

Livret de liaison entre la première et la terminale NSI

Ce livret de liaison entre la première et la terminale rassemble les notions essentielles à maîtriser à l'entrée en terminale. Il est très fortement conseillé de l'utiliser pour parfaire ses connaissances et tester ses compétences avant la rentrée de septembre 2023. Vous êtes libres de rédiger vos réponses dans l'éditeur de votre choix (Thonny, VS Code, Replit ou Capytale).

QUESTIONS DE COURS

Cette partie liste, sous forme de questions, toutes les notions de première à connaître parfaitement¹. Les réponses se trouvent dans votre cours de première.

Histoire de l'informatique

- Qui est considéré comme le père de l'informatique ?
- Quelles sont les dates clés du développement de l'informatique ?

Architecture des ordinateurs

- Quels sont les principaux composants d'un ordinateur et leurs rôles respectifs ?
- Quelle est la différence entre la mémoire RAM et la mémoire ROM ?
- Quels sont les différents types de mémoire ?
- Expliquez la différence entre un processeur 32 bits et un processeur 64 bits.
- Qu'est-ce qu'un bus informatique et à quoi sert-il ?

Bases de Python

- Combien de types de base existe-t-il en Python et quels sont-ils ?
- Donnez des exemples de syntaxe pour déclarer une condition en Python.
- Comment déclarer une boucle en Python et quelle est la syntaxe pour réaliser une boucle « pour » ?
- Comment écrire une fonction en Python et comment l'appeler ?
- Quelle est la différence entre une liste et un tuple en Python ?
- Comment déclarer un dictionnaire en Python et quelle est la syntaxe pour accéder à une valeur dans un dictionnaire ?

Représentation de l'information

- Comment sont représentés les nombres entiers positifs en binaire et en hexadécimal ?
- Comment sont représentés les nombres entiers relatifs en machine ?
- Comment sont représentés les nombres réels en machine ?
- Comment sont représentés les textes en informatique ?
- Qu'est-ce que l'encodage Unicode ?

1. Certaines notions n'ont malheureusement pas été abordées cette année faute de temps. Il est conseillé de les aborder en autonomie même si elles seront reprises en intégralité en classe de terminale

Logique booléenne

- Quelles sont les principales opérations logiques et leurs symboles ?
- Qu'est-ce qu'une table de vérité ?

Données en tables (fichiers CSV)

- Qu'est-ce que le format CSV et à quoi sert-il ?
- Quelle est la structure d'un fichier CSV ?
- Comment appelle-t-on les données de la première ligne d'un fichier CSV ?
- Comment lire et écrire un fichier CSV en Python ?

Web

- Quel est le rôle du HTML dans la création d'un site Web ?
- Quelle est la structure d'une page HTML et comment la décrire avec des balises ?
- Qu'est-ce qu'une feuille de style CSS ? À quoi sert-elle ?
- Qu'est-ce que le protocole HTTP et comment fonctionne-t-il ?
- Expliquez la différence entre les méthodes GET et POST pour envoyer des informations à un serveur via un formulaire HTML.
- Qu'est-ce que le langage JavaScript ? Quel est son rôle dans la création d'un site Web ?
- Qu'est-ce que le DOM (Document Object Model) et comment interagir avec lui en utilisant JavaScript ?

Algorithmique

- Expliquez le principe du parcours séquentiel d'une liste.
- Comment fonctionne la recherche dichotomique dans une liste triée ?
- Qu'est-ce que la complexité d'un algorithme et comment la mesurer ?
- Citez les différentes classes de complexité d'un algorithme.
- Expliquez les principes des algorithmes de tri par sélection et par insertion. Quelle est leur complexité ?
- Qu'est-ce que la terminaison et la correction d'un algorithme et comment les démontrer ?
- Expliquez le principe des algorithmes gloutons et donnez un exemple d'application.
- Comment fonctionne l'algorithme des k plus proches voisins et quelles sont ses applications ?

Réseaux

- Quels sont les principaux composants d'un réseau informatique et quel est leur rôle ?
- Quels sont les principaux protocoles utilisés sur Internet ?
- Expliquez le principe du modèle OSI à sept couches et donnez quelques exemples de protocoles utilisés à chaque couche.
- Qu'est-ce que le protocole TCP/IP et comment fonctionne-t-il ?
- Expliquez la différence entre les adresses IP et les adresses MAC.
- Citez des commandes Linux relatives au réseau.

Systemes

- Qu'est-ce qu'un système d'exploitation et quel est son rôle ?
- Quelles sont les principales différences entre Linux, Windows et MacOS ?
- Citez des exemples de commandes Linux et leur usage.

EXERCICES À MAÎTRISER

Cette partie contient des exercices sur des algorithmes classiques à savoir programmer à l'entrée de la terminale. Il est important de prendre le temps de la recherche et de faire un premier jet de l'algorithme sur feuille avant de passer à la programmation sur machine. Pour chaque exercice, on pourra se poser la question de la complexité de l'algorithme utilisé. On pourra également, proposer une batterie de tests (en utilisant le mot clé `assert`) pour vérifier la correction de chaque fonction.

La plupart de ces exercices sont extraits de l'excellent site web <https://e-nsi.gitlab.io/pratique/N1/> sur lequel vous pourrez également retrouver les corrections (après avoir cherché suffisamment longtemps et sérieusement : il s'agit ici de s'auto-évaluer de façon honnête).

Exercice 1 - Indice du minimum d'une liste

Écrire une fonction `indice_min` qui prend en paramètre une liste non vide de nombres et qui renvoie l'indice de la première occurrence du minimum de cette liste. On n'utilisera pas les fonctions natives `min` et `index`.

Exercice 2 - Remplacer une valeur

Écrire la fonction `remplacer` prenant en paramètres :

- une liste d'entiers `valeurs`;
- un entier `valeur_cible`;
- un entier `nouvelle_valeur`

et renvoyant une nouvelle liste contenant les mêmes valeurs que `valeurs`, dans le même ordre, sauf `valeur_cible` qui a été remplacé par `nouvelle_valeur`.

Exercice 3 - Maximum

Écrire une fonction `maximum` prenant en paramètre une liste non vide de nombres nommée `nombres` et renvoyant le plus grand élément de cette liste. Chacun des nombres utilisés est de type `int` ou `float`. On n'utilisera pas les fonctions natives `max`, ainsi que `sort` ou `sorted`.

Exercice 4 - Indice de la première occurrence

Écrire une fonction `indice` qui prend en paramètres `element` un nombre entier, `liste` une liste de nombres entiers, et qui renvoie l'indice de la première occurrence de `element` dans `liste`. La fonction devra renvoyer `None` si `element` est absent de `liste`. On n'utilisera pas ni la fonction `index`, ni la fonction `find`.

Exercice 5 - Inversion d'une chaîne de caractères

Écrire une fonction `inverser` qui prend en paramètre une chaîne de caractères `mot` et qui renvoie une nouvelle chaîne de caractères en inversant ceux de la chaîne `mot`.

On s'interdit d'utiliser les fonctions `reverse` et `reversed`, ainsi que les tranches du type `mot[::-1]`.

Exercice 6 - Dernière occurrence

Écrire une fonction `derniere_occurrence`, prenant en paramètre une liste non vide d'entiers et un entier `cible`, et qui renvoie l'indice de la dernière occurrence de `cible`. Si l'élément n'est pas présent, la fonction renvoie la longueur de la liste. On n'utilisera pas la fonction `index`.

Exercice 7 - Moyenne simple

Écrire une fonction en Python qui prend en entrée une liste de nombres et qui renvoie leur moyenne.

Exercice 8 - Palindrome

Un palindrome est un mot qui se lit lettre par lettre, de gauche à droite, exactement comme de droite à gauche.

Écrire une fonction `est_palindrome` qui prend en entrée une chaîne de caractères et qui renvoie `True` si elle est un palindrome, `False` sinon.

Exercice 9 - Double du précédent dans une liste

Écrire une fonction `nombres_puis_double` qui prend en paramètre une liste de nombres entiers, et qui renvoie la liste (éventuellement vide) des couples d'entiers (a, b) qu'il peut y avoir dans la liste tels que b suit a et $b = 2 * a$.

Exercice 10 - Occurrences du minimum

On dispose d'une liste non vide `donnees` d'entiers : les mesures d'un phénomène étudié. Écrire une fonction telle que `occurrences_mini(donnees)` renvoie un tuple composé de deux parties :

- la valeur minimale du phénomène étudié;
- la liste des indices (rangés par ordre croissant) qui sont associés à la valeur minimale.

On s'interdira d'utiliser les fonctions `min` et `index`.

Exercice 11 - Multiplication sans symbole *

Écrire la fonction `multiplication` prenant en paramètres deux nombres entiers `a` et `b` et renvoyant le produit de ces deux nombres. Les seules opérations autorisées sont l'addition et la soustraction.

Exercice 12 - Maximum d'une liste : valeur et indice

Écrire une fonction `valeur_et_indice_du_max` qui prend en paramètre une liste de nombres entiers `valeurs` et renvoie un couple donnant le plus grand élément de cette liste, ainsi que l'indice de la première apparition de ce maximum dans la liste. Pour une liste vide, cette fonction renvoie `(None, None)`.

Il est interdit d'utiliser les fonctions `~index` ainsi que `max`.

Exercice 13 - Factorielle

En mathématiques, la factorielle d'un entier naturel n est le produit des nombres entiers strictement positifs inférieurs ou égaux à n . Par convention $0! = 1$ et on note, pour tout entier $n > 1$, $n! = n \times (n - 1) \times \dots \times 1$. Par exemple, la factorielle de 7 se note $7!$ et est égale à $7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 5040$.

Écrire une fonction qui calcule la factorielle d'un nombre donné.

Exercice 14 - Tri par sélection

Écrire une fonction `tri_selection` qui prend en paramètre une liste `liste` de nombres entiers et qui trie cette liste en place (c'est-à-dire que la liste est modifiée) par ordre croissant des valeurs.

On utilisera l'algorithme suivant. On parcourt la liste de gauche à droite :

- on recherche le minimum de la liste entre cette position courante et la fin de la liste ;
- on échange alors les deux valeurs.

Exercice 15 - Mot à trou

Un *mot* est ici une chaîne de caractères composée uniquement de lettres de l'alphabet. Un *mot à trous* comporte également zéro, une ou plusieurs fois le caractère « . ». On considère que `mot_complet` correspond à `mot_a_trous`, si on peut remplacer chaque « . » de `mot_a_trous` par une lettre de façon à obtenir `mot_complet`. Par exemple :

- `INFO.MA.IQUE` est un mot à trous ;
- `INFORMATIQUE` est un mot qui lui correspond ;
- `AUTOMATIQUE` est un mot qui ne lui correspond pas.

Écrire une fonction telle que `correspond(mot_complet, mot_a_trous)` renvoie un booléen qui détermine si `mot_complet` correspond à `mot_a_trous`.

Exercice 16 - Recherche

Écrire deux algorithmes de complexités différentes qui recherchent un élément donné dans une liste.

Exercice 17 - Dictionnaire des nombres d'occurrences dans une phrase

Écrire une fonction `occurrence_caracteres` prenant comme paramètre une chaîne de caractères `phrase`. Cette fonction doit renvoyer un dictionnaire des nombres d'occurrences des caractères présents dans `phrase`.

Exercice 18 - Anniversaires

On dispose d'un dictionnaire Python dans lequel :

- les clés sont les prénoms de différentes personnes (il n'y a aucun prénom en double);
- les valeurs sont les mois de naissance de ces personnes stockées sous forme de nombres entiers (1 pour janvier, ..., 12 pour décembre).

Par exemple : `naissances = {'Nicolas': 10, 'Antoine': 7, 'Camille': 7}`

Écrire une fonction `anniversaires` prenant en paramètres le dictionnaire `naissances` décrit ci-dessus ainsi qu'un numéro de mois `numero_mois` et renvoyant une liste contenant les prénoms des personnes nées durant ce mois.

Exercice 19 - Rendu de monnaie

On s'intéresse au problème du rendu d'une quantité `somme_a_rendre` de monnaie.

On suppose qu'on dispose d'un nombre infini :

- de billets de 5 euros;
- de pièces de 2 euros;
- de pièces de 1 euro.

Écrire une fonction nommée `rendu` dont le paramètre est un entier positif `somme_a_rendre` qui renvoie un tuple de trois entiers qui correspondent aux nombres de billets de 5 euros de pièces de 2 euros et de pièces de 1 euro à rendre afin que le total rendu soit égal à `somme_a_rendre`, avec le moins de billets et de pièces possible. On commencera par rendre le nombre maximal de billets de 5 euros, puis celui des pièces de 2 euros et enfin celui des pièces de 1 euro.

Exercice 20 - Moyenne de notes avec coefficients

Les résultats aux évaluations d'un élève sont regroupés dans une liste non vide composée de couples (note, coefficient). Dans ce couple :

- `note` est un nombre de type flottant (`float`) compris entre 0.0 et 20.0;
- `coefficient` est un nombre de type entier (`int`) strictement positif

Écrire une fonction `moyenne_ponderee` qui renvoie la moyenne pondérée de cette liste donnée en paramètre.

Exercice 21 - Suite de Fibonacci

Les premiers termes de la suite de Fibonacci sont : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Les deux premiers termes sont 0 et 1. À partir du troisième terme, un terme est la somme des deux précédents.

Écrire une fonction telle que `fibonacci(n)` renvoie le terme d'indice `n` de la suite de Fibonacci. Il s'agit de programmer une fonction très élémentaire, pour des valeurs de `n` inférieures à 25.

Exercice 22 - Liste triée ?

Écrire la fonction `est_triee` qui prend en paramètre une liste de valeurs numériques et qui détermine si cette liste est triée dans l'ordre croissant. La fonction renvoie un booléen `True` ou `False`.

On n'utilisera ni `sort` ni `sorted` dans cet exercice.