

## Programmation : TP 18

**Objectifs :** [C] Arithmétique en précision arbitraire.

On représente un nombre positif par la liste de ses chiffres (en base 10). Pour faciliter les manipulations, on utilise des listes *doublement chaînées* où chaque élément pointe sur l'élément suivant **et** le précédent. De plus, la liste est *tenue* par le premier élément et le dernier.

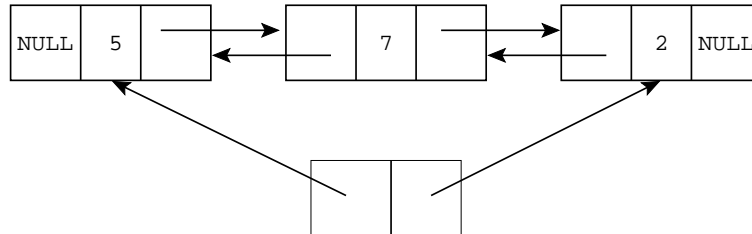


FIG. 1 – Représentation du nombre 572

1. Définir les types `ldc` qui permet de représenter une liste doublement chaînée et `npa` qui permet de représenter un nombre en précision arbitraire (NB : `ldc` et `npa` sont des pointeurs vers les structures). Un `npa` est *vide* si ses deux pointeurs sont nuls.

Définir les fonctions d'ajout à gauche et à droite d'un chiffre :

```
void ajoute_a_gauche(npa, int);
void ajoute_a_droite(npa, int);
```

2. Définir la fonction d'affichage d'un nombre.

```
void print(npa);
```

3. Définir la fonction de conversion d'une chaîne de caractères en `npa`.

```
npa atonpa(char*);
```

4. Définir la fonction d'addition de deux nombres `npa`.

```
npa add(npa, npa);
```

5. Définir la fonction de multiplication de deux nombres `npa`.

```
npa mult(npa, npa);
```