

**[CYBER1][2023-2024] Rattrapage (2h00)**  
**Algorithmique 1**

NOM :

PRÉNOM :

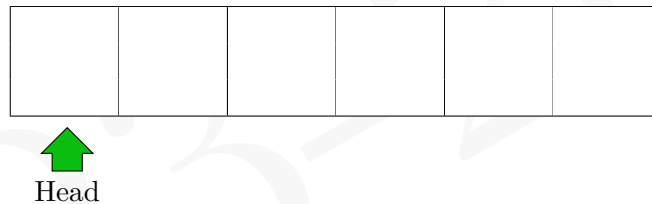
Vous devez respecter les consignes suivantes, sous peine de 0 :

- I) Lisez le sujet en entier avec attention
- II) Répondez sur le sujet
- III) Ne détachez pas les agrafes du sujet
- IV) Écrivez lisiblement vos réponses (si nécessaire en majuscules)
- V) Ne trichez pas

## 1 Questions (5 points)

- 1.1 (1 point) Écrivez l'état d'une file après avoir effectué ces opérations (la file est considérée comme initialement vide), puis, indiquez quel élément sortira de la file lors du prochain « dequeue », ainsi que celui qui sortira en dernier :**

enfiler 2, enfiler 4, défiler, enfiler 3, enfiler 1, défiler, défiler, enfiler 7, enfiler 5, défiler, enfiler 6

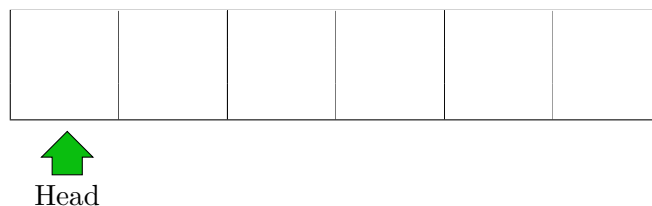


Prochain élément qui sortira :

Dernier élément qui sortira :

- 1.2 (1 point) Écrivez l'état d'une pile après avoir effectué ces opérations (la pile est considérée comme initialement vide), puis, indiquez quel élément sortira de la pile lors du prochain « pop », ainsi que celui qui sortira en dernier :**

empiler 2, empiler 4, dépiler, empiler 3, empiler 1, dépiler, dépiler, empiler 7, empiler 5, dépiler, empiler 6



Prochain élément qui sortira :

Dernier élément qui sortira :

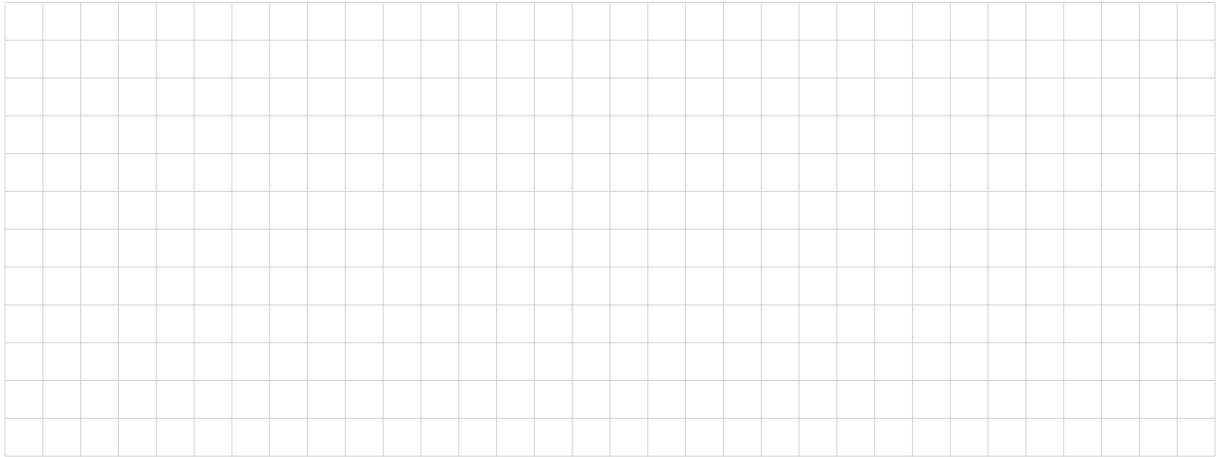
**1.3 (1,5 point)** En admettant que l'on dispose d'une pile vide et que les éléments « 1 2 3 4 5 6 » arrivent en entrée dans cet ordre exclusivement, décrivez les scénarios permettant d'obtenir les sorties suivantes :

*exemple : pour « A B C » en entrée, on peut obtenir « B C A » en sortie en faisant :*

*« push A », « push B », « pop », « push C », « pop », « pop »*

*On a bien inséré A, puis B, puis C, mais l'ordre de sortie est différent suivant les « pop »*

2, 1, 3, 6, 5, 4



4, 3, 5, 2, 1, 6



2, 4, 3, 5, 6, 1



1.4 (1,5 point) Exécutez l’algorithme suivant, et notez l’état d’avancement des variables (inscrivez l’état initial dans la ligne prévue à cet effet) :

*Vous exécuterez la fonction suivante avec comme paramètres  $a = 16435$  et  $b = 4242$  :*

```
int FunctionXYZ(int a, int b)
{
    int num = 0;

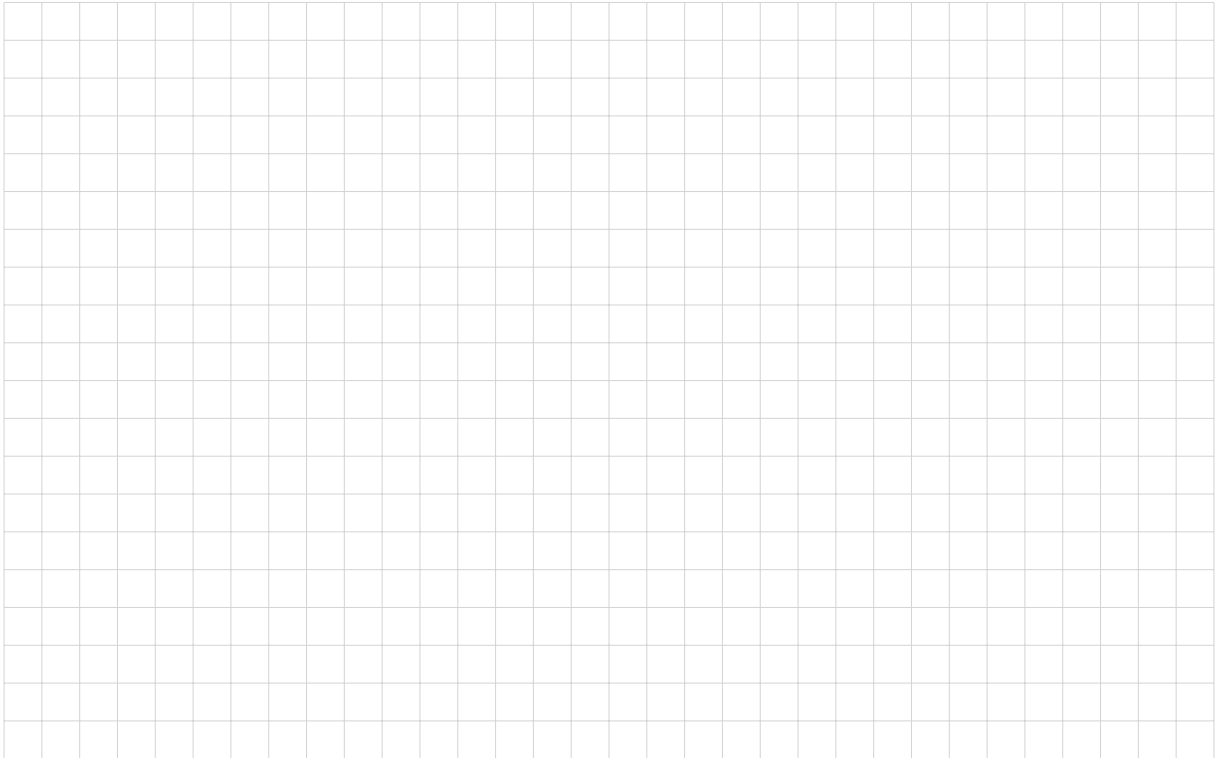
    if (a <= 0)
        return (-1);

    while ((a > 0) && (b > 0))
    {
        num += 1;
        a = a / 10;
        b = b / 10;
    }
    return (num);
}
```

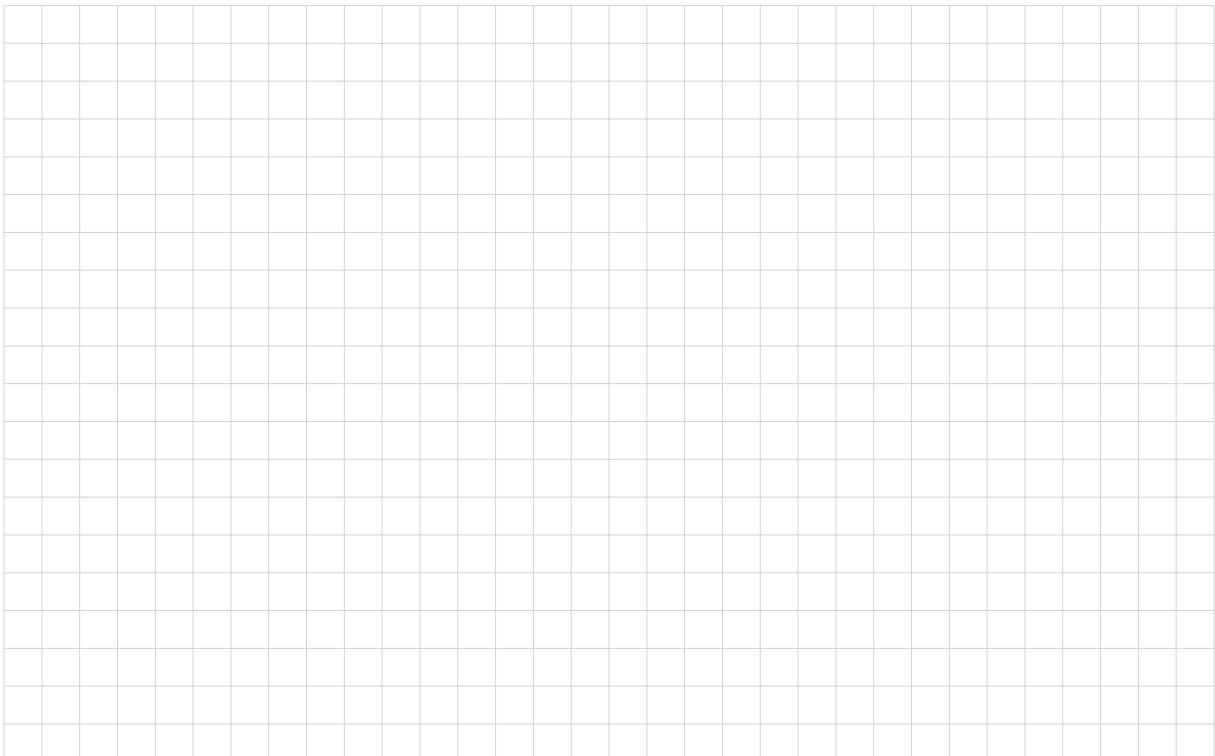
tour	a	b	num
<i>État initial</i>			

## 2 Algorithmes (15 points)

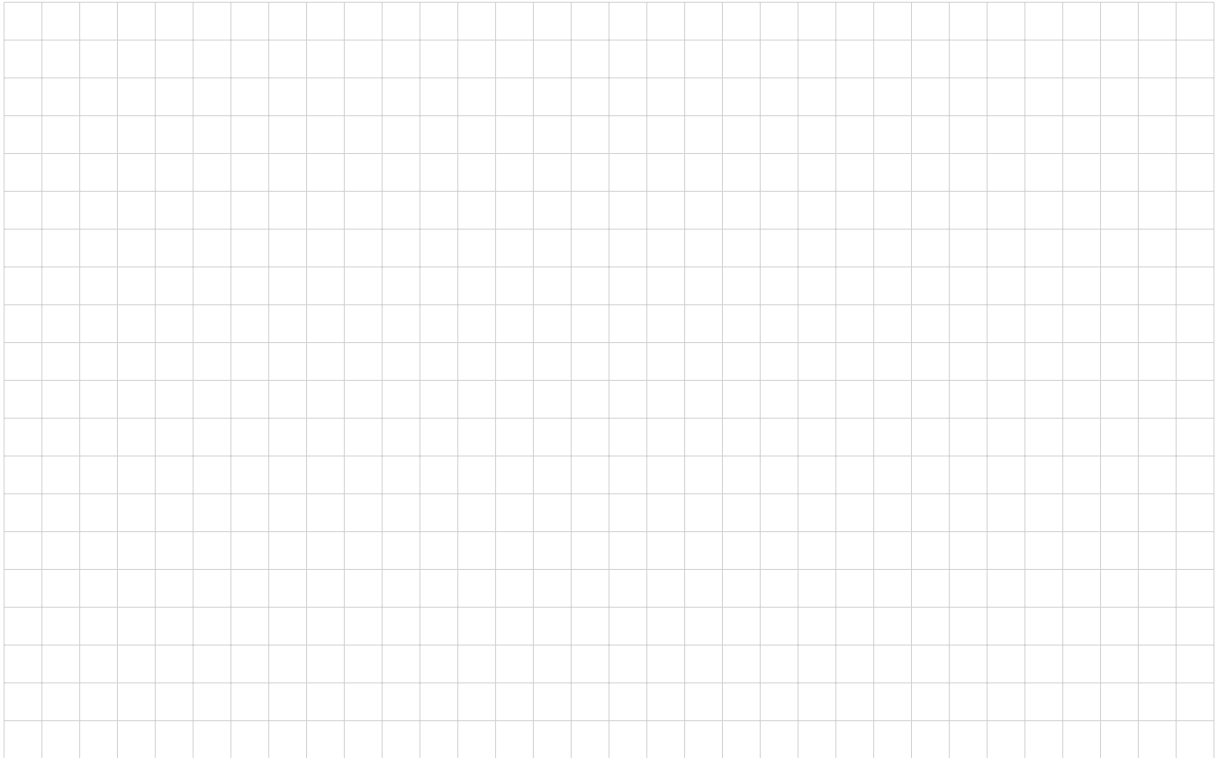
- 2.1 (2 points) Écrivez une fonction récursive « *ProductNInt* » calculant le produit des N premiers entiers



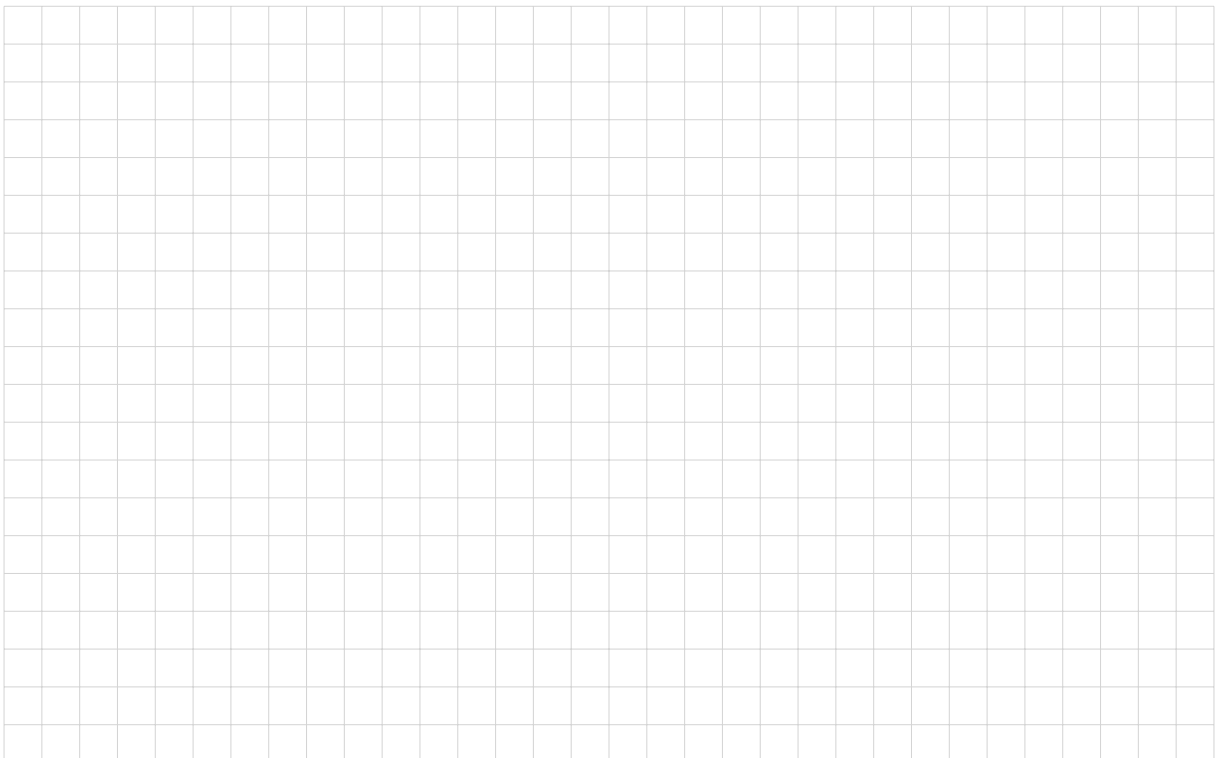
- 2.2 (2 points) Écrivez une fonction itérative « *my\_strlen* » calculant la taille d'une chaîne de caractères



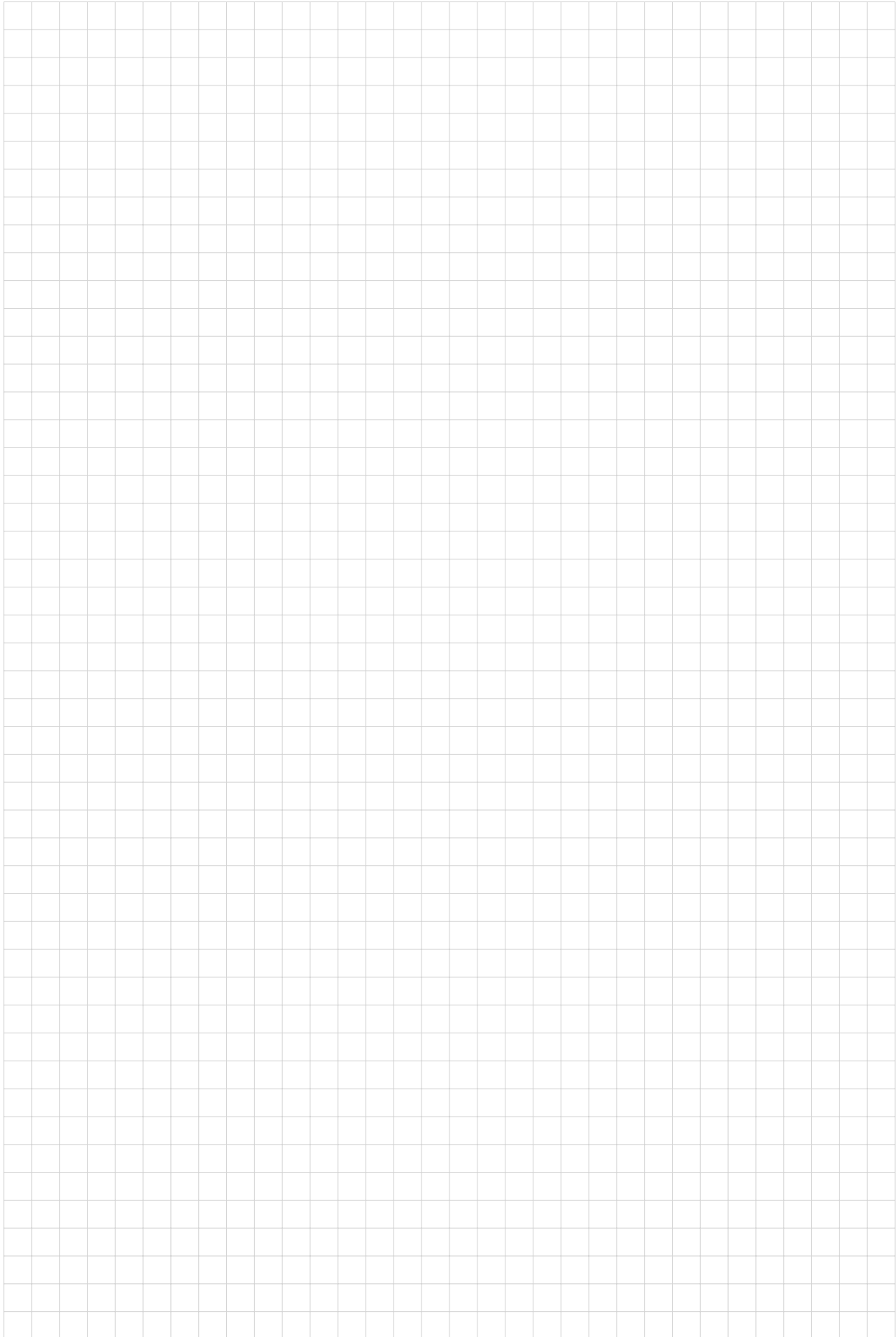
**2.3 (0,5 point) Écrivez une structure de données « *my\_stack* » pouvant servir de pile (à base de pointeurs ou de tableaux)**



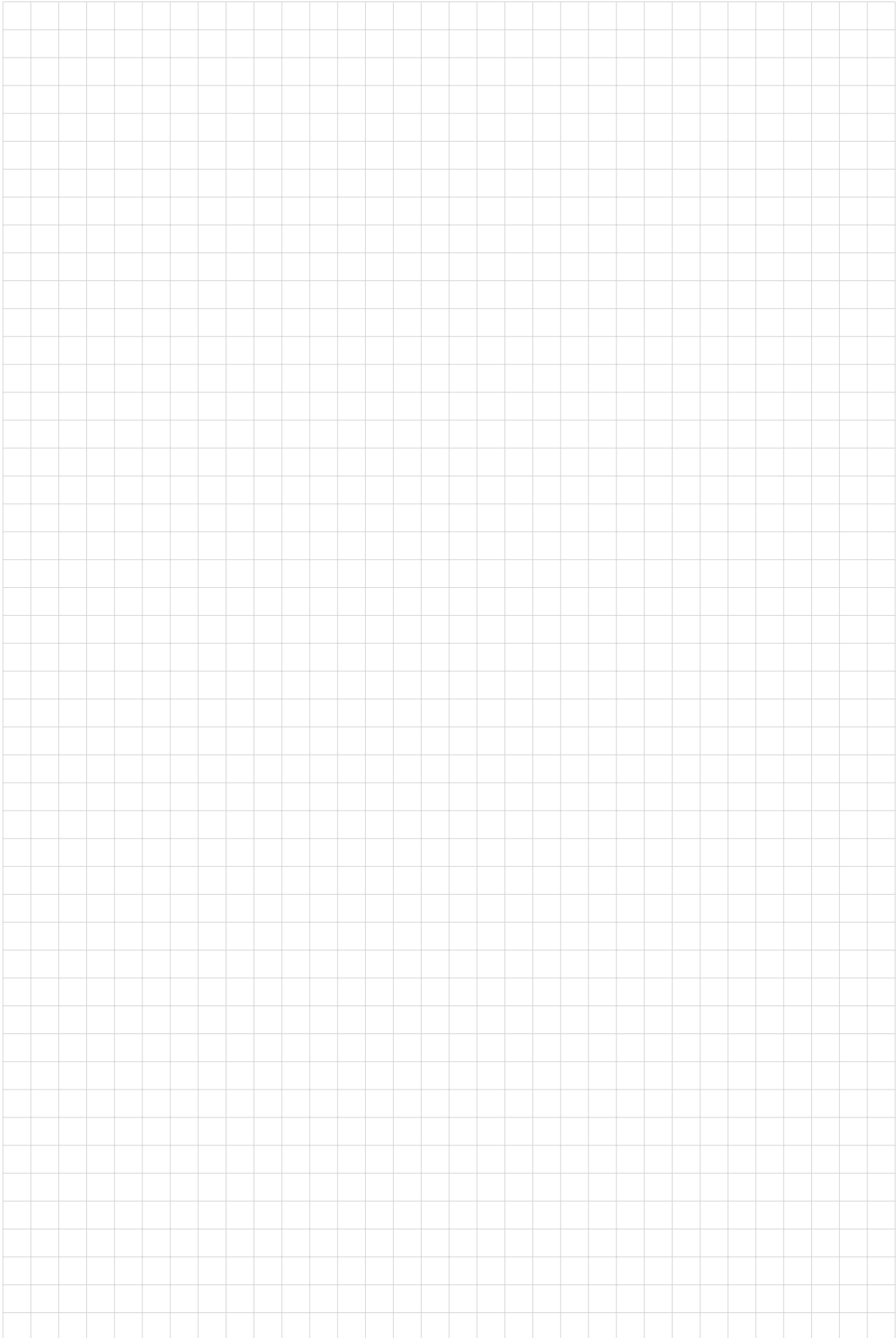
**2.4 (0,5 point) Écrivez une structure de données « *my\_queue* » pouvant servir de file (à base de pointeurs ou de tableaux)**



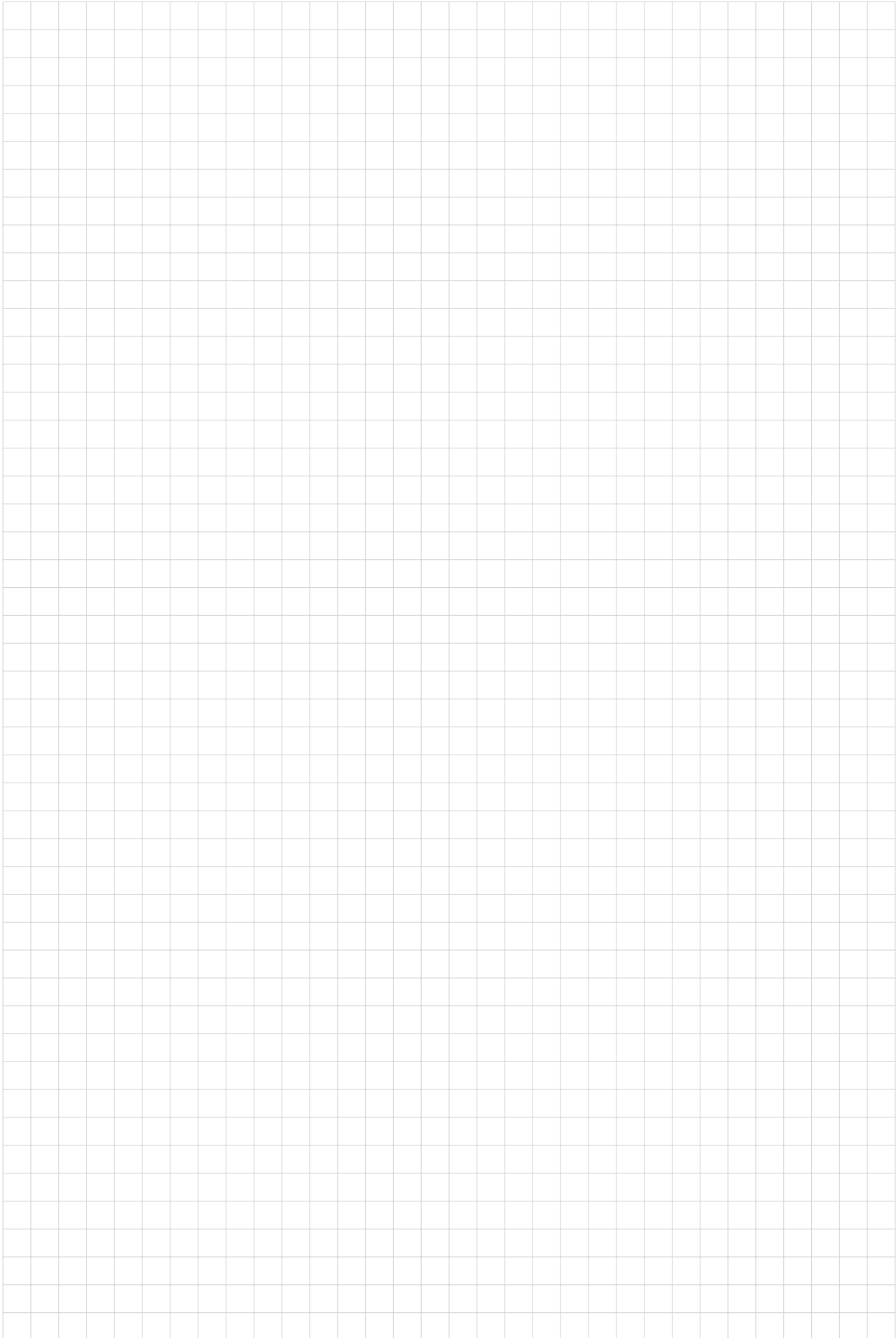
**2.5 (2,5 points)** Écrivez une fonction « *push* » pouvant servir à empiler un élément dans votre précédente structure « *my\_stack* »

A large grid of graph paper, consisting of 20 columns and 30 rows of small squares, intended for writing the code for the 'push' function.

**2.6 (2,5 points)** Écrivez une fonction « *pop* » pouvant servir à dépiler un élément dans votre précédente structure « *my\_stack* »

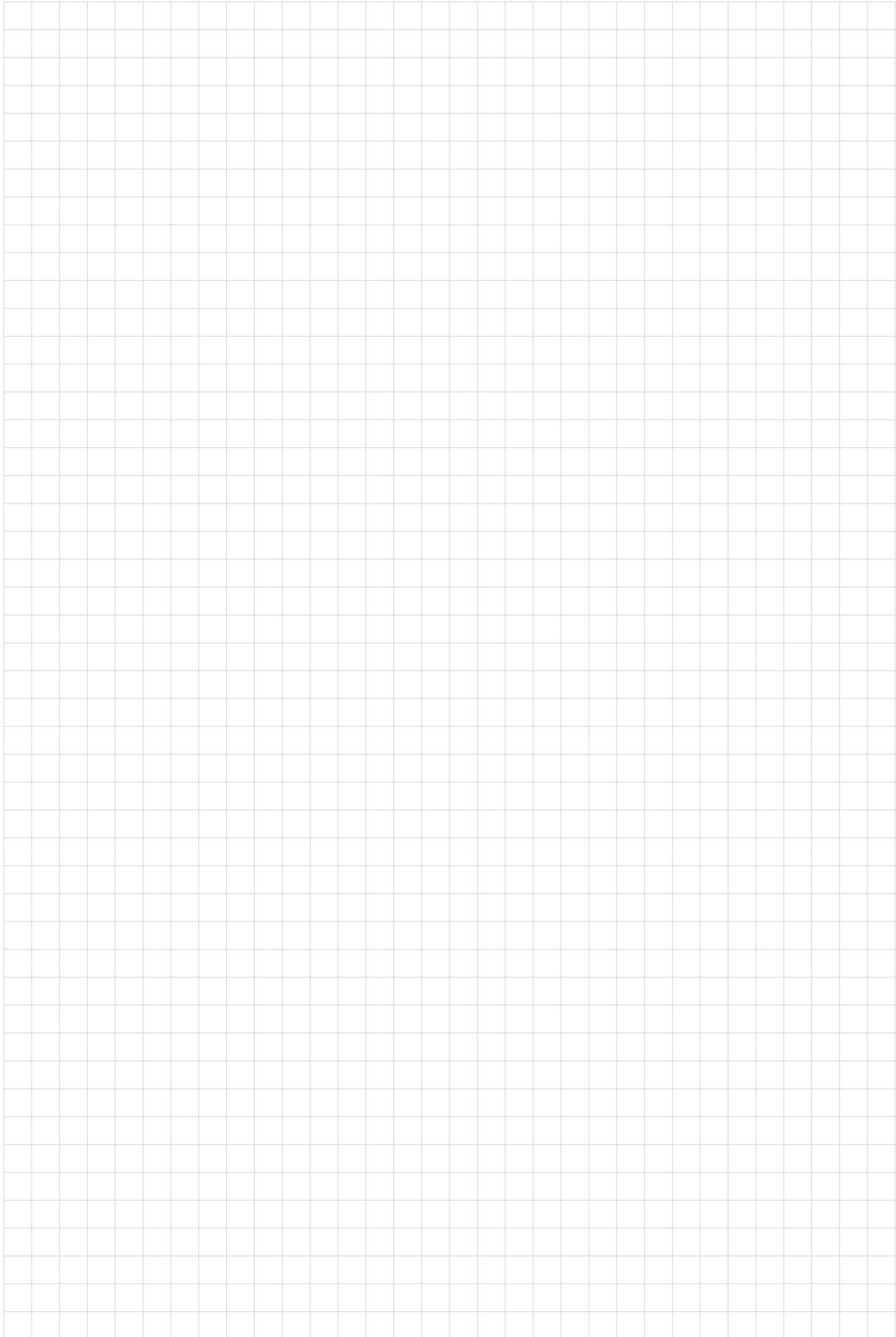
A large grid of graph paper, consisting of 20 columns and 40 rows of small squares, intended for writing the code for the 'pop' function.

**2.7 (2,5 points)** Écrivez une fonction « *enqueue* » pouvant servir à enfiler un élément dans votre précédente structure « *my\_queue* »

A large grid of graph paper, consisting of 20 columns and 30 rows of small squares, intended for writing the function.



**2.8 (2,5 points)** Écrivez une fonction « *dequeue* » pouvant servir à défiler un élément dans votre précédente structure « *my\_queue* »

A large grid of graph paper, consisting of 20 columns and 30 rows of small squares, intended for writing the implementation of the dequeue function.

# **RATTRAPAGE ALGORITHMIQUE 1**