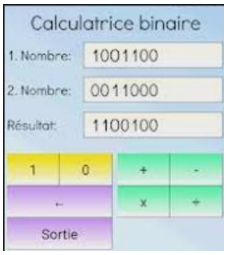
CALCULATRICE BINAIRE

TP v1.1 en lien avec le cours Codage binaire des nombres.

I. SUJET.

Le but de ce TP est d'écrire une application Calculatrice Binaire qui propose à l'aide d'un menu :

- Calcul de la valeur d'un nombre binaire.
- Ecriture binaire d'un entier selon les 2 encodages : par soustractions et par divisions.
- Réécriture sur N bits d'une écriture sur moins de bits.
- Addition binaire.
- Inversion bit à bit.
- Ecriture binaire de l'opposé.
- (Et plus 👍)



II. DATE DE RENDU DU TP.

26/03/2024 minuit.

III. IMPLEMENTATION.

A faire sur Replit.

1. Rappel : Que renvoient les instructions suivantes ? '1' + '1' → …… 3 * '0' → ……..
2. Ecrire une fonction calculer_binaire( ) qui renvoie la valeur d'un écriture binaire en entrée.
3. Ecrire une fonction encodage_par_soustraction( ) :
   a. Paramètres d'entrée : N le nombre de bits, n un nombre entier.
   b. Corps : algorithme par soustractions successives.
   c. Sortie : l'écriture binaire sur N bits de ce nombre entier.
4. Ecrire une fonction encodage_par_division( ) :
   a. Paramètres d'entrée : N le nombre de bits, n un nombre entier.
   b. Corps : algorithme par divisions successives.
   c. Sortie : l'écriture binaire sur N bits de ce nombre entier.
5. Ecrire une fonction écriture_sur_Nbits( ) : (sans utiliser la fonction zfill( ))
   a. Paramètres d'entrée : un nb binaire (type str), N le nombre de bits.
   b. Sortie : le nombre binaire (type str) réécrit sur N bits, avec des 0 à gauche si besoin.

6. Ecrire une fonction additionner_binaire( ) avec :
    a. 2 paramètres d'entrée : nbbinaire1, nbbinaire2, deux nombres binaires (type str) écrits sur N bits.
    b. Corps : **Utilisation d'un dictionnaire (cours p.13 exo ❷). Aucun calcul sur les entiers !**
    c. 2 sorties : la somme binaire (type str) + un indicateur pour dire s'il y a eu overflow (retenue en débordement à gauche de l'écriture binaire de la somme).
7. Ecrire une fonction inverser_bits( ) :
    a. Paramètre d'entrée : un nb binaire (type str) de longueur N bits.
    b. Sortie : la chaîne des caractères inversés bit à bit.
8. Ecrire une fonction opposé_binaire( ) :
    a. Paramètres d'entrée : un nombre entier positif ou négatif, N le nombre de bits.
    b. Corps de la fonction : utiliser l'une des 2 fonctions d'encodage binaire ainsi que les 3 fonctions précédentes pour obtenir l'écriture binaire sur N bits de l'opposé de l'entier en entrée.
    c. Sortie : l'écriture binaire (type str) sur N bits de l'opposé de l'entier en entrée.
9. Ecrire le programme principal :
    a. Demander sur combien de bits on va travailler durant tout le programme.

       Bien gérer les longueurs de bits → sécurité sur les nombres entrés aussi bien binaires qu'entiers !!
    b. un menu.
10. Bonus : rajouter des fonctionnalités supplémentaires et/ou un bel affichage.

IV. ATTENTION !

1. Un soin tout particulier est attendu dans l'ergonomie utilisateur : être capable de se mettre à la place d'un utilisateur novice qui utilisera votre programme :
    - L'utilisateur est-il bien guidé dès le début ?
    - L'utilisation du programme est-elle claire et simple ?
    - Les affichages sont-ils propres et aérés ?
2. Cette ergonomie va de pair avec un testing rigoureux du programme : interdiction de rendre un travail non testé, insuffisamment testé en se plaçant du point de vue d'un utilisateur novice.

   Faites tester votre programme par au moins une personne non informaticienne !
3. Pas d'utilisation de la déclaration global dans les fonctions : une fonction a des paramètres d'entrée et de sortie !

V. CRITERES D'EVALUATION DU TP.

Clarté du code (4 pts) :	
○ *Noms explicites des objets (variables, fonctions etc.) :*	/ 1 pt
○ *Entrées et sorties correctes des fonctions. Annotations de type dans l'entête des fonctions :*	/ 1 pt
○ *Organisation logique et aérée du code dont le programme principal :*	/ 1 pt
○ *Présence de commentaires pour identifier les différentes parties du code ou expliquer, préciser une partie du code :*	/ 1 pt
Qualité du code (16 pts) :	
○ *Programme répondant au cahier des charges (présence des différentes fonctionnalités, exécution totale ou partielle sans erreurs etc.) :*	/ 9 pts
○ *Simplicité de la mise en œuvre (présence de code inutile, boucles For indicées quand boucles For directes possibles, solutions algorithmiques compliquées, etc.) :*	/ 2 pts
○ *Ergonomie côté utilisateur (introduction, présence d'instructions utilisateur, propreté et organisation des affichages, clarté des textes affichés, fautes de français, , sécurisation des entrées, etc.) :*	/ 5 pts
Malus – Bonus :	
○ *Respect de la date finale de rendu :*	Malus -1 pt / h
○ *Régularité du travail :*	Malus -2 pts
○ *Aller plus loin que ce qui est demandé :*	Bonus +2 pts
Voir commentaires aussi sur Replit. *Note finale*	/ 20 pts