A quoi faire attention au TPI

Mettre en place **une vérification des caractères**, permettre uniquement des chiffres (sauf le « . » pour le binaire). De plus il faut faire attention aux **conversions** comme ceux des boites de texte où l’utilisateur va entrer ce qu’il veut.

S’assurer de ce qui est attendu concernant le projet, les **fonctionnalités nécessaires** et ce qui doit être fait en plus.

Faire en sorte d’avoir des **options/fenêtre/onglets** pour séparer les fonctionnalités (additionner, soustraire, convertir, etc…). L’utilisation du programme doit être **simple** et **intuitif**, avec une aide pouvant guider légèrement l’utilisateur si besoin.

Faire sa propre **table de convertisseur**, attention à bien gérer le « . ».

Trouver un moyen de **mettre à jour** très régulièrement le **journal de travail** et le **rapport** et l’envoyer au chef /expert de projet plusieurs fois par semaine.

Faire un document complet permettant de **partager et transférer** le programme au complet, il faut pouvoir prendre le programme **d’un cd** et le **mettre** sur un ordinateur, un document permettant d’utiliser le programme au complet doit permettre une utilisation **simple** et **rapide**.

Viser **35 pages** de rapport minimum si possible, cela n’est pas obligatoire mais peut être utile à s’assurer qu’il soit suffisamment complet.

Faire attention aux **normes ETML** concernant le codage, le nom des dossiers et fichiers.

S’assurer de pouvoir correctement utiliser un Git afin de faire du **versioning**, une version chaque jour de l’application, le code, etc…, cependant il ne faut **pas prendre les librairies**, il faut **tester** afin de savoir si le code fonctionne **sans** **les** **librairies**.

Petit dictionnaire pour le TPI

**Récupérer un caractère spécifique :**

Permet de retrouver un caractère suivant sa position dans une chaîne de caractère.

**value**.Substring(\***position à partir du quel afficher les caractères**\*, \***nombre de caractère à afficher**\*);

<https://msdn.microsoft.com/fr-fr/library/aka44szs(v=vs.110).aspx>

**Compter le nombre de caractère :**

Permet de compter le nombre de caractère d’une chaîne de caractère.

**value**.Length;

<https://msdn.microsoft.com/fr-fr/library/aka44szs(v=vs.110).aspx>

**Vérification avec regex :**

Permet de s’assurer qu’il n’y ait que des 0 ou des 1 ou autre suivant ce que l’on définit.

Regex **Value** = new Regex("**^[0-1]+$**");

if (**rgx**.IsMatch(**valueBox**) && **rgx**.IsMatch(**valueBox2**))

<https://stackoverflow.com/questions/273141/regex-for-numbers-only>