

Convertisseur de base

En Langage C

Code source:

<https://github.com/FaroukBel/Convertisseur-de-base>

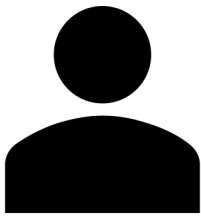
FAROUK BEL KHYATE

AMINE ASSALANE

Présentation Du Problème

- Conversion d'une base à une autre (2-10;5-6;7-4...)

Utilisateur



Nombre: 45
Base d'origine: 6
Base finale: 3

Convertisseur.exe

Nombre en base finale: 1002

Algorithmes

Convertisseur de base

Conversion de base:

Utilisateur



Nombre: AE6
Base d'origine: 16
Base finale: 3

`Fonction main()`

Convertisseur.exe

AE6 → Conversion en décimale → 2790
`Fonction aDeci()`

2790 → Conversion en base 3 → 11002000
`Fonction deDeci()`

Résultat en base 3: 1100200

Fonctions

Convertisseur de base

- L'utilisateur entre au clavier un nombre N, la base de ce nombre (base1) et la base (base2) qu'il veut l'en convertir.

```
// Program main
int main()
{
    char *nbr[100];
    int base1, base2;
    char res[100]; // Initiation du resultat de conversion du
                    // decimale au bases
    int len = strlen(nbr);
    // Saisie
    printf("Donner le nombre: ");
    scanf("%s", nbr);

    printf("Donner son base: ");
    scanf("%d", &base1);

    for (int i = len - 1; i >= 0; i--)
    {
        // Verification de la validité du nombre entre au clavier
        if (val(nbr[i]) >= base1)
        {
            printf("Nombre invalide");
            return -1;
        }
    }

    printf("Donner la base desire: ");
    scanf("%d", &base2);

    // Resultat
    printf("L equivalent de %s en base %d est "
           " %s\n", nbr, base2, deDeci(res, base1, base2, nbr));

    return 0;
}
```

La fonction main() permet la saisie des variables (Nombre, base1, base2) et les entrer comme paramètre dans la fonction deDeci() pour calculer le résultat et l'afficher.

- Puis le program convertir ce nombre en décimale (base 10) par une fonction (aDeci()).

```
// Fonction pour convertir un nombre d'une base au décimale
int aDeci(char *str, int base1)
{
    int len = strlen(str);
    int power = 1;
    int num = 0;
    int i;

    // Equivalent de decimale = str[len-1]*1 + str[len-2]*base +
    str[len-3]*(base^2) + ...
    for (i = len - 1; i >= 0; i--)
    {
        num += val(str[i]) * power;
        power = power * 10;
    }

    return num;
}
```

La fonction aDeci() permet la conversion d'un nombre d'une base quelconque à la base décimale par l'algorithme suivant:

$$\text{decimale} = \sum(\text{digit} \times \text{base}^{\text{index de digit}})$$

La vérification se fait par la fonction val():

```
int val(char c)
{
    if (c >= '0' && c <= '9') // Verification par code ASCII
        return (int)c - '0';    // Cast du char en int
    else
        return (int)c - 'A' + 10;
}
```

- Après le program convertir le décimale à la base désiré a l'aide de la fonction (deDeci()).

```
// Fonction pour convertir un décimale
// a une base (base2)
char* deDeci(char res[], int base1,int base2, char *inputNum)
{
    int i = 0; // Initiation d'index
    int deci = aDeci(inputNum, base1);
    char str[100];

    // Convertir un decimale a une base par
    // le deviser plusieurs fois en prenons le reste
    // comme resultat
    while (deci > 0)
    {
        res[i] = reVal(deci % base2);

        deci /= base2;

        i++;
    }
    res[i] = '\0';

    // Inverser le reste
    strev(res);

    return res;
}
```

La fonction deDeci() permet la conversion d'un nombre d'une base décimale à une base quelconque par l'algorithme suivant:

En divisons le nombre décimale par la base de destination jusqu'au le quotient soit 0 et on prend le reste en inverse

La vérification se fait par la fonction `reVal()`:

```
int reVal(char c)
{
    if (c >= '0' && c <= '9') // Verification par code ASCII
        return (int)c - '0';    // Cast du char en int
    else
        return (int)c - 'A' + 10;
}
```

L'inverse de la chaîne des restes se fait à l'aide de la fonction `strev()` suivante:

```
// Fonction pour inverser une chaîne de char
void strev(char *str)
{
    int len = strlen(str);
    int i;
    for (i = 0; i < len/2; i++)
    {
        char temp = str[i];
        str[i] = str[len-i-1];
        str[len-i-1] = temp;
    }
}
```

Conclusion

Convertisseur de base

Aventages:

- Ce program peut faire le calcule des complement un et deux d'un nombre de base 2 -> 36
- Convertir d'une base quelconque au decimale

Limitations:

- Ce program ne peut pas faire une conversion des nombres flotants
- Il faut toujours enter un nombre en majuscule (cas d'un nombre qui contient des lettres)