit pouvoir quitter l'application sur les 2 interfaces et effectuer une nouvelle demande sur la première interface.

Outil logiciel :
Vous utiliserez l'IDE Visual Studio avec le langage C#

Equipe de développement :
Le travail sera réalisé en individuel

Contraintes de développement :
- Le logiciel imposera à l'utilisateur la vérification des données saisies à l'étape 1. La seconde étape des traitements (points 4 à 8) ne sera accessible que si la 1ère étape s'est terminée avec succès.
- Les variables seront déclarées de façon réfléchie et adaptée à leur contenu. Par exemple, pour une valeur numérique de 0 à 64, un type Int32 ou Int64 est surdimensionné.

Document à fournir :
Un dossier compressé complet contenant :
- L'application C# complète

Remarque :
Les aides proposées en annexes sont facultatives. Des solutions n'utilisant pas les méthodes présentées sont possibles.
Vous veillerez à commenter votre code pour expliquer votre démarche.

Annexe 1 : méthode ToString pour la conversion de données

Source : https://docs.microsoft.com/fr-fr/dotnet/api/system.convert.tostring?view=netcore-3.1#System_Convert_ToString_System_Byte_System_Int32_

ToString(Byte, Int32)

Convertit la valeur d'un entier non signé 8 bits en sa représentation sous forme de chaîne équivalente dans une base spécifiée.

C# Copier

```
public static string ToString (byte value, int toBase);
```

Parameters

value Byte

Entier non signé 8 bits à convertir.

toBase Int32

Base de la valeur de retour, qui doit être 2, 8, 10 ou 16.

Returns

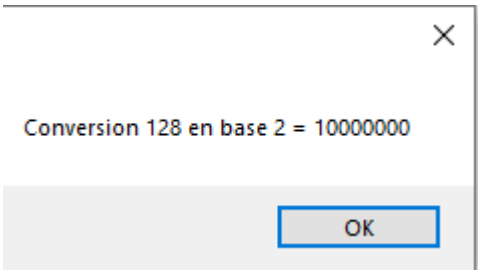
String

Représentation sous forme de chaîne de `value` en base `toBase`.

Exemple :

```
Int16 value = 128;
Int16 numBase = 2;
string number = Convert.ToString(value, numBase);

MessageBox.Show ("Conversion " + value + " en base " + numBase + " = " + number);
```



Annexe 2 : méthode ToByte pour la conversion de données

Source : https://docs.microsoft.com/fr-fr/dotnet/api/system.convert.tobyte?view=netcore-3.1

ToByte(String, Int32)

Convertit la représentation sous forme de chaîne d'un nombre dans une base spécifiée en entier non signé 8 bits équivalent.

C# Copier

```
public static byte ToByte (string value, int fromBase);
```

Parameters

value String

Chaîne contenant le nombre à convertir.

fromBase Int32

Base du nombre figurant dans `value`, qui doit correspondre à 2, 8, 10 ou 16.

Returns

Byte

Entier non signé 8 bits équivalent au nombre dans `value`, ou 0 (zéro) si `value` est `null`.

Exemple :

```
Int16 numBase = 2;
string value = "10000000";
byte number;

number = Convert.ToByte(value, numBase);
MessageBox.Show ("Conversion " + value + " en base " + numBase + " = " + number);
```

Annexe 3 : tableau de valeurs

Source : https://docs.microsoft.com/fr-fr/dotnet/csharp/programming-guide/arrays/

Tableaux (guide de programmation C#)

20/07/2015 • 2 minutes de lecture •

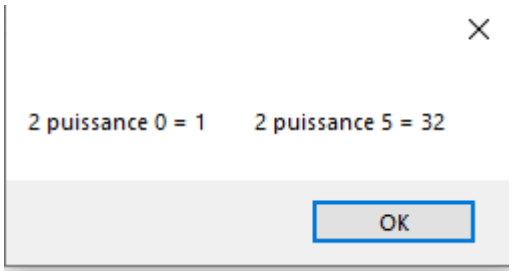
Vous pouvez stocker plusieurs variables du même type dans une structure de données de type tableau. Vous déclarez un tableau en spécifiant le type de ses éléments. Si vous souhaitez que le tableau stocke des éléments de n'importe quel type, vous pouvez spécifier `object` en tant que type. Dans le système de type unifié de C#, tous les types (les types référence et valeur, prédéfinis ou définis par l'utilisateur) héritent directement ou indirectement du type Object.

C# Copier

```
type[] arrayName;
```

Exemple :

```
Int16[] array1 = new Int16[] { 1, 2, 4, 8, 16, 32, 64, 128};
MessageBox.Show("2 puissance 0 = " + array1[0] + "        2 puissance 5 = " + array1[5]);
```



Annexe 4 : méthode Split pour extraire des valeurs d'une chaine de caractères

Source : https://docs.microsoft.com/fr-fr/dotnet/api/system.string.split?view=netcore-3.1

String.Split Method

Namespace: System
Assembly: System.Runtime.dll

Retourne un tableau de chaînes qui contient les sous-chaînes de cette instance, séparées par les éléments d'une chaîne ou d'un tableau de caractères Unicode spécifiés.

Exemple :

```
String numeroTelephone = "01-52-89-45-74";
String[] tel = numeroTelephone.Split('-');
MessageBox.Show("Voici le n° de téléphone : " + tel[0] + tel[1] + tel[2] + tel[3] + tel[4] + ".\n" + tel.Length + " parties composent ce numéro de téléphone");
```