

2018

Convertisseur de bases pour des nombres entiers et réels, codés sur 32 bits

Carbonara Christian – CIN4B

ETML

07/06/2018

Table des matières

1. Spécifications	3
1.1. Titre	3
1.2. Description	3
1.3. Matériel et logiciels à disposition	4
1.4. Prérequis	4
1.5. Cahier des charges	4
1.5.1. Objectifs et portée du projet (objectifs FFOR)	4
1.5.2. Caractéristiques des utilisateurs et impacts	4
1.5.3. Fonctionnalités requises (du point de vue de l'utilisateur)	4
1.5.4. Contraintes	5
1.5.5. Travail à réaliser par l'apprenti	5
1.5.6. Si le temps le permet	5
1.6. Les points suivants seront évalués	5
2. Planification initiale	6
3. Analyse	6
3.1. Opportunités	6
3.2. Document d'analyse et conception	6
3.2.1. Les conversions minimal/maximal avec 32 bits signés	6
3.2.2. Maquette graphique de manière générale	6
3.2.3. Maquette graphique pour les conversions	7
3.2.4. Maquette graphique pour les soustractions	9
3.2.5. Maquette graphique pour les additions	10
3.2.6. Conception du programme	11
3.3. Conception des tests	12
3.4. Planification détaillée	13
4. Réalisation	17
4.1. Fonctionnalités	17
4.1.1. Indexe des abréviations pour les noms d'objet	17
4.1.2. Lancement du programme	17
4.1.3. Comment entrer une valeur à convertir	18
4.1.4. Vérifications du nombre entré par l'utilisateur	18
4.1.5. Conversion de décimal à binaire avec nombre entier signé entier	20
4.1.6. Conversion de décimal à binaire avec nombre à virgule	22
4.1.7. Conversion de décimal à octal avec nombre entier	22
4.1.8. Conversion de binaire à décimal avec nombre entier	22

4.1.9.	Conversion de binaire à octal avec nombre entier.....	22
4.2.	Dossier de réalisation	23
4.2.1.	Logiciels nécessaires.....	23
5.	Tests.....	23
5.1.	Dossier des tests	23
6.	Conclusion.....	25
6.1.	Bilan des fonctionnalités demandées.....	25
6.1.1.	Conversion de décimal signé à binaire	25
6.1.2.	Conversion de décimal à virgule en binaire	27
6.1.3.	Conversion de décimal en octal non signé	27
6.1.4.	Décimal à virgule en octal.....	27
6.1.5.	Binaire en décimal non signé.....	27
6.1.6.	Binaire en octal non signé	27
6.1.7.	Binaire signé dans les autres bases	28
6.1.8.	Additions et soustractions (entiers, positifs et/ou négatifs)	28
6.2.	Bilan de la planification.....	28
6.3.	Bilan personnel.....	28
7.	Divers	29
7.1.	Journal de travail	29
7.2.	Webographie.....	29
8.	Annexes	30

1. SPÉCIFICATIONS

1.1. TITRE

Convertisseur de bases (binaire, octal, décimal et hexadécimal) pour des nombres entiers et réels (à virgules fixes), codés sur 32 bits

1.2. DESCRIPTION

Il s'agit d'implémenter une application, en C#, qui va permettre aux utilisateurs (des élèves et des enseignants), de saisir une base numérique (bases binaires, octales, décimales, hexadécimales, nombres à virgules fixes), un nombre et le programme affiche le même nombre dans les autres bases. De plus, le programme doit permettre d'accomplir des opérations élémentaires d'additions et soustractions sur des nombres entiers positifs et négatifs.

Le dispositif indiquera les détails des opérations, et se fera jusqu'à 32 bits signés pour les nombres binaires. En aucun cas l'utilisation de fonctions prédéfinies du langage ne sera acceptée. Les élèves doivent parfois résoudre des exercices, ou des situations, qui nécessitent des conversions de nombres d'une base à une autre. Ou encore, faire des additions ou des soustractions.

Par exemple, comment convertir un nombre réel à virgule fixe de décimal en hexadécimal (ou $134.45^{10} \rightarrow ?.^{16}$).

Actuellement, à l'ETML, en informatique, un module nommé ELEOC-NUM pour électronique numérique, traite ce sujet. Les supports présentent de la théorie et des exercices. Toutefois, il conviendrait de pouvoir laisser les élèves résoudre des mêmes exercices, mais avec des données différentes, et de pouvoir vérifier leurs réponses de manière automatique.

L'idée est d'avoir une application qui permette aux élèves de s'entraîner pour les fonctionnalités suivantes, avec des mots de 32 bits :

- Conversions de nombres décimaux, binaires, octaux et hexadécimaux (entiers et/ou réels à virgules fixes) en nombres dans les 3 autres bases que la base courante.
- Additions et soustractions de 2 nombres binaires, octaux et hexadécimaux (entiers, positifs et/ou négatifs)
- Conversion d'un nombre réel à virgule fixe, décimal, positif ou négatif, en nombre binaire à virgule fixe et réciproquement.

Le candidat devra d'abord analyser les domaines d'applications du sujet pour chaque base considérée, afin que le tout soit cohérent. Par exemple, avec 32 bits binaires signé, quel minima et quel maxima peut-on utiliser pour convertir en hexadécimal ?

1.3. MATÉRIEL ET LOGICIELS À DISPOSITION

1 ordinateur standard ETML, avec la structure habituelle.

1.4. PRÉREQUIS

Avoir suivi les modules ELEOC, et de programmation en C#.

1.5. CAHIER DES CHARGES

1.5.1. Objectifs et portée du projet (objectifs FFOR)

Force :

- Possède déjà des idées concernant les possibilités pour coder le programme afin d'effectuer des conversions, additions et soustractions

Faiblesse :

- Difficulté à rédiger le rapport

Opportunité :

- Créer un programme qui sera repris par l'ETML pour effectuer des corrections, vérifications et aider les élèves ainsi que les enseignants

Risque :

- Manque de temps/connaissances

1.5.2. Caractéristiques des utilisateurs et impacts

Les utilisateurs sont en 1^{ère} année de l'ETML ainsi que très probablement des utilisateurs d'autres années, certains n'ont pas de connaissances poussées de l'informatique, il faut donc que le programme soit simple d'utilisation et intuitif pour tous.

Le programme peut aider les élèves à se corriger, à comprendre ce qu'ils ont fait de faux et comment ils doivent faire pour effectuer une conversion, addition ou soustraction.

1.5.3. Fonctionnalités requises (du point de vue de l'utilisateur)

Conversions des nombres décimaux, binaires, octaux et hexadécimaux (entier et/ou réels à virgules fixes) en nombres dans les 3 autres bases que la base courante

Additions et soustractions de 2 nombres binaires, octaux et hexadécimaux (entiers, positifs et/ou négatifs)

Conversion d'un nombre réel à virgule fixe, décimal, positif ou négatif, en nombre binaire à virgule fixe et réciproquement.

Possibilité de voir le calcul entier, que ce soit pour la conversion, l'addition et la soustraction.

1.5.4. Contraintes

Les fonctions prédéfinies ne sont pas autorisées et ne doivent en aucun cas être utilisées.

1.5.5. Travail à réaliser par l'apprenti

L'apprenti doit créer un programme permettant aux élèves ainsi qu'aux enseignants de corriger les exercices d'ELEOC, le programme doit donc permettre la conversion de nombre positifs ou négatifs à virgule fixe entre différents formats (décimal, binaire, octal et hexadécimal), en plus de cela il doit être possible de soustraire et additionner des nombres entiers positifs et/ou négatifs dans différents formats (binaire, octal et hexadécimal), tout cela doit être simple d'utilisation pour chaque élève et enseignants de l'ETML en plus d'être intuitif.

Une planification doit être faite au début du projet avec un Gantt, un journal de travail doit aussi être fait et tenu à jour. Un rapport doit être fait et complet, il doit contenir l'intégralité du projet.

Tout le projet doit être mis sur un dépôt Git que l'élève pourra choisir par lui-même.

Une analyse du programme concernant l'application et ses fonctionnalités doit être faite, permettant de s'assurer de la cohérence du programme entre ses différentes fonctionnalités et utilisations.

1.5.6. Si le temps le permet

Améliorer l'apparence du programme pour le rendre visuellement plus beau et plus intuitif. Permettre de mettre en fichier PDF le résultat de l'opération ainsi que le calcul effectué pour l'obtenir.

1.6. LES POINTS SUIVANTS SERONT ÉVALUÉS

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20):

- Conversion de nombres entiers binaires signés dans les autres bases
- Conversion de nombres réels signés, à virgule fixe, dans les autres bases
- Conversion de nombres décimaux signés dans les autres bases
- Conversion de nombres décimaux signés, à virgule fixe, dans les autres bases

- Conversion de nombres hexadécimaux signés dans les autres bases
- Ergonomie intuitive et agréable pour tout type d'utilisateur de l'ETML
- La réalisation doit prendre en compte au moins un point significatif des bonnes pratiques en matière d'écoconception

2. PLANIFICATION INITIALE

La planification initiale se trouve en annexe.

3. ANALYSE

3.1. OPPORTUNITÉS

Approfondir ses connaissances en C#, opportunité de créer un programme qui sera utilisé en ELEOC, par les élèves de l'ETML en première année voir des années suivantes, ainsi que les enseignants qui pourront corriger des exercices.

3.2. DOCUMENT D'ANALYSE ET CONCEPTION

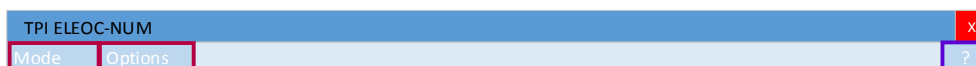
3.2.1. Les conversions minimal/maximal avec 32 bits signés

	Décimal	Binaire	Octal	Hexadécimal
Minimum	-2147483648	10000000000000000000000000000000 Ou 1 0 ^{x31}	20000000000 Ou 2 0 ^{x10}	8000 0000 Ou 8 0 ^{x7}
Maximum	2147483647	01111111111111111111111111111111 Ou 0 1 ^{x31}	17777777777 Ou 1 7 ^{x10}	7FFF FFFF Ou 7 F ^{x7}

3.2.2. Maquette graphique de manière générale

Les maquettes qui seront affichées représentent ce que le programme devait ressembler au début du projet, il est possible que ce ne soit pas le cas et que pour certaines raisons le produit final diffère de celles-ci.

Chaque interface possède la même barre d'action se trouvant tout en haut de l'interface du programme, comprenant **2 boutons avec une liste déroulante** et **1 bouton simple**. (Barre d'action affichée ci-dessous)



Le premier bouton avec une liste déroulante se nomme « **Mode** », cela permet à l'utilisateur de changer d'interface suivant l'opération voulue, il n'aura qu'à cliquer dessus pour qu'une liste apparaisse laissant le choix du mode pour convertir, additionner ou soustraire.

Le second bouton qui lui aussi est avec une liste déroulante nommée « **Options** » permettra à l'utilisateur de changer le format de son nombre,

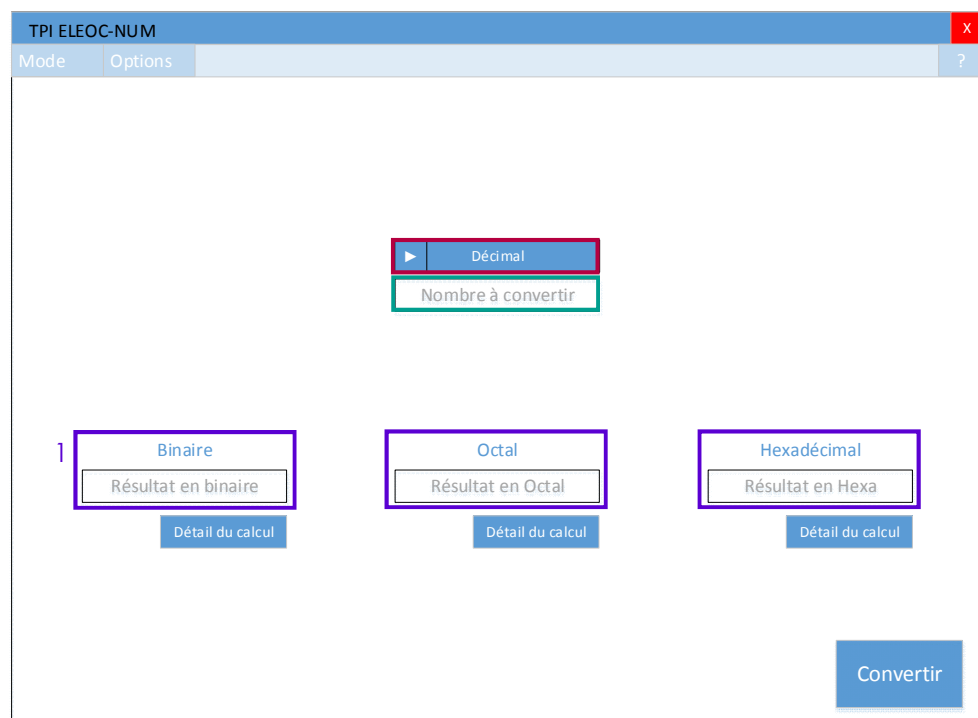
par exemple de choisir s'il veut effectuer la conversion voulue avec notamment un nombre à virgule ou non, il pourra aussi définir s'il s'agit d'un nombre signé ou non, cela permettra aux utilisateurs d'effectuer leurs opérations suivent ce qu'ils veulent et ce qu'ils ont appris, de plus cela évitera d'afficher des informations ou données qu'ils n'ont pas encore appris évitant ainsi de les embrouiller.

Un bouton d'aide « ? » sera disponible pour indiquer des informations concernant les nombres qu'ils peuvent entrer, le maximum et minimum possible, des aides concernant le manuel d'utilisation et de l'utilisation du programme.

3.2.3. Maquette graphique pour les conversions

Pour la première maquette il s'agit de celle concernant la conversion de nombres, il s'agit de l'interface qui s'affichera à l'utilisateur lorsque celui-ci ouvrira le programme.

Voici la maquette en question :



La maquette graphique de l'interface de conversion de bases est présentée dans une fenêtre intitulée "TPI ELEOC-NUM". Elle comporte une barre de menu avec "Mode" et "Options", et un bouton d'aide "?".

Le contenu principal de l'interface est structuré comme suit :

- Un bouton "Décimal" avec une flèche de sélection à gauche, et un champ "Nombre à convertir" juste en dessous.
- En dessous, trois boutons de sélection de format : "Binaire", "Octal" et "Hexadécimal".
- Chaque bouton de format est associé à un champ de résultat ("Résultat en binaire", "Résultat en Octal", "Résultat en Hexa") et un bouton "Détail du calcul".
- Un bouton "Convertir" est situé en bas à droite.

Le bouton "Binaire" est marqué avec un "1" à sa gauche, indiquant qu'il est sélectionné par défaut.

Concernant le nombre à convertir, il est possible de définir son format (binaire, décimal, octal ou hexadécimal) via **la liste déroulante**, en choisissant un format cela modifiera par la même occasion **les autres champs** où seront affichés les résultats. Par exemple si l'utilisateur ne veut plus convertir un nombre décimal mais binaire, lorsqu'il aura sélectionné le format binaire via **la liste déroulante**, à ce moment-là la zone de résultat en binaire (1) deviendra un résultat en décimal avec non pas le titre « Binaire » mais bien « Décimal ».

L'utilisateur entre la valeur qu'il veut contenir dans la zone de texte mis à sa disposition, cependant si l'utilisateur a choisi que le nombre à convertir est un nombre à virgule il y aura 2 zones de texte l'une à côté de l'autre, entre eux un point servira à les séparer en indiquant par la même occasion où se trouve la virgule, dans la première zone il s'agira du nombre se trouvant avant la virgule, la deuxième zone sera donc celle pour les nombres se trouvant après la virgule.

Voici ce à quoi ressemble l'interface lorsque l'utilisateur choisi de convertir un nombre à virgule :

▶

Décimal

Nombre avant la virgule

.

Nombre après la virgule

Même si les calculs ne sont pas affichés de base il est cependant possible d'y avoir accès via le bouton « Détail du calcul »,

Seul les résultats de la conversion sont affichés et non les calculs pour y parvenir, cela dans le but de ne pas afficher beaucoup d'informations à l'écran évitant d'embrouiller l'utilisateur et par la même occasion que cela soit illisible.

Pour chaque conversion il est possible de voir le calcul, pour cela il suffit de cliquer sur détail du calcul. Voici un exemple de conversion de décimal à binaire :

Détail du calcul en binaire
✕

97/2 = 48
48/2 = 24
24/2 = 12
12/2 = 6
6/2 = 3
3/2 = 1
1/2 = 0

Reste 1
Reste 0
Reste 0
Reste 0
Reste 0
Reste 1
Reste 1

Tableau de puissance

2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	0	0	0	1

Résultat

97 = 1100001

↑

↓

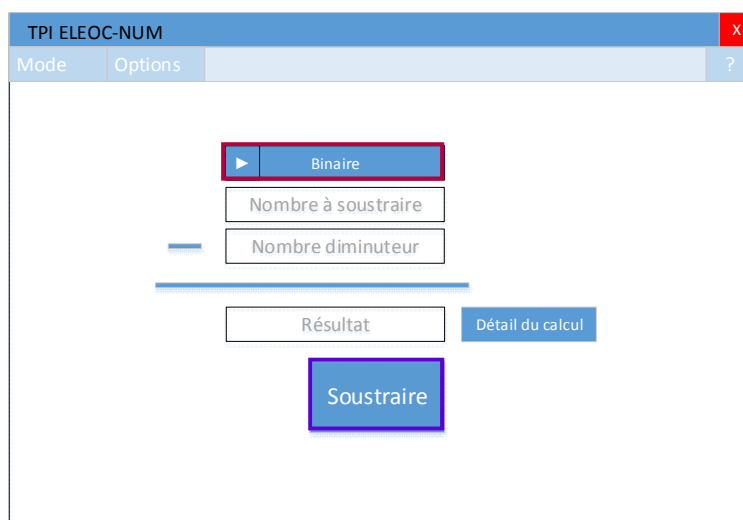
Lire de bas en haut

Les étapes sont toutes affichées, le « Reste » est affiché et une flèche partant de bas en haut indique dans quel sens il doit être lu, cela permet

de comprendre comment faire le calcul soi-même et de comprendre son erreur lors des corrections notamment, ainsi si l'utilisateur n'est pas à l'aise en ELEOC ou qu'il a de la peine, cela lui permettra de s'améliorer et de l'aider. Un tableau des puissances où sont entré **les restes** est visible, en dessous se trouve le résultat du calcul.

3.2.4. Maquette graphique pour les soustractions

Pour la deuxième maquette il s'agit de celle concernant la soustraction. Celle-ci s'affichera lorsque l'utilisateur choisira le mode « Soustraire ».

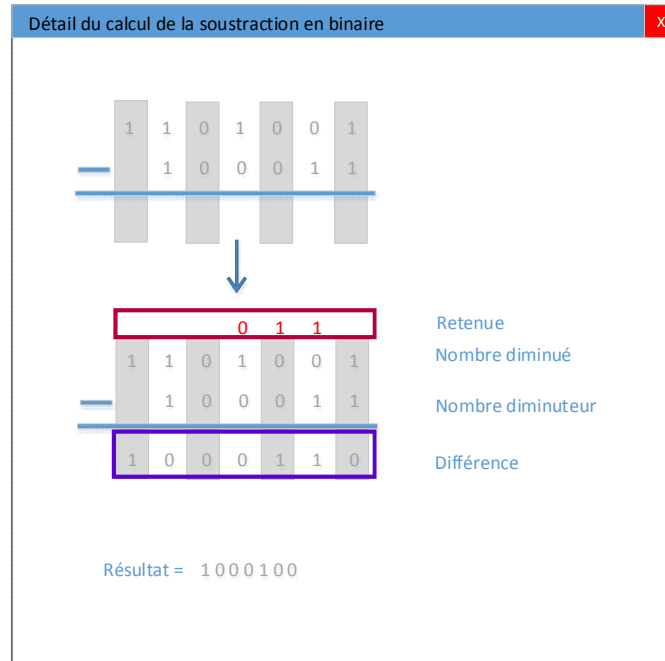


Maquette graphique pour la soustraction. L'interface est intitulée "TPI ELEOC-NUM" et comporte une barre de mode avec "Mode" et "Options". Le mode "Soustraire" est sélectionné. Le formulaire contient :

- Un bouton "Binaire" avec une flèche de sélection.
- Un champ "Nombre à soustraire".
- Un champ "Nombre diminueur".
- Un champ "Résultat".
- Un bouton "Détail du calcul".
- Un bouton "Soustraire" en bas.

Il est possible de choisir le format du nombre entré via **la liste déroulante**, l'utilisateur une fois avoir choisi le format du nombre qu'il veut soustraire, peut par la suite entrée le nombre qu'il veut soustraire dans la case « Nombre à soustraire » et le nombre diminueur dans la case « Nombre diminueur », un bouton « **Soustraire** » permet d'effectuer l'opération afin d'obtenir le résultat dans la case « Résultat ».

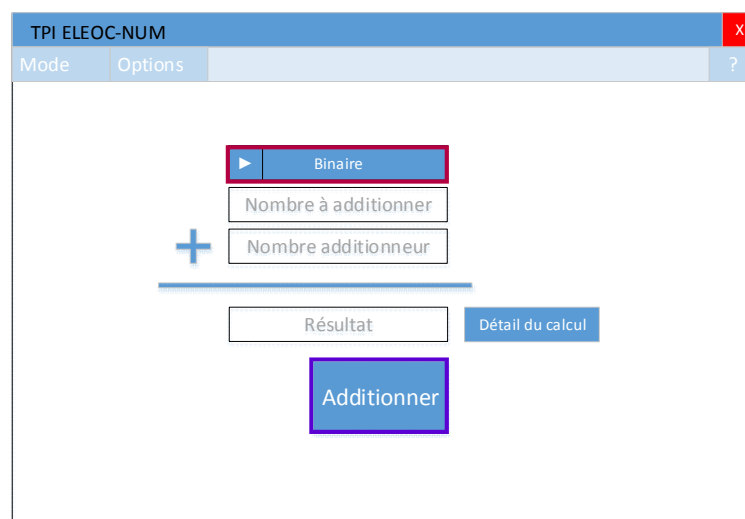
Le bouton «Détail du calcul» permet d'affiche tout le calcul nécessaire pour obtenir le résultat. (Image ci-dessous)



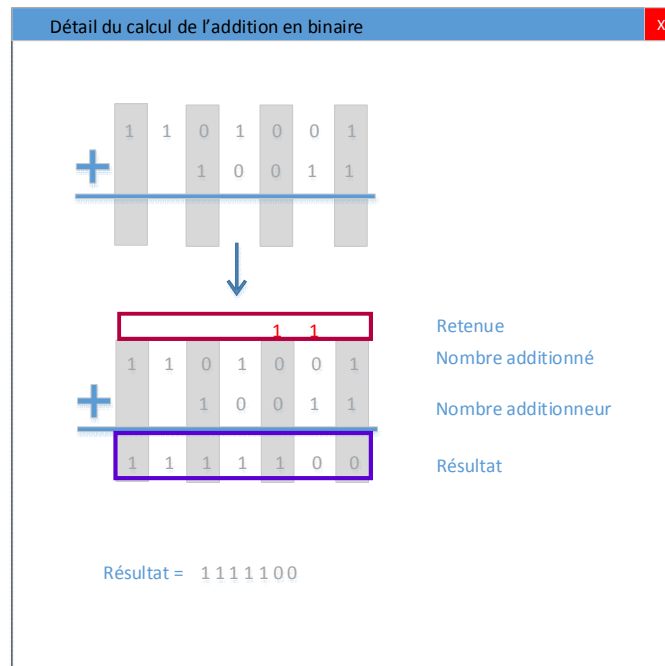
Dans la première partie l'équation est simplement posée affichant à l'utilisateur ce qu'il a entré, en dessous l'opération commence affichant **les retenues** en dessus, une colonne à droite indique quelques informations à l'utilisateur (retenue, nombre diminué, nombre diminueur et différence), **le résultat** est affiché en dessous du calcul, il est remis proprement en dessous.

3.2.5. Maquette graphique pour les additions

Cette maquette reprend l'interface de la maquette concernant le mode de soustraction, l'apparence est donc presque identique à celle-ci. (Image ci-dessous)



Il est donc possible comme pour la soustraction, de choisir le format du nombre entré via la **liste déroulante**, l'utilisateur une fois avoir choisi le format du nombre qu'il veut additionner, peut par la suite entrée le nombre qu'il veut additionner dans la case « Nombre à additionner » et le nombre additionneur dans la case « Nombre additionneur », un bouton « **Additionner** » permet d'effectuer l'opération afin d'obtenir le résultat dans la case « Résultat ».



Dans la première partie l'équation est simplement posée affichant à l'utilisateur ce qu'il a entré, en dessous l'opération commence affichant **les retenues** en dessus, une colonne à droite indique quelques informations à l'utilisateur (retenue, nombre diminué, nombre diminuteur et différence), **le résultat** est affiché en dessous du calcul, il est remis proprement en dessous.

3.2.6. Conception du programme

Le programme a besoin de pouvoir effectuer différentes opérations de calculs (conversion, addition et soustraction), afin d'éviter que cela nécessite trop de ressource ces opérations seront séparées, le programme possèdera des « modes », cela permettra de n'effectuer que l'opération voulue en affichant uniquement l'interface du « mode » choisi sans afficher l'interface des autres, pour changer de « mode » le programme possèdera une barre d'action avec un « Split Button », il s'agit d'un bouton avec une liste déroulante qui permettra de choisir le « mode » désiré pour soit convertir, additionner ou soustraire.

Dans cette même barre d'action il y aura encore un autre « Split Button », celui-ci servira à choisir différentes options comme la possibilité de définir notamment si l'on veut convertir des nombres avec virgule ou non, permettant aux élèves n'étant pas à l'aise en ELEOC de ne pas

s'embrouiller à remplir de zéro ou de ne rien mettre dans la case après la virgule, donnant par la même occasion un résultat qui n'affichera aucune virgule évitant aussi de s'embrouiller et les erreurs de manipulation en entrant des chiffres après la virgule par erreur.

Pour la conversion il y aura donc comme vu au-dessus la possibilité de le faire sans virgule donnant uniquement des conversions avec des chiffres entiers, il n'y aura donc qu'une boîte de texte où l'utilisateur entrera la valeur qu'il veut convertir. Si l'utilisateur choisi d'effectuer une conversion avec virgule, une deuxième boîte de texte apparaîtra à droite de celle de base, un point s'affichera entre ces deux, il s'agit de la virgule pour ainsi les séparer en indiquant que la deuxième contiendra ce qu'il y a après la virgule, la séparation de ce qu'il y a avant la virgule et après permet d'éviter des erreurs du programme via notamment les regex qui seront utilisés pour vérifier le format des valeurs entrées ainsi que faciliter la conversion de ce qu'il y a avant la virgule et après.

Au sujet de la façon dont les conversions seront faites, des tableaux seront utilisés pour convertir, un algorithme sera mis en place pour afficher à l'utilisateur le calcul effectué et la démarche pour réussir à obtenir le résultat, le but est que l'utilisateur puisse obtenir le résultat avec les démarches nécessaires au calcul mais aussi qu'il puisse comprendre comment l'atteindre, cela lui permettra de se corriger, de plus s'il a des difficultés pour effectuer le calcul de lui-même il faut que le programme, en affichant le calcul puisse le guider et l'aider à comprendre comment il doit faire.

Les additions et les soustractions utiliseront aussi des tableaux, certains tableaux contiendront les résultats des calculs et d'autres « les restes » afin d'afficher le calcul en entier, une comparaison entre les cellules du tableau permettra d'effectuer les calculs notamment en binaire.

3.3. CONCEPTION DES TESTS

Test à effectuer	Ce qui est recherché
Faire des conversions, additions et soustractions avec des lettres ou autres caractères lorsque le format ne doit pas le permettre	<ol style="list-style-type: none">1. Vérifier s'il y a des erreurs ou crash du programme2. Empêcher une mauvaise manipulation
Vérifier les limites possibles concernant la taille des calculs ou des nombres	<ol style="list-style-type: none">1. Vérifier s'il y a des erreurs ou crash du programme2. Limiter la taille maximal si besoin
Fournir le manuel d'utilisation avec le programme à plusieurs personnes pour qu'elles utilisent le programme	<ol style="list-style-type: none">1. Vérifier la simplicité d'utilisation du programme2. S'assurer que le manuel d'utilisation soit clair et complet

Tester sur différent PC n'ayant pas la même configuration	<ol style="list-style-type: none"> 1. Voir comment le programme réagit, s'il est bien compatible et qu'il n'y a aucune erreur 2. Si ce n'est pas le cas il sera possible de soit corriger, soit indiquer pourquoi il ne fonctionne pas
Essayer de convertir, additionner et soustraire avec un nombre supérieur à 32 bits	<ol style="list-style-type: none"> 1. Voir si le programme empêche bien la manipulation lorsque la limite est dépassée
Convertir, additionner et soustraire de petites valeurs jusqu'au maximum possible	<ol style="list-style-type: none"> 1. Vérifier si les fonctionnalités du programme fonctionnent correctement 2. Vérifier si une petite valeur pose problème et inversement avec une grande valeur
Effectuer les calculs avec nombres positifs et négatifs ainsi qu'avec virgule	<ol style="list-style-type: none"> 1. Vérifier le bon fonctionnement des calculs

3.4. PLANIFICATION DÉTAILLÉE





La colonne concernant les liens et références de la planification n'a pas été reprise ci-dessous, la raison étant qu'il n'y a aucune information écrite sous celle-ci, de plus en retirant ces colonnes cela permet une meilleure visibilité.

Semaine 1		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
Lundi - 7 mai		Max. 12 (3h)
Entretien	2	Entretien concernant le TPI.
Planification initiale	8	Création de la planification initiale et mise à jour.
Journal de travail	2	Création du journal de travail et mise à jour.
Mercredi - 9 mai		Max. 27 (4h45)
Absence - Imprévis	15	Joutes sportives.
Planification initiale	4	Finalisation de la planification initiale.
Journal de travail	2	Mise à jour du journal de travail.
Rapport	4	Création du rapport, rédaction de la partie Analyse et Planification initiale, mise à jour avec les informations reçus.
Réalisation de croquis	2	Création de croquis avec Visio ou un autre logiciel pour le programme, emplacement des boutons, fonctionnalités, onglets, différentes pages,
Jeudi - 10 mai		Max. 24 (4h)
Congé - Vacances	24	Ascension.
Vendredi - 11 mai		Max. 27 (4h45)
Congé - Vacances	27	Pont avec l'Ascension.
Total semaine		Max. 90 (22h30)

Semaine 2		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
Lundi - 14 mai		Max. 12 (3h)
Réalisation de croquis	4	Finalisation des croquis du programme.
Rapport	6	Finalisation de la partie Analyse, planification initiale.
Entretien	2	Entretien avec M. Gruaz concernant l'avancement du projet.
Mercredi - 16 mai		Max. 27 (4h45)
Journal de travail	1	Mise à jour du journal de travail.
Recherche	8	Recherche concernant la conversion du décimal au binaire et inversement.
Coder le programme	12	Création du programme avec les norme ETML et des commentaires. Mise en place de l'interface pour convertir et préparation pour les futurs opérations (addition, soustraction, etc...). Ajout d'options pour choisir entre les conversions. Ajout de commentaire.
Coder pour convertir	6	Mettre en place les paramètres et le code pour convertir du décimal à binaire et inversement.
Jeudi - 17 mai		Max. 24 (4h)
Journal de travail	1	Mise à jour du journal de travail.
Recherche	8	Recherche concernant la conversion en octal et hexadécimale.
Coder pour convertir	8	Finalisation et correction du code pour convertir de décimal à binaire et inversement.
Coder le programme	7	Avancement dans le programme.
Vendredi - 18 mai		Max. 27 (4h45)
Journal de travail	1	Mise à jour du journal de travail.
Coder pour convertir	6	Finalisation de la conversion en octal et hexadécimale.
Coder le programme	8	Correction du code, vérification du bon fonctionnement et ajustement au norme ETML.
Réalisation de croquis	4	Création du croquis pour les additions.
Rapport	8	Avancement du rapport avec les éléments actuel.
Total semaine		Max. 90 (22h30)

Semaine 3		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
Lundi - 21 mai		Max. 12 (3h)
Congé - Vacances	12	Lundi de Pentecôte.
Mercredi - 23 mai		Max. 27 (4h45)
Journal de travail	1	
Entretien	2	Entretien avec M. Gruaz concernant l'avancement du projet.
Recherche	12	Recherche sur un moyen d'additionner 2 nombres binaire.
Coder le programme	4	Mise en place de l'interface pour additionner.
Coder pour additionner	8	Avancement dans l'addition avec 2 nombre binaire.
Jeudi - 24 mai		Max. 24 (4h)
Journal de travail	1	Mise à jour du journal de travail.
Recherche	9	Recherche pour additionner avec 2 nombres binaire, octaux et
Coder le programme	6	Avancement du code avec correction.
Coder pour additionner	8	Finalisation pour additionner avec 2 nombres binaires, octaux et hexadécimaux.
Vendredi - 25 mai		Max. 27 (4h45)
Journal de travail	1	Mise à jour du journal de travail.
Réalisation de croquis	4	Création du croquis pour soustraire.
Coder le programme	8	Correction du code si nécessaire et préparation de l'interface pour
Coder pour soustraire	4	Préparation pour soustraire 2 nombres binaires.
Tests	3	Test sur le bon fonctionnement du programme.
Rapport	7	Avancement du rapport avec les éléments actuels.
Total semaine	90	Max. 90 (22h30)

Semaine 4		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
Lundi - 28 mai		Max. 12 (3h)
Journal de travail	1	Mise à jour du journal de travail.
Entretien	2	Entretien avec M. Gruaz concernant l'avancement du projet.
Manuel d'utilisation	4	Création du manuel d'utilisation.
Recherche	5	Recherche pour soustraire 2 nombres binaires.
Mercredi - 30 mai		Max. 27 (4h45)
Journal de travail	1	Mise à jour du journal de travail.
Recherche	10	Recherche pour soustraire 2 nombres binaires, octaux et hexadécimaux.
Coder pour soustraire	10	Avancement du code pour soustraire 2 nombres binaires, octaux et
Coder le programme	6	Avancement dans le code avec correction.
Jeudi - 31 mai		Max. 24 (4h)
Journal de travail	1	Mise à jour du journal de travail.
Coder pour soustraire	10	Finalisation du code pour soustraire 2 nombres binaires, octaux et
Coder le programme	9	Correction, vérification du code et ajustement si nécessaire.
Tests	4	Vérification des normes de codage de l'ETML.
Vendredi - 1 juin		Max. 27 (4h45)
Journal de travail	1	Mise à jour du journal de travail.
Rapport	12	Avancement de le rapport avec les informations actuels.
Manuel d'utilisation	6	Vérification et correction du rapport.
Coder le programme	6	Avancement dans le manuel d'utilisation.
Tests	2	Vérification du code et correction si nécessaire.
Total semaine	90	Max. 90 (22h30)

Semaine 5		
Tâche	Durée [1/4 h.]	Explications: qu'est-ce qui se fait et comment ?
Lundi - 4 juin 		
	12	Max. 12 (3h)
Journal de travail	1	Mise à jour du journal de travail.
Rapport	6	Avancement dans le rapport.
Manuel d'utilisation	3	Avancement du manuel d'utilisation.
Entretien	2	Entretien avec M. Gruaz concernant l'avancement du projet.
Mercredi - 6 juin 		
	27	Max. 27 (4h45)
Journal de travail	3	Mise à jour du journal de travail avec correction si nécessaire.
Rapport	18	Finalisation du rapport.
Manuel d'utilisation	6	Finalisation du manuel d'utilisation.
Jeudi - 7 juin 		
	24	Max. 24 (4h)
Journal de travail	4	Mise à jour du journal de travail avec correction si nécessaire.
Rapport	10	Vérification du rapport et correction si nécessaire.
Manuel d'utilisation	6	Vérification du manuel d'utilisation et correction si nécessaire.
Fin du projet	4	Vérification que tout soit en ordre, envoi du projet au chef de projet et
Vendredi - 8 juin 		
	27	Max. 27 (4h45)
Fin du projet	27	Projet terminé et rendu.
Total semaine	90	Max. 90 (22h30)

4. RÉALISATION

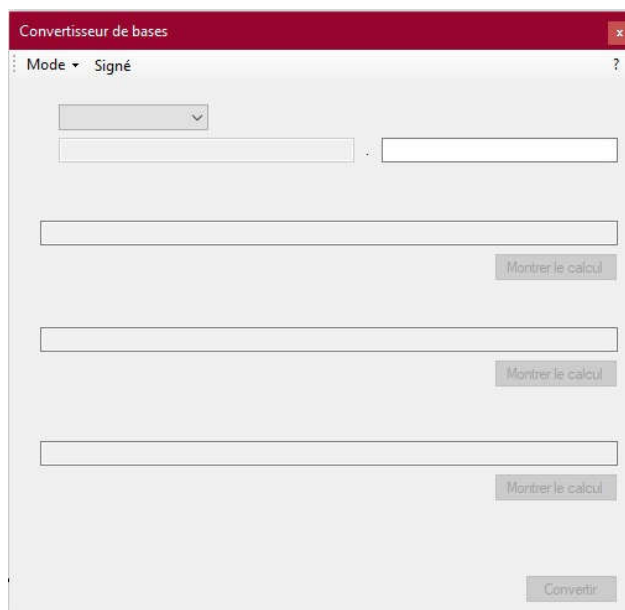
4.1. FONCTIONNALITÉS

4.1.1. Indexe des abréviations pour les noms d'objet

Nom complet	Abréviation
Combo Box	cob
Form	frm
Label	lbl
Text Box	txb
Tool Strip	ts
Tool Strip Button	tsb
Tool Strip Drop Down Button	tsddb
Tool Strip Menu Item	tsmi

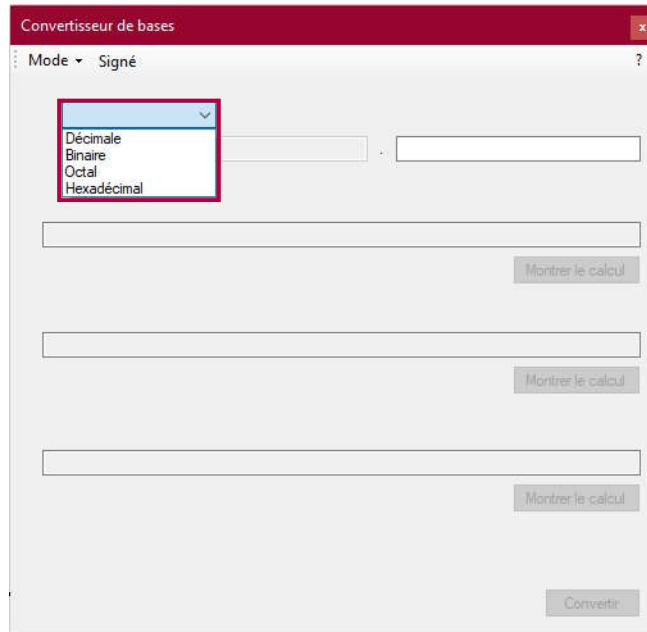
4.1.2. Lancement du programme

Lors du lancement du programme, plusieurs objets du programme sont modifiés, notamment le bouton pour convertir, les boutons pour afficher le calcul, la zone de texte où l'utilisateur entre la valeur à convertir, les noms concernant en quoi le nombre de l'utilisateur sera converti. N'affichant que le minimum à l'utilisateur avant qu'il fasse le choix du format duquel sera converti le nombre qu'il choisira. (Image de l'interface lors de l'ouverture ci-dessous)



4.1.3. Comment entrer une valeur à convertir

Il faut tout d'abord **choisir le format** du quel l'utilisateur veut effectuer la conversion avant de convertir. (Image ci-dessous)



Une fois le format choisi la case où l'utilisateur peut entrer le nombre à convertir devient disponible. Une vérification du nombre entré lorsque l'utilisateur choisi un format se fait, par exemple si l'utilisateur choisi le format décimal et a entré un nombre en décimal comme « 9148 » mais venait à changer de format pour le binaire, dans ce cas la case où l'utilisateur entre son nombre ainsi que le bouton de conversion deviendront indisponibles pour empêcher d'effectuer une conversion qui n'est pas possible en plus de se mettre en rouge afin d'indiquer que le format est incorrect.

4.1.4. Vérifications du nombre entré par l'utilisateur

Lorsque l'utilisateur entre une valeur, il est immédiatement vérifié, en premier le programme vérifie s'il est signé, si tel est le cas, alors il l'ajoute au nombre entré par l'utilisateur. (Image ci-dessous)

```
/// <summary>
/// Vérifie le nombre entré par l'utilisateur correspond au format choisi
/// </summary>
/// <param name="sender">Information sur l'objet</param>
/// <param name="e">Information sur l'événement</param>
private void txbValueUserBeforePoint_TextChanged(object sender, EventArgs e)
{
    string valueUserToCheck = "";

    // Vérifie s'il s'agit d'une conversion signée
    if(lblSign.Text == "+" || lblSign.Text == "-")
    {
        valueUserToCheck = lblSign.Text + txbValueUserBeforePoint.Text;
    }
    else
    {
        valueUserToCheck = txbValueUserBeforePoint.Text;
    }
}
```

Ensuite, le programme va récupérer le format choisi par l'utilisateur, ainsi il sera possible d'adapter les vérifications selon le format choisi, si l'utilisateur a choisi le format décimal à convertir, dans ce cas une vérification avec un regex « checkValDec » sera utilisé. (Image ci-dessous de ce regex) (Image ci-dessous)

```
// Regex pour vérifier si l'utilisateur a entré uniquement des chiffres
Regex checkValDec = new Regex("[0-9]+$");
Regex checkValBin = new Regex("[0-1]+$");
```

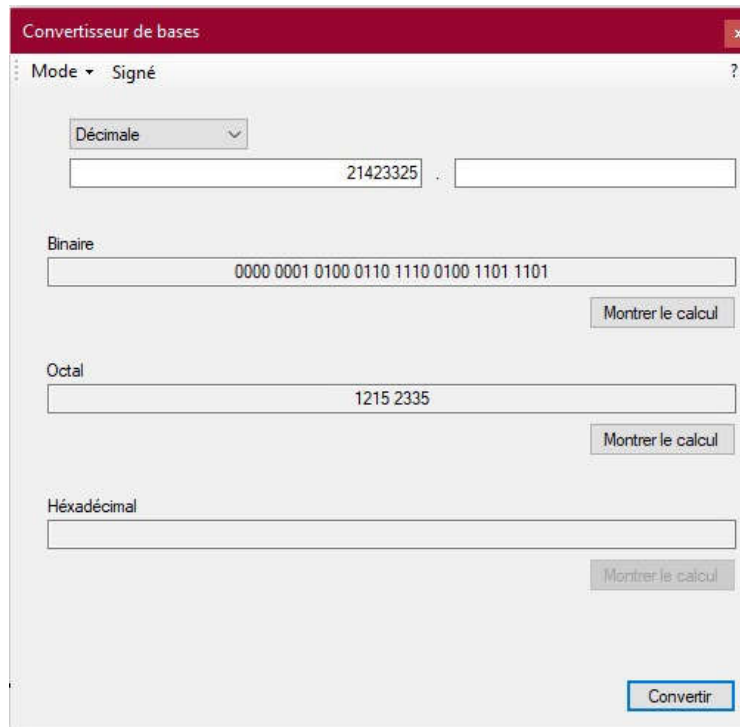
Mais aussi un « int.TryParse », cela permet d'essayer de convertir en int 32, si cela fonctionne une variable est défini avec la valeur de l'utilisateur, si au contraire cela ne fonctionne pas alors le programme ne le converti pas et il sort du if. Si au contraire cela fonctionne il va donc récupérer la valeur dans une variable ainsi que rendre disponible le bouton de conversion à l'utilisateur. (Image ci-dessous pour le format binaire)

```
// Permet de vérifier si le format du nombre entré correspond au format choisi
switch (formatSelect)
{
    // Si l'utilisateur a choisi de convertir du décimal
    case DEC_TEXT:

        // Vérifie si le nombre entré correspond au format autorisé
        if (checkValDec.IsMatch(txbValueUserBeforePoint.Text) && int.TryParse(valueUserToCheck, out valueUser))
        {
            valueUser = Convert.ToInt32(txbValueUserBeforePoint.Text);
            txbValueUserBeforePoint.BackColor = Color.White;
            btnConvert.Enabled = true;
        }
        else
        {
            txbValueUserBeforePoint.BackColor = Color.MediumVioletRed;
            btnConvert.Enabled = false;
        }
        break;
```

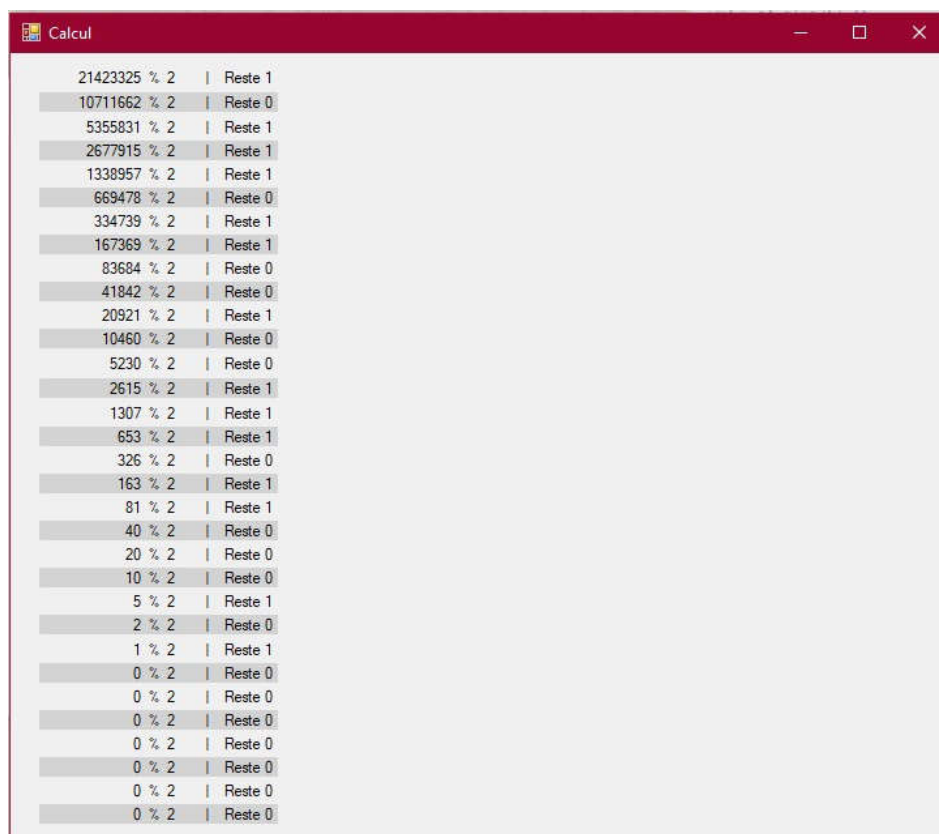
4.1.5. Conversion de décimal à binaire avec nombre entier signé entier

Lorsque l'utilisateur entre le nombre qu'il veut convertir et clique sur le bouton « Convertir », le bouton « Montrer le calcul » devient disponible afin d'afficher le calcul de conversion au complet. (Image ci-dessous)



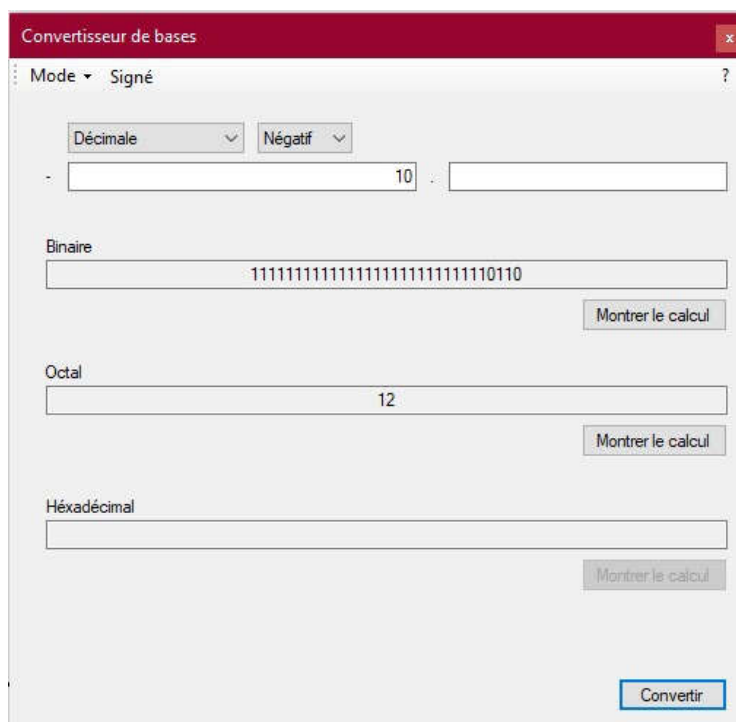
Le résultat affiché est sur 32 bits, il n'est pas obligatoire qu'il soit signé, en effet il est possible d'effectuer l'opération sans, mais il est aussi possible de l'effectuer avec pour cela il faut aller sur « Signé » dans la barre des tâches et cliquer sur « Oui » ou « Non ».

Pour la conversion sans le nombre entré est divisé par deux, si le chiffre est impair alors le reste qui n'est autre que le surplus empêchant la division par deux sans que cela devienne un chiffre à virgule est stocké dans un tableau, le résultat de la division est aussi stocké dans un second tableau et cela est fait en boucle, s'ils sont stockés dans un tableau chacun c'est pour pouvoir les récupérer lorsque l'on veut afficher le calcul. (Image démontrant comment est affiché le calcul)



21423325	% 2		Reste 1
10711662	% 2		Reste 0
5355831	% 2		Reste 1
2677915	% 2		Reste 1
1338957	% 2		Reste 1
669478	% 2		Reste 0
334739	% 2		Reste 1
167369	% 2		Reste 1
83684	% 2		Reste 0
41842	% 2		Reste 0
20921	% 2		Reste 1
10460	% 2		Reste 0
5230	% 2		Reste 0
2615	% 2		Reste 1
1307	% 2		Reste 1
653	% 2		Reste 1
326	% 2		Reste 0
163	% 2		Reste 1
81	% 2		Reste 1
40	% 2		Reste 0
20	% 2		Reste 0
10	% 2		Reste 0
5	% 2		Reste 1
2	% 2		Reste 0
1	% 2		Reste 1
0	% 2		Reste 0
0	% 2		Reste 0
0	% 2		Reste 0
0	% 2		Reste 0
0	% 2		Reste 0
0	% 2		Reste 0
0	% 2		Reste 0

Si l'utilisateur choisi de convertir un nombre signé cela ne s'arrête pas là, si le nombre est positif alors cela ne change pas, cependant s'il est négatif alors il va effectuer d'autres opérations en plus, il va donc convertir en binaire positif, mais par la suite il va inverser les « 0 » et les « 1 » et ajouter un bit au tout début. (Exemple de résultat en binaire négatif ci-dessous)



Convertisseur de bases

Mode : Signé

Décimale Négatif

- 10

Binaire

111111111111111111111111111110110

Montrer le calcul

Octal

12

Montrer le calcul

Héxadécimal

Montrer le calcul

Convertir

L'affichage du calcul ne fonctionne pas pour les nombre signé négatif et la conversion de décimal signé négatif en octal ne fonctionne pas.

4.1.6. Conversion de décimal à binaire avec nombre à virgule

Concernant la conversion de décimal à binaire avec virgule, il est possible d'effectuer cette conversion en non signé, la valeur après la virgule est multipliée par 2 jusqu'à ce qu'il n'y ait plus rien après la virgule, ce qui dépasse de la virgule est retenu dans un tableau afin d'afficher le résultat à l'utilisateur.

4.1.7. Conversion de décimal à octal avec nombre entier

Le fonctionnement est le même que celui de décimal à binaire à quelques exceptions près, tout d'abord la division se fait par 8, le résultat et le reste empêchant le résultat d'être un chiffre qui ne soit pas à virgule est retenu dans un tableau chacun pour être affiché à l'utilisateur lorsque celui-ci veut afficher le calcul en entier.

4.1.8. Conversion de binaire à décimal avec nombre entier

Pour cette conversion une boucle effectue l'opération, elle permet à chaque fois d'effectuer, Pour le calcul il s'agit de ce que contient la table (soit 0 ou 1) multiplier par 2 à la puissance de la position dans la table, donc si le bit est à 0 le résultat sera de 0 quoi qu'il arrive pour cette multiplication, s'il s'agit du 4^{ème} bit cela signifie que le 2 sera à la puissance de 4 pour ainsi à la fin additionner chacun des résultats. (Image concernant cette conversion ci-dessous)

```

/// <summary>
/// Fonction permettant de convertir de binaire à décimal le nombre entré par l'utilisateur
/// </summary>
/// <returns>Nombre de caractère à afficher</returns>
private int ConvertBinToDec()
{
    int nbrBitsInTabDec = 0;
    string valueUserToConvert = Convert.ToString(valueUser);

    int lengthValueUser = valueUserToConvert.Length;

    // Boucle permettant de convertir de binaire en décimal
    for (int posTable = lengthValueUser; posTable > 0; posTable--)
    {
        tabConvCalculDec[posTable - 1] = Convert.ToInt32(valueUserToConvert.Substring(lengthValueUser - posTable, 1));
        tabConvDec[posTable - 1] = tabConvCalculDec[posTable - 1] * Convert.ToInt32(Math.Pow(Convert.ToDouble(2), Convert.ToDouble(posTable - 1)));
        nbrBitsInTabDec++;
    }

    return nbrBitsInTabDec;
}

```

4.1.9. Conversion de binaire à octal avec nombre entier

Le fonctionnement est similaire à celui de décimal à binaire, cependant la particularité avec l'octal est qu'il faut regrouper par groupe de 3 bits, voici un tableau pour mieux comprendre cela :

Puissance de 2	2 ²	2 ¹	2 ⁰	2 ²	2 ¹	2 ⁰	2 ²	2 ¹	2 ⁰
En binaire	1	0	0	0	0	1	0	1	0
En octal	4			1			2		
Total en octal	412								

Comme pour le binaire chaque résultat est gardé dans un tableau pour pouvoir les afficher.

4.2. DOSSIER DE RÉALISATION

4.2.1. Logiciels nécessaires

Microsoft Visual Studio 2017, cependant la version de Visual Studio avec la version de NET Framework 4.5.2 est nécessaire, sans cela le programme ne peut se lancer.

5. TESTS

5.1. DOSSIER DES TESTS

Tests effectués	Résultat
Conversion de décimal à binaire avec des lettres ou nombre plus grand que ce qui est prévu	<ul style="list-style-type: none">• Le programme empêche la conversion• la case où le nombre a été entré devient rouge• Le bouton de conversion devient inutilisable par l'utilisateur
Vérifier les limites possibles concernant la taille des calculs ou des nombres	<ul style="list-style-type: none">• Après vérification il est possible de convertir de décimal jusqu'à « 2147483647 », cependant en négatif une erreur survient si le nombre est inférieur à « - 2147483647 »• Pour la conversion à partir d'un nombre binaire il est possible d'aller uniquement jusqu'à « 111111111 » la raison n'a pas été trouvée
Fournir le manuel d'utilisation avec le programme à plusieurs personnes pour qu'elles utilisent le programme	Ce test n'a pas pu être effectué, le programme n'étant pas terminé il manque certaines fonctionnalités, le manuel d'utilisation n'a donc pas été créé
Tester sur différent PC n'ayant pas la même configuration	Le test a été fait sur un ordinateur, celui-ci possédait Windows 10 avec Visual Studio 2017, pour pouvoir lancer le programme il faut avoir installé Net Framework

	version 4.5.2, sans cela le programme ne fonctionne pas
Essayer de convertir, additionner et soustraire avec un nombre supérieur à 32 bits	<ul style="list-style-type: none">• Pour la conversion à partir d'un nombre en décimal le programme empêche la conversion lorsque le nombre est supérieur à 32• Concernant la conversion à partir d'un nombre en binaire, le test n'a pu être effectué car le programme empêche la conversion lorsque le nombre atteint « 111111111 » comme si celui-ci dépassait 32 bits, cela a été confirmé lors de la vérification de conversion en int32, qui si la conversion n'est pas possible s'arrête empêchant ainsi la conversion• En ce qui concerne l'addition et la soustraction ces fonctionnalités ne sont pas disponibles par le programme et non donc pas pu être vérifiées
Effectuer différentes conversions à virgule avec différentes valeurs et plusieurs fois d'affilé	<ul style="list-style-type: none">• Pour la conversion de décimal à binaire, tout fonctionne correctement, la conversion fonctionne jusqu'à 32 bits après la virgule, cependant rien est indiqué à l'utilisateur que le programme ne va pas plus loin que 32 bits• Pour la conversion de décimal à octal, la première conversion fonctionne mais les prochaines conversions si le programme n'est pas relancé sont fausses
Effectuer les calculs avec nombres positifs et négatifs ainsi qu'avec virgule	<ul style="list-style-type: none">• Les conversions des nombres à virgule ne peuvent être signés, il est possible de soit convertir en signé soit à virgule et non signé à virgule

6. CONCLUSION

6.1. BILAN DES FONCTIONNALITÉS DEMANDÉES

Concernant les fonctionnalités disponibles par le programme, la première est celle de la conversion, il est donc possible de convertir de :

- Décimal à binaire signé
- Décimal à virgule en binaire
- Décimal en octal non signé
- Décimal à virgule en octal
- Binaire en décimal non signé
- Binaire en octal non signé

6.1.1. Conversion de décimal signé à binaire

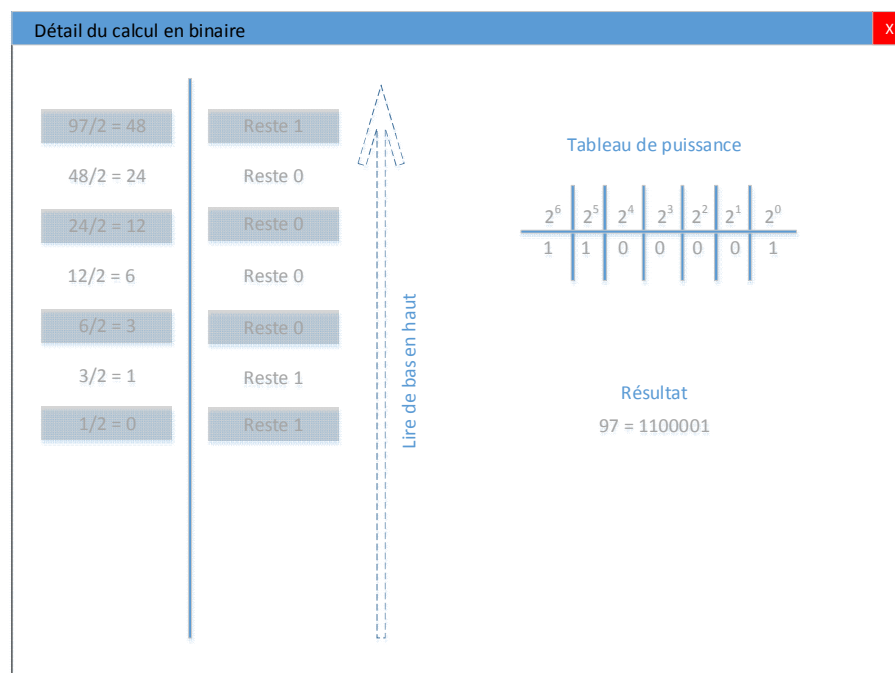
La conversion de décimal signé à binaire fonctionne correctement, pour le positif il n'y a aucun problème pour convertir un nombre jusqu'à un maximum de 32 bits, pour les nombre négatif la conversion fonctionne correctement jusqu'à un maximum de 32 bits, cependant une erreur survient concernant la conversion d'un nombre négatif en particulier, il s'agit du nombre « -2147483648 » qui est le nombre maximal négatif sur 32 bits. Cette erreur est due au code qui le récupère sous forme positive pour la conversion, ce qui donne le nombre « 2147483648 » qui est supérieur à 32 bits.

Lorsque celui-ci est positif il est alors possible d'afficher le calcul nécessaire pour obtenir le résultat. Cependant en le comparant avec ce qui était prévu dans la maquette, le résultat de chaque calcul n'est pas mis dans un tableau avec la table des puissances, une flèche indiquant dans quel sens le résultat du calcul doit être n'a pas été mise ainsi que le résultat, ces éléments doivent donc être rajouté, permettant à l'utilisateur de mieux comprendre le fonctionnement de la conversion, de pouvoir retrouver son erreur s'il a mal converti et de s'améliorer.

Ce que le programme affiche lorsque l'utilisateur veut afficher le calcul :

21423325 % 2	Reste 1
10711662 % 2	Reste 0
5355831 % 2	Reste 1
2677915 % 2	Reste 1
1338957 % 2	Reste 1
669478 % 2	Reste 0
334739 % 2	Reste 1
167369 % 2	Reste 1
83684 % 2	Reste 0
41842 % 2	Reste 0
20921 % 2	Reste 1
10460 % 2	Reste 0
5230 % 2	Reste 0
2615 % 2	Reste 1
1307 % 2	Reste 1
653 % 2	Reste 1
326 % 2	Reste 0
163 % 2	Reste 1
81 % 2	Reste 1
40 % 2	Reste 0
20 % 2	Reste 0
10 % 2	Reste 0
5 % 2	Reste 1
2 % 2	Reste 0
1 % 2	Reste 1
0 % 2	Reste 0
0 % 2	Reste 0
0 % 2	Reste 0
0 % 2	Reste 0
0 % 2	Reste 0
0 % 2	Reste 0
0 % 2	Reste 0

Maquette graphique pour afficher le résultat :



Lorsque le nombre à convertir est négatif le résultat affiché reste celui de la conversion en positif, pour y remédier il faudrait modifier le calcul en stockant lorsque le résultat de conversion dans un autre tableau, car en ce moment il remplace la valeur du tableau dans lequel sont stockés les résultats de calcul de la conversion positive ce qui empêche de récupérer ces résultats pour afficher le calcul complet à l'utilisateur. Une modification de l'interface de l'affichage du calcul est nécessaire pour afficher la conversion de binaire positif en négatifs.

6.1.2. Conversion de décimal à virgule en binaire

La conversion concernant les nombres à virgule ne pose aucun problème, elle fonctionne sans aucun souci, concernant ce qu'il y a avant la virgule, cela ne changera pas, elle sera convertie comme s'il s'agissait d'un nombre entier, pour ce qu'il y a après la virgule elle sera convertie séparément et ajoutée au résultat de la conversion de ce qu'il y a avant la virgule. Il n'est cependant pas possible d'afficher le calcul, pour cela il faudrait stocker chaque résultat dans un tableau pour les récupérer lorsque l'utilisateur veut afficher le calcul, l'interface d'affichage aura elle aussi besoin d'être modifiée.

6.1.3. Conversion de décimal en octal non signé

La conversion fonctionne correctement, l'affichage du calcul est possible, cependant comme pour la conversion de décimal à binaire concernant l'affichage du calcul il n'y a pas la table des puissances ni le résultat, cela doit aussi être ajouté afin d'aider l'utilisateur à se corriger et comprendre comment convertir pour ainsi, le permettre de s'améliorer si besoin. La conversion est donc uniquement en non signé ou positif, cependant si l'on veut convertir en négatif cela convertira qu'en positif et non négatif, pour cela il faudra modifier le code afin de permettre cela.

6.1.4. Décimal à virgule en octal

La conversion fonctionne pour la première fois, si le programme n'est pas relancé alors les suivantes seront presque toutes fausses, cette erreur n'a pas encore été trouvée jusqu'à ce jour (07.06.2018), en effet lors de la conversion si l'on clique directement sur le bouton « Convertir » alors le résultat sera faux, cependant pour une raison inconnue lors de la modification du nombre à convertir il s'avère que le programme peut convertir des fois faux et d'autres fois juste.

6.1.5. Binaire en décimal non signé

La conversion fonctionne correctement, le calcul ne peut cependant pas être affiché, comme pour les autres il faut stocker chaque résultat dans un tableau pour les récupérer lorsque l'utilisateur souhaite afficher les résultats de conversion avec un tableau des puissances montrant ce que vaut chaque bit suivant sa position. Pour une raison inconnue jusqu'au 07.06.2018, la conversion ne peut pas être effectuée pour un nombre binaire supérieur à « 111111111 », ce nombre est considéré comme état supérieur à 32 bits ce qui ne devrait pas être le cas.

6.1.6. Binaire en octal non signé

L'utilisateur peut convertir le nombre binaire sans problème, le calcul ne peut cependant pas être affiché, tout comme les autres il faut stocker chaque résultat dans un tableau afin de pouvoir les récupérer pour les afficher à l'utilisateur.

6.1.7. Binaire signé dans les autres bases

En effet il manque la conversion de binaire signé dans les différentes bases qu'est le décimal, octal et hexadécimal, pour cela il faut créer une autre méthode, le fonctionnement du code est fait de sorte à l'aide de switch et de méthode de pouvoir permettre de facilement changer de quel base l'utilisateur doit convertir et de plus facilement adapter le code suivent le choix de l'utilisateur fait. Pour la conversion en hexadécimal il serait possible d'utiliser une conversion comme s'il s'agissait d'une comparaison avec un dictionnaire, si le nombre en binaire est de « 1010 » alors le programme va chercher dans sa liste à quoi cela correspond pour le convertir, s'il n'y a pas assez de bits puisque qu'il faut convertir par groupe de 4 bits il remplace les manquants par des 0. Pour convertir en signé il faudrait convertir en binaire négatif avant de convertir en hexadécimal.

Pour convertir en octal il suffirait de grouper par groupe de 3 bits comme lors de la conversion de binaire à octal en non signé, cependant il pourrait avoir besoin d'utiliser une conversion comme s'il s'agissait d'un dictionnaire comparant ce qu'il faut convertir avec une liste afin d'éviter les erreurs possibles.

Pour ce qui est de la conversion de binaire négatif en décimal il serait possible d'effectuer la conversion inverse de décimal négatif à binaire.

6.1.8. Additions et soustractions (entiers, positifs et/ou négatifs)

Pour cette partie il était prévu que le programme fonctionne avec des « modes », cela dans le but d'éviter de surcharger le programme en chargeant la page de conversion, addition et soustraction sur différents onglets ou autre.

6.2. BILAN DE LA PLANIFICATION

Du retard a été pris le premier jour pour prendre connaissance et comprendre les critères d'évaluations, par la suite, du retard a été pris pour le rapport sur lequel trop de temps y a été consacré, cela étant dû à la partie analyse sur lequel j'ai eu de la difficulté à terminer.

La planification n'a pas pu être suivie à cause du retard pris lors des 2 premières semaines, le projet c'est avéré plus compliqué que ce qui était prévu, cela empêchant de suivre la planification et obligeant de s'adapter à la situation actuelle.

6.3. BILAN PERSONNEL

Le projet est très intéressant car, en effet il s'agit de faire un programme de conversion dont l'utilisation est destinée aux élèves ainsi qu'aux enseignants. Étant moi-même un élève il s'agit donc de faire le programme qui d'un certain point de vue doit non seulement aider les élèves et les enseignants que ce soit pour effectuer des corrections, comprendre

comment convertir, additionner et soustraire mais aussi un programme que j'aurais aimé avoir afin de m'entraîner à convertir et de pouvoir vérifier ou m'aider à comprendre en ELEOC.

Malgré tout le projet possède quelques points négatifs, il s'agit d'un programme qui doit pouvoir afficher les calculs, cela demande du temps pour mettre en place le code qui permet d'éviter des erreurs et de s'adapter à ce que veut faire l'utilisateur, tout en retenant tout ce qui est nécessaire pour afficher les calculs à l'utilisateur, il est donc possible de vite s'y perdre dans les conversions notamment.

Je remercie mon chef de projet M. Gruaz pour m'avoir aidé lors du projet lors des entretiens, je remercie aussi M. Da Silva Diogo pour m'avoir aidé concernant certains points.

7. DIVERS

7.1. JOURNAL DE TRAVAIL

Le journal de travail se trouve en annexe.

7.2. WEBOGRAPHIE

Inspiration pour l'interface de la soustraction :

http://www.groupeisf.net/automatismes/Numeration/Numeration_binaire/Ressources/Logique_combinatoire/html/01/28-ess0102006.htm

Inspiration pour le code de conversion avec virgule :

<https://www.developpez.net/forums/d1168230/c-cpp/c/debuter/conversion-virgule-fixe-virgule-flottante/>

Inspiration pour le code de conversion de binaire à décimal :

<https://www.culture-informatique.net/conversion-binaire-decimal-hexadecimal-main/>

Pour récupérer un caractère d'une chaîne de caractère :

[https://msdn.microsoft.com/fr-fr/library/aka44szs\(v=vs.110\).aspx](https://msdn.microsoft.com/fr-fr/library/aka44szs(v=vs.110).aspx)

Pour compter le nombre de caractère d'une chaîne de caractère :

[https://msdn.microsoft.com/fr-fr/library/aka44szs\(v=vs.110\).aspx](https://msdn.microsoft.com/fr-fr/library/aka44szs(v=vs.110).aspx)

Vérification avec regex :

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

Comment empêcher un utilisateur d'écrire dans une Combo Box en modifiant le style :

<https://stackoverflow.com/questions/85702/how-can-i-make-a-combobox-non-editable-in-net>

Comment afficher une nouvelle fenêtre :

<https://codes-sources.commentcamarche.net/forum/affich-298450-ouverture-form-avec-bouton>

Site utilisé pour vérifier les conversions non signées :

http://www.naa.fr/support/cnv_bin.html

Site utilisé pour vérifier les conversions signées :

http://www.binaryconvert.com/result_signed_int.html?decimal=045050

Comment convertir de binaire signé positif à négatif :

<http://villemin.gerard.free.fr/Wwwgymm/Numerati/BINAIRE/Negatif.htm>

8. ANNEXES

Planification initiale

Journal de travail