

Exercices C7 VBA Binaire infodauphine.com

1. **Fonction *dectobinbit(n,bits,ligne)*** qui a comme paramètres d'entrée un nombre n positif à convertir sur le nombre de **bits spécifié** et à placer sur cette *ligne*. La fonction retourne *vraie* si la conversion est possible sur ce nombre de bits.
2. **Programme *saisiBin*** qui va **saisir** un entier x , puis le nombre de bits nb et la ligne nl où l'on souhaite placer la conversion en base 2 de x . On se servira de *dectobinbit*. Si la conversion n'est pas possible, on affichera le message "Pas assez de bits" et on **ressaisira** le nombre de bits.
3. **Fonction *C2Dec(ligne, bits)*** qui retourne la valeur en base 10 du binaire en **complément à 2** sur cette *ligne* à partir de son nombre de *bits*.
4. Ecrire une **fonction booléenne *doubler*** qui aura comme paramètre d'entrée *ligne* représentant le numéro de ligne d'un binaire sur 8 bits et qui va placer sur la ligne en dessous la valeur binaire sur 8 bits qui aura **le double de sa valeur**. On ne fera pas de conversion en base 10, et on vérifiera que son double peut tenir sur 8 bits, auquel cas on retournera la valeur vrai. Sinon, on retourne faux et la ligne ne devra pas être remplie.
5. **Procédure *sommeBin(ligne,bits)*** qui va placer sur cette *ligne* **l'addition des binaires** présents sur les 2 lignes au dessus et qui ont ce nombre de bits. Si l'addition est impossible, on affichera "L'addition est trop grande".
6. **Fonction *compare(n1,n2)*** qui pour les 2 binaires positifs sur 8 bits situés aux ligne $n1$ et $n2$, retourne vrai si celui de $n1$ est **inférieur ou égal** à celui de $n2$. On ne fera pas de conversion.
7. **Fonction *divbin(ligne,n)*** qui retourne vrai si cet octet est **divisible par 2^n** (sans faire de conversion)

Bonus: conversion avec binaire stocké en valeur chaîne ou numérique

- b1) Ecrire la fonction *binText(n)* qui retourne la valeur en base 2 sous forme de chaîne du nombre positif n . n sera ici de type *Double* pour pouvoir dépasser 2 milliards (*Long* ne peut lui pas les dépasser).
- b2) Ecrire une fonction *decFromText(bin)* qui retourne la valeur en base 10 du nombre binaire représenté par la chaîne *bin*.
- b3) Créer un fonction *getBinNumber(n As Long) As Double* qui retourne la valeur en base 2 du nombre n . La valeur binaire sera retournée sous forme numérique (donc *getBinNumber(25)* devra retourner le nombre *11001*). On utilise *Double* pour pouvoir avoir un nombre dépassant 2 milliards et donc représenter en base 2 un binaire de plus de 10 bits.
- b4) Créer une fonction *getFromBinNumber(bin As Double) As Long* qui retourne la valeur en base 10 du binaire *bin*, qui est une variable numérique. Donc *getFromBinNumber(11001)* doit retourner la valeur 25.