

# S'approprier les fondements de l'informatique et du numérique.



## NSI - Numérique & Sciences Informatiques - Les 4 axes du programme

? **Les données**, qui représentent sous une forme numérique unifiée des informations (textes, images, sons, etc...)

? **Les algorithmes**, qui spécifient de façon abstraite et précise des traitements à effectuer sur les données à partir d'opérations élémentaires.

? **Les langages**, qui permettent de traduire les algorithmes abstraits en programmes.

? **Les machines**, et leurs systèmes d'exploitation, qui permettent d'exécuter des programmes en enchaînant un grand nombre d'instructions simples, assurant la persistance des données par leur stockage, et de gérer les communications. On y inclut les objets connectés et les réseaux.

## NSI - Les activités de la classe de première

?? La démarche de projets est au coeur des apprentissages.

## ?? Mini-projet : calculer ses notes au BAC

```
# Proportion contrôle continu et épreuves terminales et mention par défaut
pourcentageCC = 40
pourcentageTerminales = 60
mention = ""

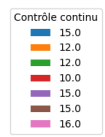
def calculMoyenne(notes,coef):
    """
    Fonction qui calcule une moyenne à partir d'une liste de notes et de coefficients
    """
    print(len(notes),len(coef))
    assert len(notes)==len(coef) ,"Il y a une différence entre le nombre de notes et de coefficients"
    cumul = 0
    for i in range(len(notes)):
        assert notes[i] <=20 , "La note est supérieure à 20"
        assert notes[i] >0 , "Une note est inférieure à zéro"
        assert coef[i] >0 , "Un coef est inférieure à zéro"
        cumul = cumul + (notes[i]*coef[i])
    moyenne = (cumul/sum(coef))
    return moyenne

def saisir(matieres):
    """
    fonction qui saisie les notes
    """
    listeNotes =[]
    for i in range(len(matieres)):
        txt = "Quelle est votre note en "+ matieres[i] + " ?"
        note=float(input(txt))

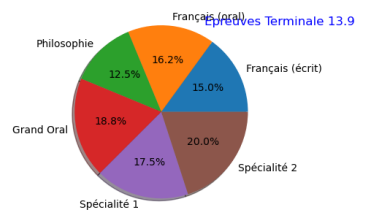
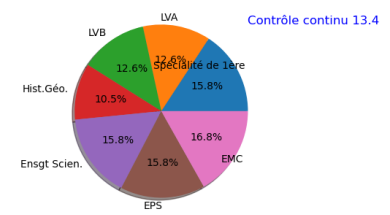
        assert note <=20 , "La note est supérieure à 20"
        assert note >0 , "Une note est inférieure à zéro"
        listeNotes.append(note)
    return listeNotes

# Saisie des notes de contrôle continu
notesControleContinu = saisir(MatieresControleContinu)
#print(notesControleContinu)

# Saisie des notes des épreuves terminales
notesEpreuvesTerminales = saisir(MatieresEpreuvesTerminales)
#print(notesEpreuvesTerminales)
```

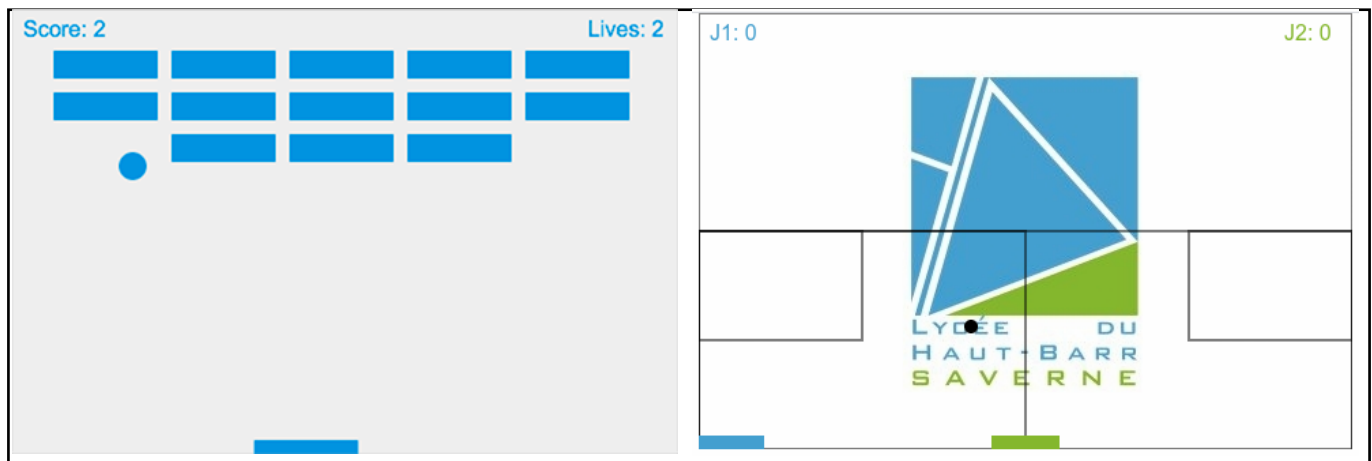


Mes résultats au baccalauréat



Votre moyenne au bac est de 13.7 sur 20 avec la mention assez bien.

## ?? Mini-projet jeu en HTML et JavaScript



## ?? Traitement des données météo de la station du lycée



## ?? PARTICIPATION AU CAPTURE THE FLAG



## ?? Architecture des systèmes et systèmes d'exploitation

A disposition des élèves, un serveur interne



## ?? Les algorithmiques

Algorithmique IA – Machine Learning  
Algorithmique des k plus proches voisins  
Algorithmique de tris

## NSI - Les activités de la classe de terminale

? Structures de données ( arbres, piles, files)

? Langages ( récursivité, objets, etc..)

? Bases de données SQL

? Composants : Processeur et SOC

? Protocoles et routages

Algorithm Path Finding (any) Graphs Initial graph

Max number of steps 250 (Status: Success) Wait

From A To G Bag Heap Bag str Label, cost Heuristic Cost Update result if On target if Update location Extraction Stop if Found target Stop location Extraction Neigh. order West CCW Go to neigh. if ...or better c Mark as seen On push Mark if... Best now Skip if... Best first - R

Graph info: Size: 21, Edges: 23, Simple graph

Fonction cherche(graphe, **départ**, **arrivée**):

- sac (tas minimal): contient les sommets à explorer
- coûts: marquage (meilleur coût total depuis départ)
- preds: meilleur prédécesseur de chaque sommet ajouté
- résultat = Inf (meilleur coût trouvé pour rejoindre arrivée)

Ajouter départ au sac  
Marquer départ avec un coût de 0

Tant que le sac n'est pas vide: (x0)

sommet = extrait un élément du sac  
coût = 39  
Si sommet est arrivée avec meilleur coût: (x0)  
résultat = coût

Si sommet est arrivée:  
Reconstruire le chemin et renvoyer le résultat

Pour chaque voisin de sommet: (x14)  
coût\_voisin (166) = coût + poids arête  
Si voisin jamais vu ou meilleur: (166)  
Ajouter voisin au sac (x10)  
Marquer voisin avec coût\_voisin  
Archiver sommet comme prédécesseur de voisin

Renvoyer valeur par défaut (pas de chemin)

Sac: [ (H,65), (9,65), (8,118), (7,251), (12,166) ]  
Coûts: { A:0, 5:12, 8:118, B:32, H:65, 6:29, 10:37, 9:65, 7:251, C:39, 12:166 }

## NSI - Épreuves du bac

Épreuves du bac coef 16

? Théorie 3 heures 30

3 exercices : 15 points

? Pratique 1 H

Projet : 5 points

## NSI - Les attendues

- faire preuve d'autonomie , d'initiative et de créativité.
- présenter un problème ou sa solution, développer une argumentation.

- coopérer au sein d'une équipe en projet.
- rechercher une information, partager ses ressources.
- faire un usage responsable et critique de l'informatique.

## NSI - La suite...

Écoles d'ingénieurs

prépa MP2I (Mathématiques, physique, informatique et ingénierie) création en 2021.

Université, cursus Math/Info

BUT Informatique

BTS

Etc...