**Ne rien écrire ici**

**2)** Soit la suite **U** définie par: $\begin{cases} U_0 = 1 \\ U_n = 2*U_{n-1} + n \end{cases}$ (avec **n** un entier supérieur ou égal à 1)

**a)** **U** est une suite récurrente d'ordre :

☐ 1 ☐ 2 ☐ 5

**b)** Le $3^{ème}$ terme de la suite **U** ($U_2$) est égal à :

☐ 5 ☐ 8 ☐ 9

**c)** L'algorithme permettant de calculer $U_n$ (avec $n \geq 1$) est :

☐

```
0)Def FN terme (n : entier) : entier
1) t[0] ← 1
2) Pour i de 1 à n faire
      t[i] ← 2*t[i-1]+n
   Fin pour
3) terme ← t[n]
4) Fin terme
```

☐

```
0)Def FN terme(n : entier) : entier
1) Si n=0 alors terme ← 1
      Sinon
        terme←2*FN terme(n-1)+n
      Fin si
2) Fin terme
```

☐

```
0)Def FN terme(n : entier) : entier
1) Up ← 1
2) Pour i de 2 à n faire
         Up ← 2*Up+i
      Fin pour
3) terme ← Up
4) Fin terme
```

## Exercice 2 (3 points)

En arithmétique, un **auto-nombre** est un entier naturel **N** qui ne peut pas s'écrire sous la forme d'un nombre **M** ajouté à la somme des chiffres de **M**.

**Exemples :**

- Pour N = **21**,
  **N n'est pas un auto-nombre**, puisqu'il peut être généré à partir de la somme d'un nombre **M** égal à **15** et les chiffres qui le constituent (**1** et **5**) c'est-à-dire **21 = 15 + 1 + 5**.
- Pour N = **20**,
  **N est un auto-nombre** puisqu'il ne peut pas être généré à partir de la somme d'un nombre **M** et les chiffres qui le constituent.

**Travail demandé :**

Ecrire une **analyse** d'un module intitulé **Verif_auto_nombre**, permettant de **vérifier** si un entier naturel **N** strictement positif est un **auto-nombre**, sachant que N est déjà saisi dans l'analyse du programme principal.