

REVISION DE L'ÉPREUVE EP

https://glassus.github.io/terminale_nsi/T6_6_Epreuve_pratique/BNS_2025/

P1 parcours NSI_1

- S7.1 (nombre d'occurrences),
- S34.2 fonction qui propose de jouer au plus ou moins pour trouver une valeur,
- S5.1 (fonction renverse),
- S6.1 liste_puissances sans utiliser **,
- S10.1 (recherche valeur -> True False),
- S8.1 (element maximum d'une liste),
- S13.1 (liste, indice de la valeur recherchée),
- S19.1 (recherche du min avec retour de la premiere occurrence du minimum),
- S20.1 (fonction min et max), S3.2 (eleve du mois),
- S25.1 Écrire la fonction annee_temperature_minimale
- S48.2 variante de la recherche du min avec le calcul de la distance (Listes, fonctions),
- S25.2 (fonctions inverse_chaine et palindrome),
- S18.2 (recherche dichotomique),
- S47.2 (tri selection),
- S37.2 (tri par insertion)

P2 parcours term NSI: construction des bases en python et en algo

- S38.1 (moyenne d'une liste),
- S2.1 (fonction max_et_indice),
- S44.1 (moyenne coef),
- S9.2 (dichotomie),
- S45.1 chaine a trou et chaine normale correspondance (info*ma*tique = informatique),
- S10.2 (Code Cesar, listes, %),
- S35.1 (recherche du max dans les valeurs d'un dico),
- S12.2 numeration romaine, dictionnaire, romains = {"I":1, "V":5, "X":10, "L":50, "C":100 ...
- S16.2 (triangle de pascal, matrice),
- S27.1 (fonction verifie la liste si celle ci est triée),
- S17.2 (insertion dans un tableau à un indice donné),
- S19.2 (fonction separe, qui place les 0 à gauche et les 1 a droite. Parcours par indice avec while),
- S30.1 (liste et codage par difference),
- S36.1 (compter les espaces dans une phrase),
- S37.1 (binaire représenté par une liste de True False -> decimal),
- S37.2 (tri par insertion)

P3 parcours travail recursivité, fondements du debut de term, manip de dict, manip de matrice

- S6.2 code parfait: concatenation de chaine, dictionnaire
- S14.2 (negatif d'une image binarisée en N&B),
- S15.2 (dichotomie recursive),
- S28.2 grille (matrice) et jeu de demineur,
- S44.2 (dessin zoom du coeur),
- S45.2 Suivre un plan d'envoi de type dictionnaire plan_a = {'A':'E', 'B':'F', 'C':'D', 'D':'C', 'E':'B', 'F':'A'} et verifier si ce plan contient un cycle

P4 parcours structures de données de term, dict, pile, file, graphe, AB, POO

- S1.1 (graphe orienté, liste adjacence),
- S8.2 (parenthésage correct, pile en POO),
- S22.2 (distribution d'adresse ipv4 et POO),
- S31.2 (parcours d'un graphe representation par une liste d'adjacence),
- S30.2 (arbre binaire et expression mathematique en npi, POO),
- S33.1 (ABR, noeud en liste imbriquée, récursivité, fonction insertion),
- S35.2 (class Pile pour evaluer une expression NPI)

P5 parcours algo avancés sur les structures de données de term (fusion, infixe, taille...)

- S7.2 (fonction fusion),
- S11.1 (arbre binaire, liste imbriquée, parcours en largeur),
- S12.1 fonction fusion,
- S17.1 (arbre binaire: écrire les fonctions taille et hauteur pour une représentation de l'arbre en POO),
- S24.2 (insertion dans une ABR),
- S47.1 arbre binaire représenté en dictionnaire et contenant des clés-caractères. Ecrire une fonction taille

P6 parcours algorithmes (sequences, tri, glouton, ...)

- S2.2 (sequences),
- S10.2 (Code Cesar, listes, %),
- S9.2 (dichotomie),
- S4.2 (tri bulle),
- S5.2 (crible d'erathostène),
- S13.2 (fonction inserer dans une liste deja triée),
- S34.1 (tri selection),
- S11.2 (programmation dynamique, liste, déterminer la plus grande somme possible de ses éléments),
- S31.1 (sequence recherche motif dans une chaine),
- S32.2 (rendu glouton, listes),
- S33.2 (empaqueter, algo glouton)