

ⵜⴰⴷⵓⴷⴰ ⵜⴰ ⵎⴰⴳⵔⴰⵏⵜ  
ⵜⴰ ⵏ ⵉⵎⵓⵏⵏ ⵏ ⵉⵎⵓⵏⵏ  
ⵏ ⵉⵎⵓⵏⵏ ⵏ ⵉⵎⵓⵏⵏ



المملكة المغربية  
وزارة التربية الوطنية  
والتعليم الأولي والرياضة

المركز الجهوي لمهن التربية والتكوين لجهة الشرق  
ⵏ ⵉⵎⵓⵏⵏ ⵏ ⵉⵎⵓⵏⵏ ⵏ ⵉⵎⵓⵏⵏ ⵏ ⵉⵎⵓⵏⵏ

Projet personnel encadré dans le cadre du cycle de qualification des cadres de l'enseignement

Filière : Secondaire qualifiant Spécialité : Mathématiques

Sujet :

# Num Quik Génie des Nombres de l'Arithmétique à l'Informatique

Réalisé par les étudiants stagiaires : :

MAJRAD Khalid & LABZAR Ilham

Sous la direction du :

Pr. Younes DERFOUFI

Jury :

Pr. Younes DERFOUFI	-Encadrant
Pr. Mohammed SALHI	-Examineur

Date de la soutenance :

.... juin 2025

Année de formation :  
2024 - 2025

# TABLE DES MATIÈRES

Dédicace	5
Remerciements	6
Résumé de la recherche en français	7
Introduction	8
0.1 Contexte général du projet	8
0.2 Objectifs du projet	8
<b>1 Description de l'application</b>	<b>9</b>
1.1 Problématique	9
1.2 Caractéristiques techniques	9
1.3 Fonctionnalités de l'application	10
1.4 Présentation générale de l'application	10
1.5 Public cible	10
1.6 Pourquoi le nom <i>Num Quik</i> ?	11
<b>2 Outils et technologies utilisés</b>	<b>12</b>
2.1 Python	12
2.1.1 Sympy	13
2.2 Qt Designer	14
2.3 Visual Studio Code	15
2.4 TeXstudio et MiKTeX	16
<b>3 Théorie mathématique</b>	<b>17</b>
3.1 Nombres pairs et nombres impairs	17
3.1.1 Programme 1 : parité	17
3.2 Factorielle d'un entier	17
3.2.1 Programme 2 : Factorielle	18
3.3 Nombres parfaits	18
3.3.1 Programme 3 : parfaits	18
3.4 Nombres premiers	19
3.4.1 Programme 4 : premiers	19
3.5 Décomposition d'un entier relatif en produit des facteurs premiers	19
3.5.1 Programme 5 : Décomposition	20
3.6 Plus grand diviseur commun de deux entiers	20
3.6.1 Programme 6 : pgcd	20
3.7 Nombres premiers entre eux	21
3.7.1 Programme 7 : Premiers entre eux	21
3.8 Plus petit multiple commun de deux entiers relatifs non nuls	21
3.8.1 Programme 8 : ppcm	22
<b>4 Application sur théorèmes des nombres</b>	<b>23</b>
4.1 Motivation et Introduction	23
4.2 Présentation de l'interface graphique	23
4.2.1 Introduction à PyQt5	23
4.2.2 Fonctionnalités principales de PyQt5	24
4.2.3 Architecture de PyQt5	24
4.2.4 Pourquoi choisir PyQt5?	24
4.2.5 Limites de PyQt5	24
4.2.6 Conclusion	24
4.3 Description de l'interface Graphique	25
4.3.1 Bouton 1 : Vérifier la parité d'un nombre.	26
4.3.2 Bouton 2 : Vérifier si un nombre est premier	27
4.3.3 Bouton 3 : Décomposition d'un nombre en facteurs premiers	28
4.3.4 Bouton 4 : Vérifier si un nombre est parfait	29

4.3.5	Bouton 5 : Calcul du PGCD . . . . .	30
4.3.6	Bouton 6 : Calcul du PPCM . . . . .	31
4.3.7	Bouton 7 : Calcul du factoriel d'un nombre . . . . .	32
4.3.8	Bouton 8 : Vérifier si deux nombres sont premiers entre eux . . . . .	33
4.3.9	Bouton 9 : Fermer l'application . . . . .	33

<b>Conclusion</b>		<b>34</b>
-------------------	--	-----------

# TABLE DES FIGURES

2.1	PytHon. . . . .	12
2.2	Sympy. . . . .	13
2.3	Qt Designer. . . . .	14
2.4	Visual Studio Code. . . . .	15
2.5	TeXstudio avec MiKTeX. . . . .	16
4.1	L'interface Principale de l'application. . . . .	25
4.2	Interface de vérification de la parité d'un nombre. . . . .	26
4.3	Interface de test de primalité d'un nombre. . . . .	27
4.4	Interface de décomposition en facteurs premiers. . . . .	28
4.5	Interface de vérification des nombres parfaits. . . . .	29
4.6	Interface de calcul du PGCD de deux nombres. . . . .	30
4.7	Interface de calcul du PPCM de deux nombres. . . . .	31
4.8	Interface de calcul de la factorielle d'un nombre. . . . .	32
4.9	Interface de test de coprimauté de deux nombres. . . . .	33

Je remercie Dieu Tout-Puissant, qui ma donné force et patience, et qui a été une lumière qui a guidé mes pas sur ce chemin.

À mes chers parents, qui ont été mon soutien et mes encouragements constants, et qui m'ont inculqué les valeurs du travail acharné et de la persévérance. Que Dieu les récompense en mon nom avec la meilleure récompense.

À mes amis fidèles, qui ont partagé mes défis et mes succès, et qui ont été une source inépuisable d'inspiration et de motivation.

À tous ceux qui ont contribué à l'achèvement de ce travail, je vous dédie ce projet avec beaucoup de gratitude et d'appréciation.

# REMERCIEMENTS

Au nom de Dieu, le plus gracieux, le plus miséricordieux,

Je voudrais exprimer mes sincères remerciements et ma reconnaissance au distingué professeur **Younes DERFOUFI**, qui a été un excellent superviseur et guide tout au long de la réalisation de ce projet. Son soutien et ses conseils ont eu un impact significatif sur le développement du travail et la réalisation de ses objectifs.

Je voudrais également exprimer ma profonde gratitude au **Centre Régional des Métiers de l'Éducation et de la Formation de la Région de l'Oriental (Al Massira)** pour l'environnement éducatif stimulant et favorable qu'il offre, et pour les efforts qu'il déploie pour développer les compétences éducatives et professionnelles.

Je ne peux manquer de remercier le personnel administratif respecté, qui a joué un rôle clé en créant les conditions appropriées pour l'achèvement de ce projet, et tous les professeurs distingués qui ont enrichi ma carrière universitaire avec leurs précieuses connaissances et leurs conseils constructifs.

Enfin, j'exprime ma reconnaissance et ma gratitude à mes chers amis, qui ont toujours été une source de soutien et de motivation, et qui ont partagé ce voyage avec moi avec tant de dévouement et de sincérité.

À vous tous, je présente mes sincères remerciements et mon respect, et vous souhaite beaucoup de succès dans vos carrières académiques et professionnelles.

**Avec mes sincères remerciements et ma reconnaissance,**

**MAJRAD KHALID**

# RÉSUMÉ DE LA RECHERCHE EN FRANÇAIS

Le projet **Num Quik** est une application éducative développée en **Python** à l'aide de la bibliothèque **PyQt5**, qui vise à faciliter l'apprentissage et l'exploration des *théorèmes des nombres* à travers une interface graphique interactive, intuitive et accessible.

Cette application s'adresse principalement aux **élèves du secondaire** ainsi qu'aux **enseignants stagiaires en mathématiques**. Elle propose une série de fonctionnalités permettant de manipuler les concepts fondamentaux de l'arithmétique, notamment :

- la parité d'un nombre (pair/impair),
- la vérification de la primalité,
- la décomposition en facteurs premiers,
- la reconnaissance des nombres parfaits,
- le calcul du PGCD et du PPCM,
- le calcul de la factorielle,
- la vérification de la coprimauté entre deux nombres.

L'objectif principal du projet est de **rendre les mathématiques plus concrètes et motivantes** en automatisant les calculs tout en affichant des résultats lisibles et pédagogiques. Num Quik permet aux utilisateurs d'approfondir leur compréhension des notions arithmétiques sans recourir à des calculs manuels fastidieux.

En intégrant les bibliothèques **SymPy** pour le calcul symbolique et **Qt Designer** pour la conception de l'interface, ce projet allie efficacité, rapidité d'exécution et convivialité. Il constitue un excellent support pour illustrer les notions mathématiques en classe ou en auto-apprentissage, tout en modernisant l'enseignement des mathématiques à l'aide des technologies numériques.

## 0.1 Contexte général du projet

Num Quik est né d'un objectif simple : rendre les mathématiques plus accessibles et ludiques grâce au numérique. Dans le cadre de la formation des professeurs stagiaires, il est devenu essentiel d'intégrer des outils interactifs pour enrichir les cours et aider les élèves à mieux comprendre.

Avec Num Quik, les notions comme la parité, les nombres premiers, la décomposition, le PGCD ou encore le PPCM deviennent plus faciles à explorer. L'application aide les enseignants à capter l'attention des élèves et à rendre les notions abstraites plus concrètes, tout en rendant l'apprentissage plus amusant.

## 0.2 Objectifs du projet

Le projet **Num Quik** vise à atteindre plusieurs objectifs pédagogiques et techniques :

- **Faciliter la compréhension des notions arithmétiques** de base telles que la parité, la primalité, le PGCD, le PPCM, les factoriels, etc., à travers une interface intuitive.
- **Fournir un outil pédagogique interactif** destiné aux élèves et aux enseignants pour expérimenter et vérifier des propriétés numériques de manière concrète.
- **Renforcer l'apprentissage autonome** en permettant aux utilisateurs d'explorer les propriétés des nombres sans recourir à des calculs manuels fastidieux.
- **Offrir un support visuel et pratique** aux enseignants stagiaires pour illustrer les notions mathématiques durant les cours.

# 1 DESCRIPTION DE L'APPLICATION

**Num Quik** est une application éducative conçue pour faciliter la compréhension et la manipulation des concepts arithmétiques fondamentaux, en particulier pour les élèves du secondaire et les enseignants stagiaires en mathématiques.

L'objectif principal de l'application est d'offrir un outil interactif et visuel permettant de réaliser rapidement des calculs liés aux nombres entiers, tout en renforçant l'aspect pédagogique à travers une interface intuitive et claire.

## 1.1 Problématique

De nombreux élèves rencontrent des difficultés à comprendre les notions numériques fondamentales telles que la primalité, la parité, le PGCD, le PPCM ou encore la décomposition en facteurs premiers, surtout lorsque ces notions sont abordées de manière théorique et sans support visuel. Du côté des enseignants en formation, le besoin d'outils pédagogiques interactifs et accessibles pour illustrer ces concepts est également important.

C'est dans ce contexte qu'a été conçu Num Quik : une application simple et rapide, développée avec une interface graphique intuitive en Python, permettant d'explorer et de manipuler facilement les propriétés des nombres. Elle s'adresse aussi bien aux élèves qu'aux enseignants, en leur offrant un ensemble de fonctionnalités pour tester, calculer et visualiser les caractéristiques des entiers : parité, primalité, factorisation, nombres parfaits, PGCD, PPCM, factorielle, et coprimauté.

L'objectif est clair : rendre l'apprentissage des mathématiques plus concret, plus visuel, et plus motivant.

## 1.2 Caractéristiques techniques

L'application est développée en Python, avec une interface graphique construite via Qt Designer et intégrée grâce à la bibliothèque PyQt5. Elle est compatible avec les environnements Windows et Linux, et peut être exécutée localement sans connexion internet.

## 1.3 Fonctionnalités de l'application

L'application **Num Quik** propose une interface graphique simple et efficace permettant d'explorer diverses propriétés des nombres entiers. Les fonctionnalités disponibles sont les suivantes :

- **Parité** : Vérifie si un nombre est pair ou impair.
- **Primalité** : Détermine si un nombre est premier.
- **Décomposition** : Affiche la décomposition en facteurs premiers d'un entier.
- **Nombre parfait** : Indique si un nombre est parfait, c'est-à-dire égal à la somme de ses diviseurs propres.
- **PGCD** : Calcule le Plus Grand Commun Diviseur entre deux entiers.
- **PPCM** : Calcule le Plus Petit Commun Multiple entre deux entiers.
- **Factorielle** : Calcule la factorielle d'un nombre donné.
- **Nombres premiers entre eux** : Vérifie si deux entiers sont premiers entre eux (PGCD = 1).
- **Interface intuitive** : Interface graphique colorée et facile à utiliser, adaptée aux élèves comme aux enseignants.

## 1.4 Présentation générale de l'application

L'application **Num Quik** est développée en Python, en s'appuyant principalement sur la bibliothèque PyQt5 pour la création d'une interface graphique simple, claire et intuitive. Conçue dans un but pédagogique, elle permet aux utilisateurs élèves comme enseignants d'explorer facilement les propriétés fondamentales des nombres entiers.

L'application propose notamment les fonctionnalités suivantes :

- Saisie rapide de nombres entiers à analyser.
- Exécution automatique d'opérations arithmétiques de base : parité, primalité, PGCD, PPCM, factorielle, décomposition, etc.
- Affichage instantané et lisible des résultats, facilitant leur interprétation.
- Navigation fluide et accessible, même pour les débutants en mathématiques ou en informatique.

## 1.5 Public cible

- **Professeurs stagiaires** : un outil pratique pour illustrer les notions d'arithmétique de manière interactive, et enrichir leurs préparations de cours avec des démonstrations claires et visuelles.
- **Élèves du lycée** : une application d'appui pour explorer les propriétés des nombres (telles que la primalité, la parité, le PGCD, etc.) à travers des interactions simples, visuelles et motivantes.

## 1.6 Pourquoi le nom *Num Quik* ?

Le nom *Num Quik* est une combinaison de deux éléments clés qui reflètent la philosophie de l'application :

- « **Num** » fait référence au mot *numérique* ou *nombre*, en lien direct avec le domaine de l'arithmétique et des mathématiques discrètes abordé par l'application.
- « **Quik** » (variante stylisée de « Quick ») souligne la rapidité d'exécution et la simplicité d'utilisation. Il s'agit d'un outil pensé pour produire des résultats instantanés, tout en maintenant une clarté pédagogique.

Ce nom court et percutant évoque donc à la fois la nature du contenu traité (les nombres) et l'objectif principal de l'application : proposer une solution rapide, intuitive et efficace pour l'apprentissage et l'exploration de notions mathématiques fondamentales.

## 2 OUTILS ET TECHNOLOGIES UTILISÉS

Le développement de l'application **Num Quik** a nécessité l'utilisation de plusieurs outils et environnements logiciels adaptés à différents aspects du projet : conception de l'interface graphique, programmation, rédaction du rapport et organisation générale. Cette section présente les outils sélectionnés, leur rôle dans le projet ainsi que les raisons de leur choix.

### 2.1 Python

Python est un langage de programmation largement de haut niveau interprété, polyvalent, orienté objet et multiplateforme utilisé pour un large éventail de tâches, notamment le développement web, l'analyse de données, le calcul scientifique, l'intelligence artificielle, l'apprentissage automatique. Il y a été créé par Guido van Rossum et publié pour la première fois en 1991. Python pour sa simplicité, sa lisibilité et sa facilité d'utilisation, possède une vaste bibliothèque standard et une grande communauté. Python possède de nombreuses bibliothèques et frameworks spécialement conçues pour l'apprentissage automatique.



FIGURE 2.1 – PytHon.

Dans ce langage en utilisant la bibliothèque suivante :

## 2.1.1 Sympy

Sympy est une bibliothèque Python dédiée au calcul symbolique. Contrairement à la bibliothèque `math`, qui se concentre sur les nombres réels et entiers simples, `sympy` peut manipuler des expressions algébriques, des équations, des entiers, et même faire de l'algèbre formelle.

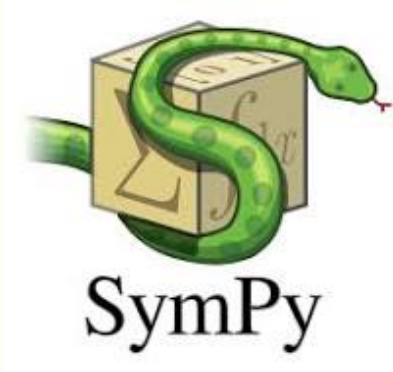


FIGURE 2.2 – Sympy.

### Utilisation dans Num Quik

- `isprime(n)` : pour tester si un nombre est premier.
- `gcd(a, b)` : pour calculer le PGCD (plus grand commun diviseur).
- `lcm(a, b)` : pour calculer le PPCM.
- `factorial(n)` : pour calculer la factorielle.
- `factorint(n)` : pour faire la décomposition en facteurs premiers.
- `Integer(n)` : conversion vers un entier symbolique pour plus de précision.

### Pourquoi Sympy :

- Le choix de la bibliothèque `sympy` pour ce projet repose sur plusieurs raisons techniques et pédagogiques essentielles :
- **Calcul symbolique précis** : Contrairement à des bibliothèques comme `math` ou `numpy`, `sympy` effectue des calculs exacts (par exemple, les fractions ne sont pas arrondies).
  - **Richesse fonctionnelle** : Elle offre un grand nombre de fonctions mathématiques avancées prêtes à l'emploi : `gcd`, `lcm`, `factorial`, `isprime`, `factorint`, etc.
  - **Facilité d'utilisation** : Sa syntaxe est claire et intuitive, ce qui permet une intégration rapide et sans complexité excessive, surtout dans des projets éducatifs comme *Num Quik*.
  - **Documentation complète** : Une communauté active et une documentation bien structurée facilitent l'apprentissage et la résolution de problèmes.
  - **Compatibilité avec PyQt5** : Elle s'intègre facilement dans des projets graphiques utilisant PyQt5, sans conflit de dépendances.

## 2.2 Qt Designer

Qt Designer est un outil visuel permettant de concevoir des interfaces graphiques (GUI) de manière intuitive et rapide. Il fait partie de la suite de développement Qt et permet de créer des interfaces sans écrire manuellement du code, en utilisant simplement le glisser-déposer d'éléments graphiques (widgets). Cet outil génère des fichiers avec l'extension `.ui`, qui peuvent ensuite être convertis en code Python à l'aide de l'outil `pyuic5`.



FIGURE 2.3 – Qt Designer.

### Utilisation dans Num Quik

L'application **Num Quik** comporte plusieurs fenêtres liées aux différentes fonctionnalités (parité, primalité, PGCD, etc.), chacune conçue dans Qt Designer. Ces interfaces ont été exportées sous forme de fichiers `.ui`, puis converties en modules Python pour être intégrées dans l'application. Cela a permis :

- Une séparation claire entre la logique de traitement (code) et l'interface graphique.
- Un gain de temps considérable dans la conception des interfaces.
- Une meilleure lisibilité et maintenance du projet.

### Pourquoi Qt Designer :

Le choix de Qt Designer dans ce projet repose sur ses nombreux avantages :

- **Interface intuitive** : Il permet de concevoir visuellement les interfaces sans avoir à coder chaque composant à la main.
- **Gain de productivité** : Grâce à son système de glisser-déposer, le temps de développement des interfaces est réduit.
- **Compatibilité avec PyQt5** : Les fichiers générés sont directement compatibles avec PyQt5, ce qui facilite leur intégration dans des projets Python.
- **Séparation du code et de l'interface** : Cela rend le projet plus modulaire, propre, et facile à maintenir.
- **Adopté dans le monde professionnel** : C'est un outil largement utilisé dans l'industrie pour les projets d'interfaces modernes et complexes.

## 2.3 Visual Studio Code

Visual Studio Code, souvent abrégé VS Code, est un éditeur de code source moderne, léger et multiplateforme développé par Microsoft. Il prend en charge une grande variété de langages de programmation, dont Python, grâce à des extensions dédiées. Son interface épurée, sa rapidité d'exécution et sa flexibilité en font l'un des environnements de développement les plus utilisés aujourd'hui.

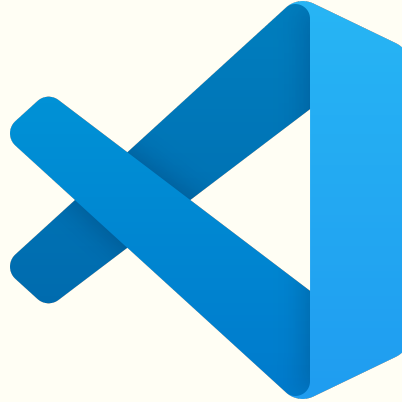


FIGURE 2.4 – Visual Studio Code.

### Utilisation dans Num Quik :

VS Code a été utilisé comme environnement de développement principal pour écrire, organiser et tester le code Python du projet. Les extensions installées (telles que Python, Pylance, Qt Tools) ont permis :

- L'autocomplétion du code et la détection d'erreurs.
- Le débogage efficace en temps réel.
- L'intégration fluide avec les fichiers générés par Qt Designer.

### Pourquoi Visual Studio Code :

- **Interface intuitive et moderne.**
- **Support étendu pour Python** via des extensions puissantes.
- **Gratuit, open source et multi-plateforme.**
- **Communauté large et documentation riche.**
- **Adapté aux petits comme aux grands projets.**

## 2.4 TeXstudio et MiKTeX

**TeXstudio** est un éditeur LaTeX avancé qui offre une interface conviviale pour écrire, éditer et compiler des documents scientifiques et techniques. Associé à la distribution **MiKTeX**, il permet de générer des documents PDF avec une mise en page professionnelle et précise.

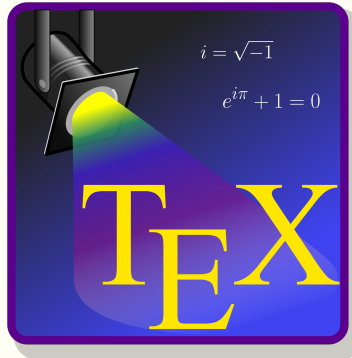


FIGURE 2.5 – TeXstudio avec MiKTeX.

### Utilisation dans Num Quik

Le rapport de projet de **Num Quik** a été entièrement rédigé à l'aide de TeXstudio. La distribution MiKTeX a permis la compilation correcte des fichiers `.tex` en `.pdf`. Cette combinaison a permis :

- Une structuration claire et hiérarchique du document.
- L'ajout automatique de table des matières, figures, sections et références.
- Une gestion efficace de la mise en page.

### Pourquoi TeXstudio et MiKTeX :

- **Qualité typographique supérieure** par rapport aux traitements de texte classiques.
- **Outils puissants** pour la gestion des références, des équations et des figures.
- **Automatisation** de la mise en page, des titres et des numérotations.
- **Gratuit et open source.**
- **Large adoption académique**, surtout dans les rapports techniques et scientifiques.

## 3.1 Nombres pairs et nombres impairs

- Tout nombre entier naturel multiple de 2 est appelé nombre **pair**.
- Tout nombre entier naturel qui n'est pas pair est dit **impair**.
- Les nombres pairs sont les nombres qui s'écrivent sous la forme  $2k$  où  $k$  est un nombre entier relatif.
- Les nombres impairs sont les nombres qui s'écrivent sous la forme  $2k + 1$  où  $k$  est un nombre entier naturel, ou sous la forme  $2k - 1$  où  $k$  est un nombre entier relatif non nul.

### Exemples :

- Soit  $a$  un nombre entier naturel non nul et différent de 1.

On pose  $A = 2a - 3$  et  $B = 4a + 2$ .

Pour  $A$  :

$$A = 2a - 2 - 1 = 2(a - 1) - 1.$$

En posant  $k = a - 1$ , on a  $A = 2k - 1$  où  $k$  est un nombre entier relatif non nul. Donc  $A$  est impair.

Pour  $B$  :

$$B = 4a + 2 = 2(2a + 1).$$

En posant  $k' = 2a + 1$ , on a  $B = 2k'$  où  $k'$  est un entier relatif. Donc  $B$  est pair.

### 3.1.1 Programme 1 : parite

```

1  from sympy import Integer
2
3
4  def test_parite(n):
5  nombre = Integer(n)
6  if nombre % 2 == 0:
7  print(f"{n} est un nombre pair.")
8  else:
9  print(f"{n} est un nombre impair.")
10
11
12 # Exemple d'utilisation :
13 n = int(input("Entrez un nombre : "))
14
15 test_parite(n)

```

## 3.2 Factorielle d'un entier

Soit  $n$  un entier naturel ( $n \geq 0$ ). La **factorielle** de  $n$ , notée  $n!$ , est définie par :

$$n! = \begin{cases} 1 & \text{si } n = 0, \\ n \times (n-1) \times (n-2) \times \cdots \times 1 & \text{si } n \geq 1. \end{cases}$$

### Exemples :

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120.$$

### 3.2.1 Programme 2 : Factorielle

```
1 from sympy import factorial
2
3
4 def calculer_factorielle(n):
5     if n < 0:
6         print("La factorielle n'est définie que pour les entiers naturels (n ? 0).")
7     else:
8         print(f"{n}! = {factorial(n)}")
9
10
11 # Exemple d'utilisation
12 n = int(input("Entrez un entier : "))
13 calculer_factorielle(n)
```

## 3.3 Nombres parfaits

Un entier naturel  $n \geq 1$  est appelé un **nombre parfait** si la somme de ses diviseurs stricts (c'est-à-dire, ses diviseurs positifs différents de  $n$ ) est égale à  $n$ .

Autrement dit :

$$\sum_{\substack{d|n \\ d < n}} d = n.$$

### Exemples :

6 est un nombre parfait car ses diviseurs stricts sont 1, 2, 3, et  $1 + 2 + 3 = 6$ .

### Autre exemple :

28 est un nombre parfait car ses diviseurs stricts sont 1, 2, 4, 7, 14, et  $1 + 2 + 4 + 7 + 14 = 28$ .

### 3.3.1 Programme 3 : parfaits

```
1 from sympy import divisors
2
3
4 def test_parfait(n):
5
6     divs = divisors(n)[: -1]
7     somme = sum(divs)
8     if somme == n:
9         print(f"{n} est un nombre parfait.")
10    else:
11        print(f"{n} n'est pas un nombre parfait.")
12
13    # Exemple d'utilisation :
14    n = int(input("Entrez un nombre : "))
15    test_parfait(n)
```

## 3.4 Nombres premiers

Un entier naturel  $n$  avec  $n \geq 2$  est appelé **nombre premier** s'il admet exactement deux diviseurs distincts : 1 et  $n$  lui-même.

En d'autres termes,  $n$  est premier si :

$$\forall d \in \mathbb{N}, d \mid n \implies d = 1 \text{ ou } d = n.$$

### Exemples :

- 2 : ses diviseurs sont 1 et 2 donc 2 est premier.
- 3 : ses diviseurs sont 1 et 3 donc 3 est premier.
- 5 : ses diviseurs sont 1 et 5 donc 5 est premier.
- 7 : ses diviseurs sont 1 et 7 donc 7 est premier.
- 4 : ses diviseurs sont 1, 2, 4 plus de deux diviseurs donc 4 n'est pas premier.
- 6 : ses diviseurs sont 1, 2, 3, 6 plus de deux diviseurs donc 6 n'est pas premier.
- 9 : ses diviseurs sont 1, 3, 9 plus de deux diviseurs donc 9 n'est pas premier.

### 3.4.1 Programme 4 : premiers

```
1 from sympy import isprime
2
3
4 def test_premier(n):
5     if isprime(n):
6         print(f"{n} est un nombre premier.")
7     else:
8         print(f"{n} n'est pas un nombre premier.")
9
10
11 # Exemple d'utilisation :
12 n = int(input("Entrez un nombre : "))
13 test_premier(n)
```

## 3.5 Décomposition d'un entier relatif en produit des facteurs premiers

Tout entier naturel  $n > 1$  peut s'écrire **de manière unique** (à l'ordre des facteurs près) comme un produit de nombres premiers.

Autrement dit, il existe des nombres premiers  $p_1, p_2, \dots, p_k$  et des entiers naturels  $a_1, a_2, \dots, a_k$  tels que :

$$n = p_1^{a_1} \times p_2^{a_2} \times \dots \times p_k^{a_k}$$

avec  $p_1 < p_2 < \dots < p_k$  et  $a_i \geq 1$  pour tout  $i$ .

### Exemples :

Décomposons  $n = 60$  en facteurs premiers.

On a :

$$60 = 2^2 \times 3^1 \times 5^1.$$

Les facteurs premiers sont 2, 3 et 5, avec des exposants 2, 1 et 1 respectivement.

### 3.5.1 Programme 5 : Décomposition

```
1 from sympy import factorint
2
3
4
5 def decomposition_f_p(n):
6 # factorint retourne un dictionnaire {facteur premier : exposant}
7 facteurs = factorint(abs(n))
8 signe = '-' if n < 0 else ''
9
10 # Construire la chaîne du produit
11 decomposition = signe + 'x'.join([f"{p}^{e}" if e > 1 else f"{p}" for p, e in facteurs.items()])
12 print(f"Decomposition en facteurs premiers de {n} :{decomposition}")
13
14
15 # Exemple d'utilisation
16 n = int(input("Entrez un entier relatif : "))
17 decomposition_f_p(n)
```

## 3.6 Plus grand diviseur commun de deux entiers

Soient  $a$  et  $b$  deux entiers relatifs non tous deux nuls. Le **plus grand diviseur commun** (PGCD) de  $a$  et  $b$ , noté  $\text{gcd}(a,b)$  ou  $\text{pgcd}(a,b)$ , est le plus grand entier positif qui divise à la fois  $a$  et  $b$ .  
Autrement dit :

$$\text{pgcd}(a,b) = \max\{d \in \mathbb{N}^* \mid d \mid a \text{ et } d \mid b\}.$$

### Exemples :

$$\text{pgcd}(12,18) = 6,$$

car les diviseurs communs de 12 et 18 sont 1, 2, 3, 6, et le plus grand est 6.

### 3.6.1 Programme 6 : pgcd

```
1 from sympy import gcd
2
3
4 def pgcd(a, b):
5 r = gcd(a, b)
6 print(f"Le PGCD de {a} et {b} est {r}.")
7
8
9 # Exemple d'utilisation :
10 a = int(input("Entrez le premier entier relatif : "))
11 b = int(input("Entrez le deuxième entier relatif : "))
12 pgcd(a, b)
```

## 3.7 Nombres premiers entre eux

Deux entiers relatifs  $a$  et  $b$  (non tous deux nuls) sont dits **premiers entre eux** si leur plus grand commun diviseur (pgcd) est égal à 1 :

$$\text{gcd}(a, b) = 1.$$

Autrement dit,  $a$  et  $b$  n'ont pas de diviseur commun autre que 1 (ou  $-1$ ).

### Exemples :

- 8 et 15 sont premiers entre eux car  $\text{gcd}(8, 15) = 1$ .
- 35 et 64 sont premiers entre eux car  $\text{gcd}(35, 64) = 1$ .
- 14 et 25 sont premiers entre eux car  $\text{gcd}(14, 25) = 1$ .
- 12 et 18 ne sont pas premiers entre eux car  $\text{gcd}(12, 18) = 6$ .
- 20 et 30 ne sont pas premiers entre eux car  $\text{gcd}(20, 30) = 10$ .

### 3.7.1 Programme 7 : Premiers entre eux

```
1  from sympy import isprime
2
3
4  def test_premier(n):
5  if isprime(n):
6  print(f"{n} est un nombre premier.")
7  else:
8  print(f"{n} n'est pas un nombre premier.")
9
10
11 # Exemple d'utilisation :
12 n = int(input("Entrez un nombre : "))
13 test_premier(n)
```

## 3.8 Plus petit multiple commun de deux entiers relatifs non nuls

Soient  $a$  et  $b$  deux entiers relatifs non nuls. Le **plus petit commun multiple** de  $a$  et  $b$ , noté  $\text{ppcm}(a, b)$ , est le plus petit entier naturel strictement positif divisible à la fois par  $a$  et par  $b$ .

Formellement :

$$\text{ppcm}(a, b) = \min\{m \in \mathbb{N}^* \mid a \mid m \text{ et } b \mid m\}.$$

### Exemples :

$$\text{ppcm}(12, 18).$$

Décomposition en facteurs premiers :

$$12 = 2^2 \times 3, \quad 18 = 2 \times 3^2.$$

On prend les facteurs premiers communs et non communs avec l'exposant maximal :

$$\text{ppcm}(12, 18) = 2^2 \times 3^2 = 4 \times 9 = 36.$$

Vérification :

$$36 \div 12 = 3, \quad 36 \div 18 = 2 \quad \text{donc divisible par 12 et 18.}$$

### 3.8.1 Programme 8 : ppcm

```
1     from sympy import lcm
2
3
4     def ppcm(a, b):
5         r = lcm(a, b)
6         print(f"Le ppcm de {a} et {b} est {r}.")
7
8
9     # Exemple d'utilisation
10    a = int(input("Entrez le premier entier (non nul) : "))
11    b = int(input("Entrez le deuxième entier (non nul) : "))
12    ppcm(a, b)
```

# 4 APPLICATION SUR THÉOÈRES DES NOMBRES

## 4.1 Motivation et Introduction

L'étude des propriétés des nombres entiers joue un rôle fondamental dans la formation mathématique, notamment dans les premières années d'apprentissage. Comprendre des notions telles que la parité, les nombres premiers, la factorisation, ou encore les relations entre les entiers (comme le PGCD ou le PPCM) est essentiel pour développer une intuition arithmétique solide.

Cependant, ces concepts peuvent paraître abstraits et difficiles à appréhender lorsqu'ils sont présentés uniquement sous une forme théorique. C'est dans ce contexte qu'est né *Num Quik*, une interface graphique interactive conçue pour rendre l'exploration des propriétés des entiers plus intuitive, visuelle et engageante.

*Num Quik* propose une série d'outils permettant à l'utilisateur :

- de vérifier rapidement si un nombre est pair, premier ou parfait ;
- de décomposer un entier en facteurs premiers ;
- de calculer le PGCD et le PPCM entre deux nombres ;
- de déterminer si deux entiers sont premiers entre eux ;
- de calculer le factoriel d'un nombre donné.

L'objectif principal de ce projet est de fournir un environnement interactif qui relie la théorie des nombres à la pratique, en automatisant les calculs fondamentaux et en rendant leur compréhension plus accessible grâce à une interface claire et colorée.

Développée à l'aide d'un langage de programmation graphique tel que **Python avec PyQt5**, cette application s'adresse à un large public : élèves, étudiants, enseignants ou toute personne souhaitant renforcer sa maîtrise des bases de l'arithmétique. *Num Quik* se veut ainsi à la fois un outil pédagogique efficace et une calculatrice intelligente des propriétés numériques.

## 4.2 Présentation de l'interface graphique

### 4.2.1 Introduction à PyQt5

PyQt5 est un ensemble de bindings Python pour le framework Qt, largement utilisé pour créer des applications avec des interfaces graphiques (GUI). Qt est un framework puissant pour le développement d'applications multiplateformes, et PyQt5 permet de l'utiliser en Python. PyQt5 permet de développer des applications avec des interfaces graphiques complexes tout en étant relativement simple à utiliser.

## 4.2.2 Fonctionnalités principales de PyQt5

- **Widgets** : PyQt5 fournit une large gamme de widgets (boutons, fenêtres, labels, champs de texte, etc.) pour construire l'interface utilisateur.
- **Gestion d'événements** : Il permet de gérer les événements tels que les clics sur les boutons, les frappes de clavier, les changements de fenêtre, etc.
- **Gestion des fenêtres** : PyQt5 permet de gérer les fenêtres principales, les fenêtres secondaires, les boîtes de dialogue, et bien d'autres éléments de l'interface.
- **Graphisme avancé** : PyQt5 offre des outils pour dessiner sur des zones spécifiques de la fenêtre à l'aide de `QPainter`, permettant la création d'applications graphiques (par exemple des jeux ou des logiciels de dessin).
- **Support multiplateforme** : Une application développée avec PyQt5 peut être exécutée sur diverses plateformes telles que Windows, macOS, et Linux sans modification majeure du code.

## 4.2.3 Architecture de PyQt5

- **QtCore** : La bibliothèque principale contenant les classes pour la gestion des événements, des threads, du système de fichiers, etc.
- **QtGui** : Cette bibliothèque gère tout ce qui concerne l'affichage graphique : les fenêtres, les images, les polices, etc.
- **QtWidgets** : Contient les classes de base pour la création de widgets, tels que les boutons, les labels, les zones de texte, etc.
- **QtMultimedia, QtWebEngine** : Pour gérer les fonctionnalités multimédia (audio, vidéo, etc.) et l'intégration du web dans une application.

## 4.2.4 Pourquoi choisir PyQt5?

- **Facilité d'utilisation** : Le framework est bien conçu et permet une prise en main rapide pour les débutants.
- **Puissance et flexibilité** : Il est très puissant pour créer des applications complexes avec une interface riche et interactive.
- **Support de la communauté** : PyQt5 bénéficie d'une grande communauté et de nombreuses ressources, ce qui facilite la résolution de problèmes et l'apprentissage.

## 4.2.5 Limites de PyQt5

- **Taille des applications** : Les applications créées avec PyQt5 peuvent être relativement lourdes.
- **Licences** : PyQt5 est distribué sous la licence GPL, mais une licence commerciale peut être requise si l'application doit être utilisée à des fins commerciales sans code source ouvert.

## 4.2.6 Conclusion

En résumé, PyQt5 est un excellent choix pour développer des interfaces graphiques modernes en Python, offrant une grande flexibilité, une large gamme de widgets et une forte intégration multiplateforme.

## 4.3 Description de l'interface Graphique

Dans ce section on va faire une discription sur l'interface graphique, qu'on a fabriqué avec la bibliothèque PyQt5 : En commençant d'abord par l'interface principale comme le montre sur cette figure suivante :

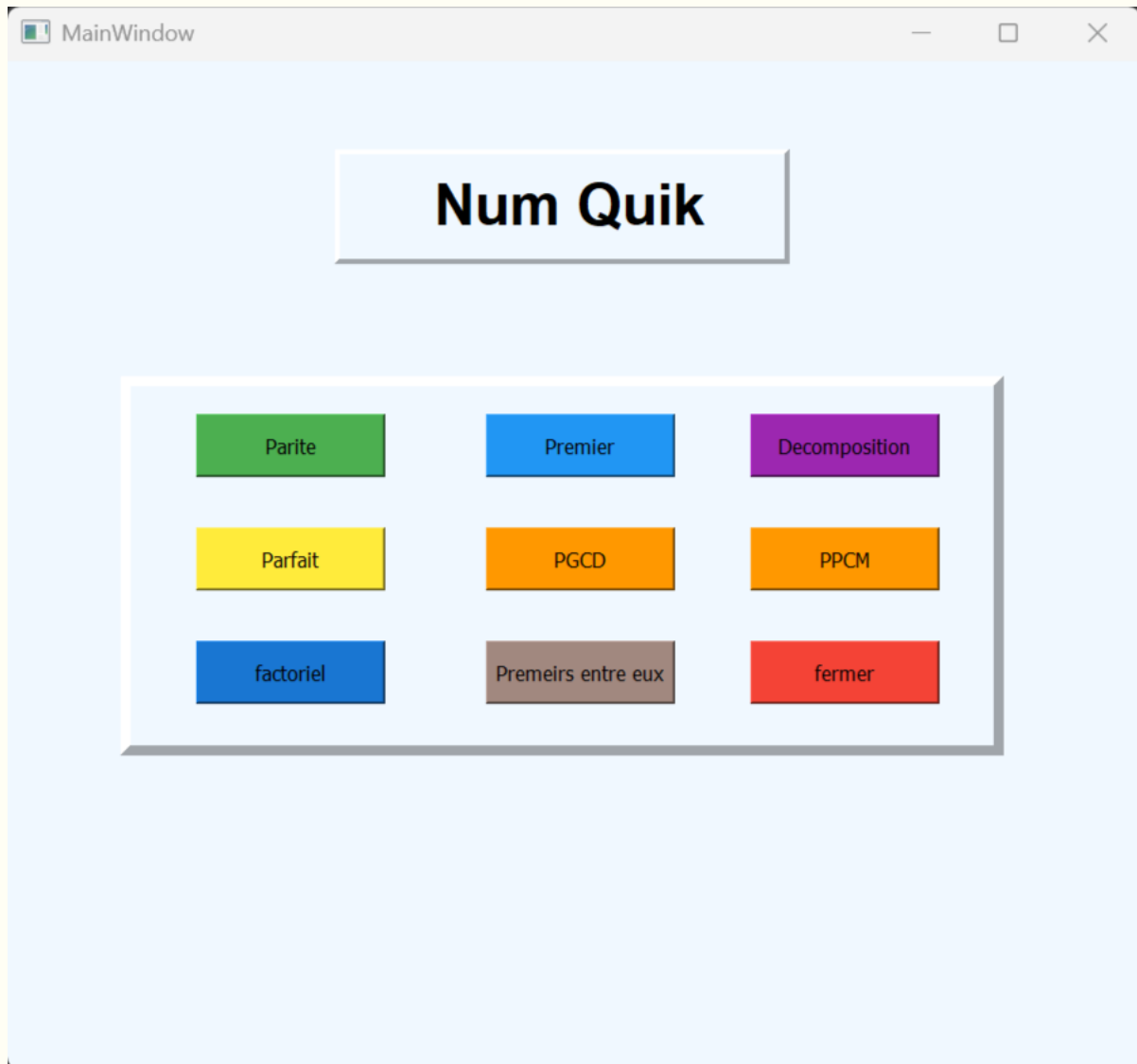



FIGURE 4.1 – L'interface Principale de l'application.

Dans cette interface, nous nous intéressons à l'étude des neuf boutons suivants :

- **Bouton 1** : Vérifier la parité d'un nombre.
- **Bouton 2** : Vérifier si un nombre est premier
- **Bouton 3** : Décomposition d'un nombre en facteurs premiers
- **Bouton 4** : Vérifier si un nombre est parfait
- **Bouton 5** : Calcul du PGCD
- **Bouton 6** : Calcul du PPCM
- **Bouton 7** : Calcul du factoriel d'un nombre
- **Bouton 8** : Vérifier si deux nombres sont premiers entre eux
- **Bouton 9** : Fermer l'application.

#### 4.3.1 Bouton 1 : Vérifier la parité d'un nombre.

Permet à l'utilisateur de vérifier la **parité d'un nombre entier**, c'est-à-dire déterminer s'il est *pair* (divisible par 2 sans reste) ou *impair*. Lorsqu'on clique sur ce bouton, une nouvelle fenêtre s'ouvre pour saisir le nombre à tester, comme montre la figure suivante :



The image shows a Windows-style dialog box with the title "Dialog" and a close button (X). The main content area has a dark blue header with the text "Vérifier la parité d'un nombre". Below the header, there is a label "Entrer la valeur de N:" followed by a white text input field. Underneath the input field is a blue button labeled "Valider". At the bottom of the dialog, there is a label "Resultat:" followed by a white text output field. In the bottom right corner, there is a red button labeled "Fermer".

FIGURE 4.2 – Interface de vérification de la parité d'un nombre.

### 4.3.2 Bouton 2 : Vérifier si un nombre est premier

Permet à l'utilisateur de vérifier si un **nombre entier est premier**, c'est-à-dire s'il ne possède que deux diviseurs : 1 et lui-même. Lorsqu'on clique sur ce bouton, une nouvelle fenêtre s'ouvre pour saisir le nombre à tester, comme montre la figure suivante :



The image shows a Windows-style dialog box with the title "Dialog" and a close button (X) in the top right corner. The main title of the dialog is "Vérifier si un nombre est premier" in white text on a dark blue background. Below the title, there are two input fields. The first is labeled "Entrer la valeur de N:" and is empty. Below it is a blue button labeled "Valider". The second input field is labeled "Resultat:" and is also empty. At the bottom right of the dialog is a red button labeled "Fermer".

FIGURE 4.3 – Interface de test de primalité d'un nombre.

### 4.3.3 Bouton 3 : Décomposition d'un nombre en facteurs premiers

Permet à l'utilisateur de décomposer un nombre entier en produit de facteurs premiers, c'est-à-dire sous la forme d'une multiplication de nombres premiers. Lorsqu'on clique sur ce bouton, une nouvelle fenêtre s'ouvre pour saisir le nombre à décomposer, comme le montre la figure suivante :

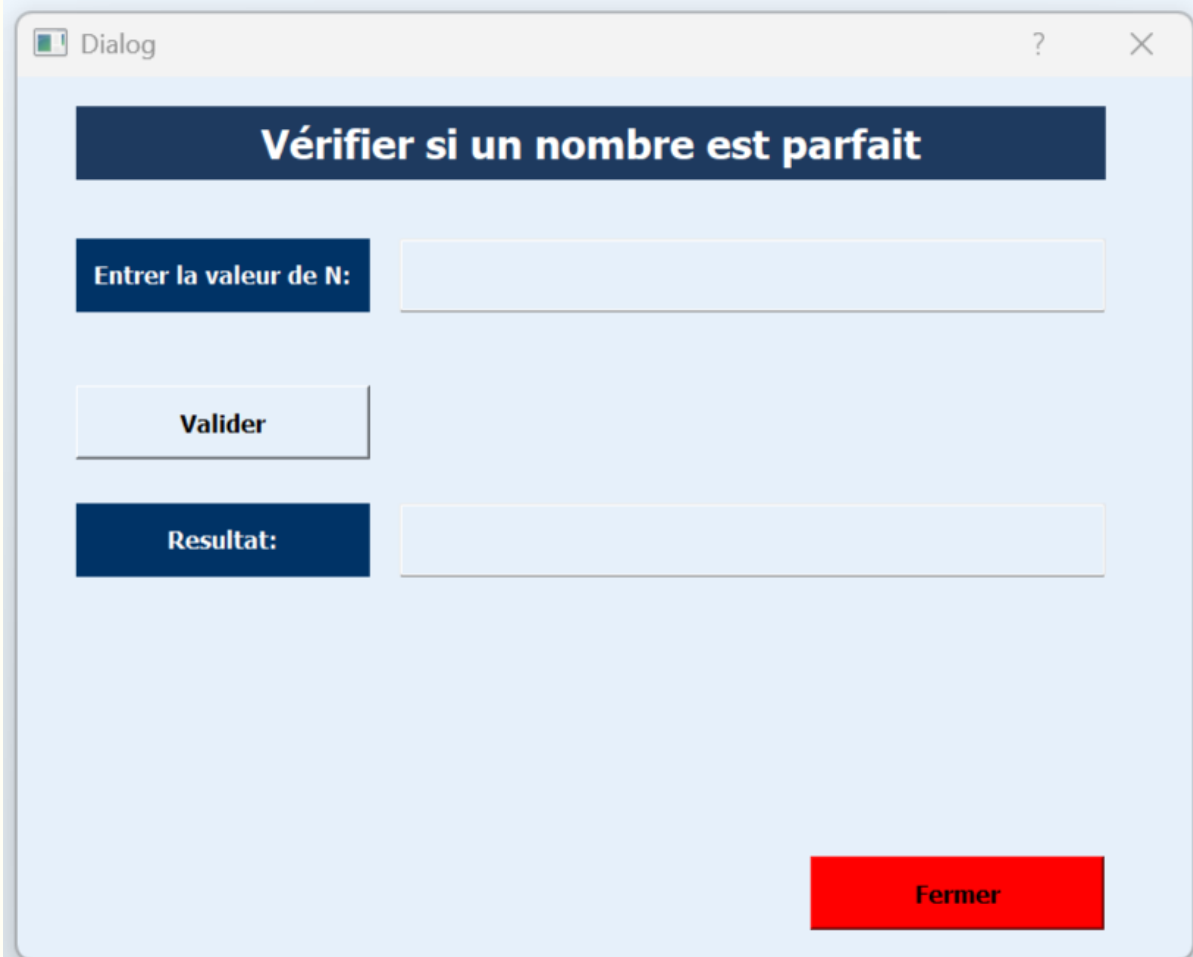


The image shows a software dialog box with a light blue background and a grey title bar. The title bar contains the text 'Dialog' on the left, a question mark icon in the center, and a close button (X) on the right. The main content area has a dark blue header with the text 'Décomposition d'un nombre en facteurs premiers' in white. Below the header, there is a dark blue label 'Entrer la valeur de N:' followed by a white text input field. Underneath the input field is a blue button labeled 'Valider'. Below the 'Valider' button is another dark blue label 'Resultat:' followed by a large white text area for the output. In the bottom right corner of the dialog, there is a red button labeled 'Fermer'.

FIGURE 4.4 – Interface de décomposition en facteurs premiers.

#### 4.3.4 Bouton 4 : Vérifier si un nombre est parfait

Permet à l'utilisateur de vérifier si un nombre entier est parfait, c'est-à-dire si la somme de ses diviseurs stricts (diviseurs positifs inférieurs à lui-même) est égale au nombre lui-même. Lorsqu'on clique sur ce bouton, une nouvelle fenêtre s'ouvre pour saisir le nombre à tester, comme le montre la figure suivante :



The image shows a Windows-style dialog box with a title bar that says "Dialog". The main content area has a dark blue header with the text "Vérifier si un nombre est parfait" in white. Below the header, there is a label "Entrer la valeur de N:" in white on a dark blue background, followed by a white text input field. Underneath the input field is a light blue button with the text "Valider" in black. Below the "Valider" button is another label "Resultat:" in white on a dark blue background, followed by another white text input field. At the bottom right of the dialog box is a red button with the text "Fermer" in white. The dialog box has standard window controls (minimize, maximize, close) in the top right corner.

FIGURE 4.5 – Interface de vérification des nombres parfaits.

#### 4.3.5 Bouton 5 : Calcul du PGCD

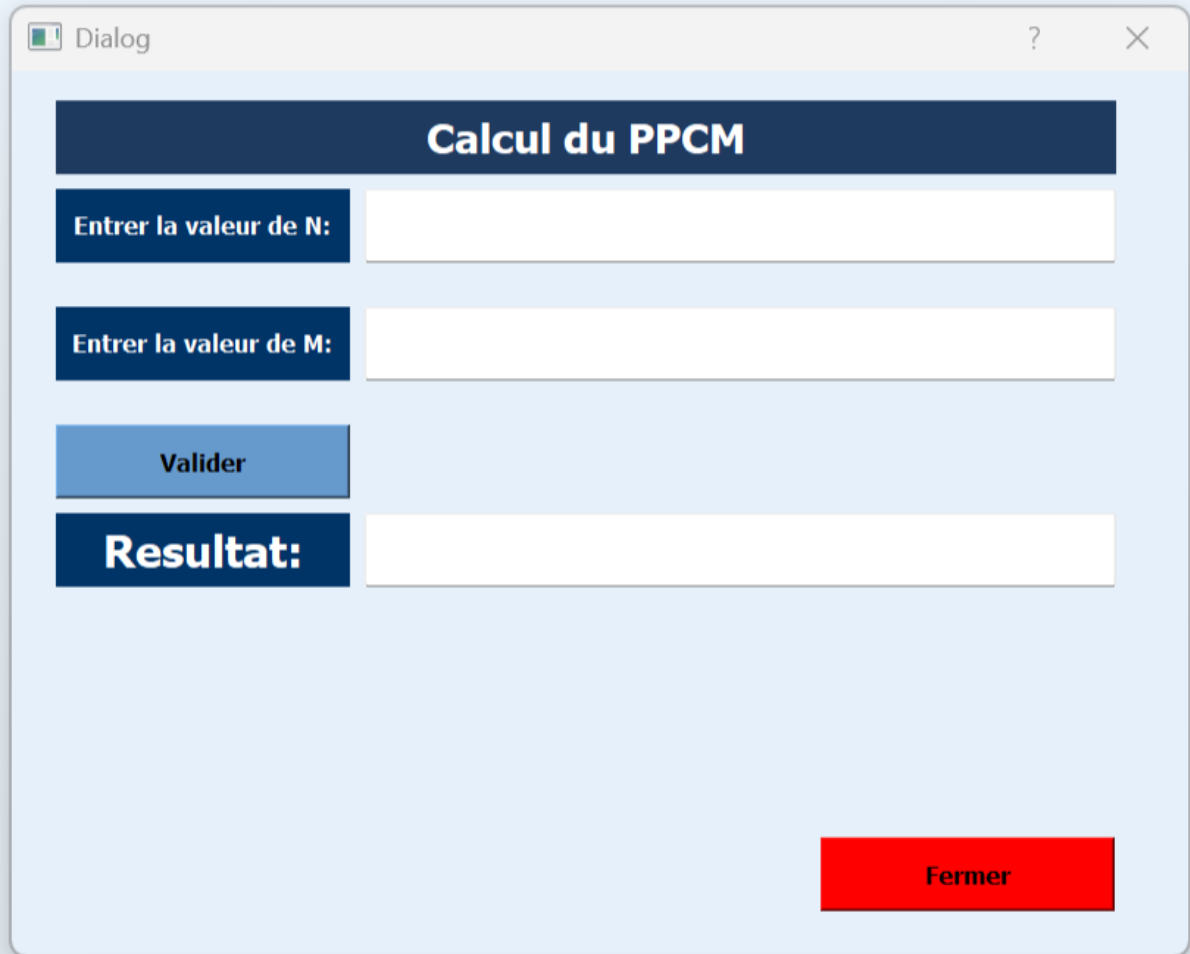
Permet à l'utilisateur de calculer le Plus Grand Commun Diviseur (PGCD) de deux nombres entiers. Le PGCD est le plus grand entier qui divise exactement les deux nombres. Lorsqu'on clique sur ce bouton, une nouvelle fenêtre s'ouvre pour saisir les deux nombres, comme le montre la figure suivante :

The image shows a software dialog box titled "Dialog" with a close button (X) and a help button (?). The main content area has a dark blue header with the text "Calcul du PGCD". Below this, there are two input fields. The first is labeled "Entrer la valeur de N:" and the second is labeled "Entrer la valeur de M:". Below the second input field is a button labeled "Valider". At the bottom left, there is a label "Resultat:" next to an empty output field. At the bottom right, there is a red button labeled "Fermer".

FIGURE 4.6 – Interface de calcul du PGCD de deux nombres.

#### 4.3.6 Bouton 6 : Calcul du PPCM

Permet à l'utilisateur de calculer le Plus Petit Commun Multiple (PPCM) de deux nombres entiers. Le PPCM est le plus petit entier strictement positif qui est un multiple commun des deux nombres. Lorsqu'on clique sur ce bouton, une nouvelle fenêtre s'ouvre pour saisir les deux nombres, comme le montre la figure suivante :

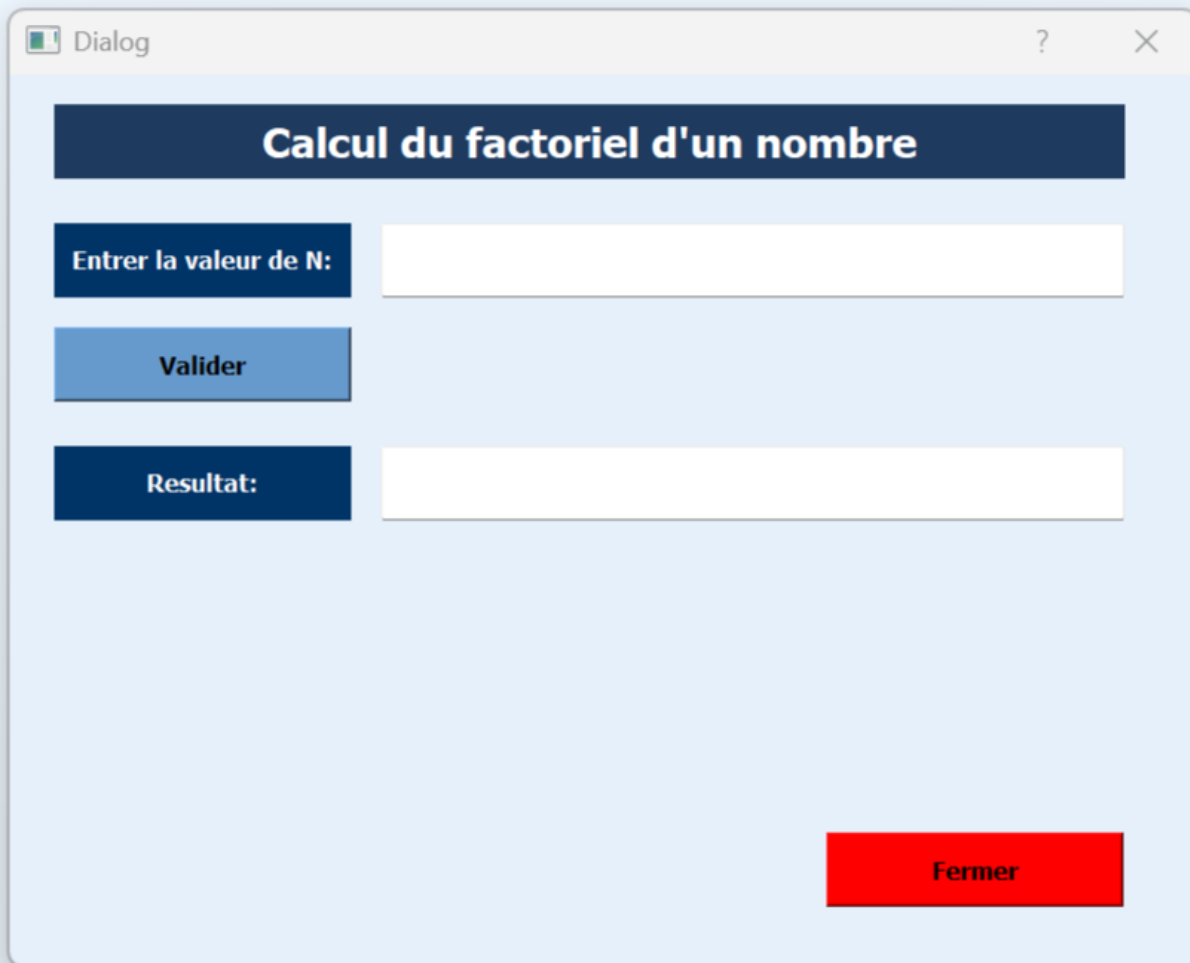


The image shows a Windows-style dialog box titled "Dialog" with a question mark and a close button in the title bar. The main content area has a dark blue header with the text "Calcul du PPCM". Below the header, there are two input fields. The first is labeled "Entrer la valeur de N:" and the second is labeled "Entrer la valeur de M:". Below these fields is a blue button labeled "Valider". At the bottom of the dialog, there is a red button labeled "Fermer". Below the "Valider" button, there is a large white rectangular area labeled "Resultat:" in dark blue text, which is intended for displaying the calculated result.

FIGURE 4.7 – Interface de calcul du PPCM de deux nombres.

#### 4.3.7 Bouton 7 : Calcul du factoriel d'un nombre

Permet à l'utilisateur de **calculer la factorielle d'un nombre entier naturel**, notée  $n!$ , qui correspond au produit de tous les entiers de 1 à  $n$ . Par convention,  $0! = 1$ . Lorsqu'on clique sur ce bouton, une nouvelle fenêtre s'ouvre pour saisir le nombre dont on souhaite calculer la factorielle, comme le montre la figure suivante :



The image shows a Windows-style dialog box titled "Dialog" with a question mark and a close button (X) in the top right corner. The main title of the dialog is "Calcul du factoriel d'un nombre" in white text on a dark blue background. Below the title, there is a dark blue label "Entrer la valeur de N:" followed by a white text input field. Underneath this is a light blue button labeled "Valider". Below the "Valider" button is another dark blue label "Resultat:" followed by a white text input field. In the bottom right corner of the dialog, there is a red button labeled "Fermer".

FIGURE 4.8 – Interface de calcul de la factorielle d'un nombre.

#### 4.3.8 Bouton 8 : Vérifier si deux nombres sont premiers entre eux

Permet à l'utilisateur de **vérifier si deux nombres entiers sont premiers entre eux**, c'est-à-dire si leur Plus Grand Commun Diviseur (PGCD) est égal à 1. Lorsqu'on clique sur ce bouton, une nouvelle fenêtre s'ouvre pour saisir les deux nombres à tester, comme le montre la figure suivante :



The image shows a Windows-style dialog box with a title bar that says "Dialog". The main content area has a dark blue header with the text "Vérifier si deux nombres sont premiers entre eux" in white. Below the header, there are four rows of input fields. The first row has a dark blue label "Entrer la valeur de N:" followed by a white text box. The second row has a dark blue label "Entrer la valeur de M:" followed by a white text box. The third row has a white button labeled "Valider". The fourth row has a dark blue label "Resultat:" followed by a white text box. At the bottom right of the dialog, there is a red button labeled "Fermer".

FIGURE 4.9 – Interface de test de coprimauté de deux nombres.

#### 4.3.9 Bouton 9 : Fermer l'application

Permet à l'utilisateur de **fermer l'application Num Quik**. Lorsqu'on clique sur ce bouton, toutes les fenêtres ouvertes sont fermées et le programme se termine proprement.

Le projet **Num Quik** constitue une contribution pédagogique innovante à l'enseignement des mathématiques, en particulier dans le domaine de l'arithmétique. À travers une interface graphique simple et intuitive, cette application permet d'explorer efficacement des notions fondamentales telles que la parité, la primalité, la décomposition en facteurs premiers, le PGCD, le PPCM, ou encore la factorielle.

Grâce à l'intégration de technologies modernes comme **Python**, **PyQt5** et **SymPy**, le projet a permis de concrétiser une idée simple : rendre les mathématiques plus accessibles, visuelles et interactives, tant pour les élèves que pour les enseignants. *Num Quik* s'inscrit ainsi dans une démarche de modernisation des pratiques éducatives, en valorisant l'utilisation d'outils numériques adaptés à l'apprentissage.

Au-delà de l'aspect technique, ce projet a aussi permis de développer des compétences transversales en algorithmique, en programmation, en conception d'interfaces, ainsi qu'en rédaction scientifique. Il représente donc une expérience formatrice à la fois sur le plan professionnel, didactique et personnel.

En somme, **Num Quik** est un outil à fort potentiel éducatif, facilement extensible à d'autres concepts mathématiques, et prêt à être utilisé dans un cadre pédagogique concret.

- [1] Ministère de l'Éducation Nationale (2019). *Orientations pédagogiques pour l'enseignement des mathématiques au cycle du baccalauréat*. Direction des Curricula, Rabat.
- [2] Tutoriels et ressources Python. <https://www.tresfacile.net/>
- [3] Qt Company. *Qt Designer Manual*. <https://doc.qt.io/qt-5/qtdesigner-manual.html>
- [4] Python Software Foundation. *Documentation officielle Python 3*. <https://docs.python.org/fr/3/>
- [5] SymPy Development Team. *SymPy Documentation*. <https://docs.sympy.org/latest/index.html>
- [6] Riverbank Computing. *PyQt5 Documentation*. <https://www.riverbankcomputing.com/static/Docs/PyQt5/>