
Ecrire directement sur les feuilles de l'énoncé.

Rendre toutes les feuilles (même vierges) avec votre nom dessus.

Documents autorisés : notes de cours et T.D., polys de cours.

Vous veillerez à soigner la présentation des programmes et à choisir des noms de variables « significatifs ». Le nombre de lignes précisé permet d'avoir une idée de la complexité du programme demandé, mais il n'est qu'approximatif et peut varier d'une solution à l'autre. Idem pour le temps indiqué qui doit vous aider à gérer le temps passé sur chaque exercice.

AF1 – Compétences méthodologiques et de conceptualisation (algorithmique)

Exercice I - Conjecture de Goldbach (environ 15 minutes)

La conjecture de Goldbach affirme que tout nombre PAIR supérieur ou égal à 4 est la somme de deux nombres premiers. Ainsi $50 = 19 + 31$

1) Goldbach

Écrire une **fonction** *goldbach* (environ 5 ou 6 lignes) qui :

- reçoit en paramètre le nombre pair x et la liste des nombres premiers inférieurs ou égaux à x
- retourne le couple de nombres premiers si la conjecture est vérifiée pour x , *False* sinon

2) Liste de Goldbach

On suppose que la conjecture est vérifiée et que la fonction *goldbach* retourne toujours un couple (ne retourne jamais *False*). Écrire une fonction *listeGoldbach* (environ 6 ou 7 lignes) qui :

- reçoit en paramètre une valeur n et la liste des nombres premiers inférieurs ou égaux à n
- retourne la liste des couples correspondant à la décomposition de Goldbach pour les **nombres pairs** de 4 à n exclu

Exemple : $\text{listeGoldbach}(20, \text{lpremiers}) \rightarrow [(2, 2), (3, 3), (3, 5), (3, 7), (5, 7), (3, 11), (3, 13), (5, 13)]$

3) Nombre jumeaux

En mathématiques, deux nombres premiers jumeaux sont deux nombres premiers qui ne diffèrent que de 2 :

Exemples : 3 et 5, 5 et 7, 17 et 19

Écrire une fonction *densiteJumeaux* (environ 6 ou 7 lignes) qui :

- reçoit en paramètre une valeur n et la liste des nombres premiers inférieurs ou égaux à n
- récupère la liste des couples de nombres premiers obtenue par la fonction écrite dans la question 2) pour la valeur n
- retourne le nombre de couples « jumeaux » dans la liste obtenue

Exemple : densiteJumeaux(20, lpremiers) → 2 les couples (3, 5) et (5, 7)

Exercice II - Chiffres romains (environ 45 minutes)

On souhaite écrire un programme permettant de convertir un nombre romain en un entier décimal.

- a) les lettres ont respectivement les valeurs suivantes, on les appellera *valeurs romaines* dans la suite de l'énoncé :

- 'M' : 1000
- 'D' : 500
- 'C' : 100
- 'L' : 50
- 'X' : 10
- 'V' : 5
- 'I' : 1

- b) si une lettre se trouve après une lettre de valeur plus grande, on ajoute sa valeur

Exemples : 'XV' représente $10+5$

'CLXXII' représente $172 (100 + 50 + 10 + 10 + 1 + 1)$

- c) si une lettre se trouve avant une lettre de valeur plus grande, on retranche sa valeur

Exemples : 'XL' représente $50-10$

'CDLIV' vaut $454 (-100+500+50-1+5)$

1) Dictionnaire des valeurs

On suppose qu'on dispose d'une liste de couples (*valeur, lettre*) associant à une valeur, la lettre correspondante :

$Isymbols = [[1000, "M"], [500, "D"], [100, "C"], [50, "L"], [10, "X"], [5, "V"], [1, "I"]]$

Écrire la fonction *creeDicos* (environ 7 lignes) qui, à partir de cette liste, crée :

- un dictionnaire associant pour chaque lettre la valeur correspondante
- et le dictionnaire inverse associant pour chaque valeur la lettre correspondante

On veut ainsi obtenir le dictionnaire :

$dicoRomain = \{'M': 1000, 'D': 500, 'C': 100, 'L': 50, 'X': 10, 'V': 5, 'I': 1\}$

$dicoDecimal = \{1000: 'M', 500: 'D', 100: 'C', 50: 'L', 10: 'X', 5: 'V', 1: 'I'\}$

2) Conversion.

Écrire une fonction *romain2dec* (environ 9 à 12 lignes) qui calcule la valeur d'un nombre romain donné sous forme de chaîne de caractères.

Cette fonction reçoit en paramètres :

- Le nombre romain sous forme de chaîne de caractères (exemple : 'DCXIV')
- Le dictionnaire *dicoRomain*

Et retourne la valeur correspondante en représentation décimale

Exemple : pour 'DCXIV' la fonction *romain2dec* retourne 614

La méthode suivante pourra être appliquée :

- i. Trouver la valeur numérique de chaque symbole du chiffre romain fourni :

'D'	'C'	'X'	'I'	'V'
500	100	10	1	5

- ii. Repérer les valeurs qui n'apparaissent pas dans l'ordre décroissant, c'est-à-dire qui sont inférieures au terme qui les suit :

500 100 10 1 5

- iii. La valeur correspondante au nombre romain est calculée en soustrayant ces valeurs et en additionnant les autres

+500 +100 +10 -1 +5

Indications : Attention aux indices dans le parcours de liste

3) Opération inverse

On souhaite écrire une fonction qui effectue l'opération inverse et retourne un nombre romain à partir d'un nombre décimal.

Exemple : pour 276 la fonction `dec2romain` retourne CCLXXVI

Pour ce faire, on va suivre les étapes suivantes :

- i. Écrire une fonction *decomposition* (environ 8 lignes) qui :
 - reçoit en paramètres un nombre *nbre* en décimal et le dictionnaire *dicoDecimal*
 - retourne la liste décrivant la décomposition de *nbre* selon les *valeurs romaines* sous forme de couples [*nb_repetitions*, *valeur*] dans l'ordre décroissant des valeurs. Si le nombre de répétitions est nul, On n'ajoute pas le couple associé.

Exemples : `decomposition(523)` → [(1, 500), (2, 10), (3, 1)]

`Decomposition(276)` → [(2, 100), (1, 50), (2, 10), (1, 5), (1, 1)]

Indications : on peut utiliser la division (//) et le reste de la division (%)

Épreuve d'Informatique – AF1 – 12 mai 2025 - 2h

- ii. Écrire la fonction *dec2romain* (environ 6 lignes) qui réalise la conversion d'un nombre décimal vers un nombre romain.

Cette fonction :

- reçoit en paramètres un nombre en décimal, et le dictionnaire *dicoDecimal*
- retourne le nombre romain correspondant

Pour ce faire, elle récupère la liste construite par la fonction précédente, puis elle transforme chaque couple de cette liste *[nb_repetitions, valeur]* par la répétition des lettres romaines associées.

Exemples : *[(2, 100)]* → "CC"
 [(1, 50), (3, 10)] → "LXXX"

Indications :

- penser à utiliser l'opérateur de répétition *** sur les chaînes de caractères
- on constatera qu'on n'aura pas la version optimale, en effet pour 43 cette méthode retournera XXXXIII et non XLIII, mais on ne demande pas d'aller plus loin dans la résolution du problème