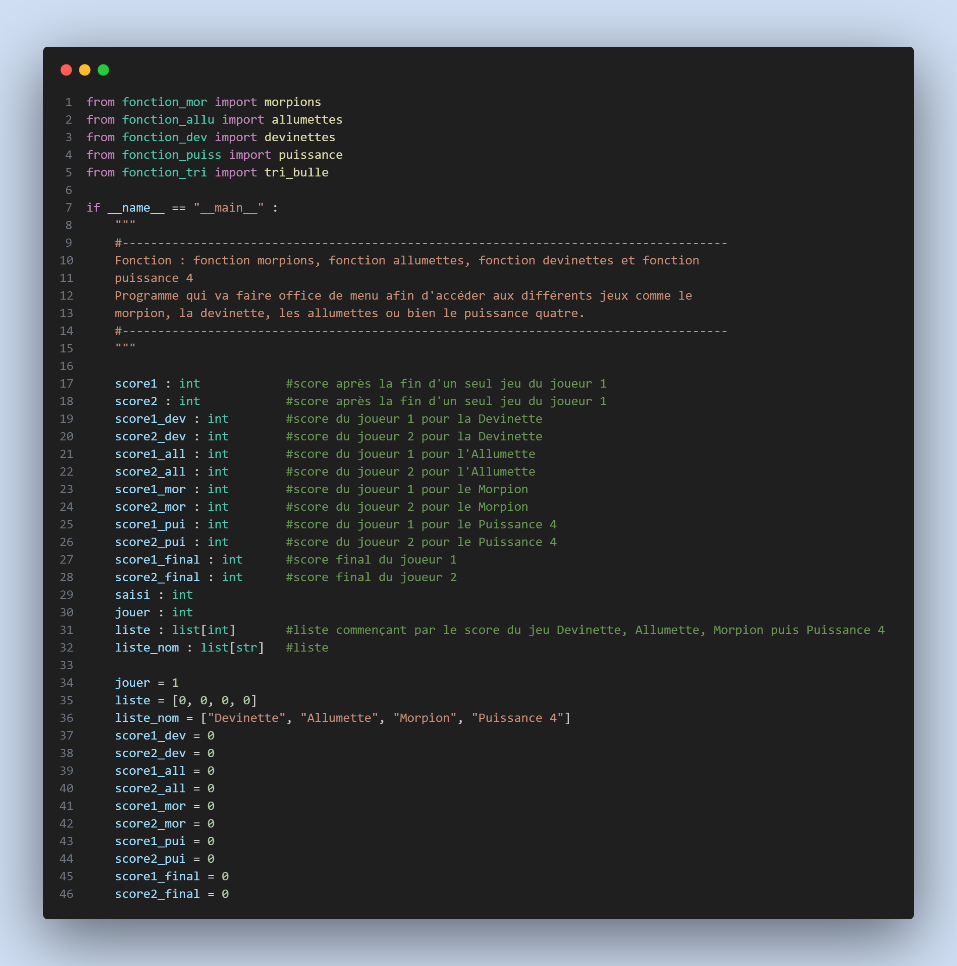
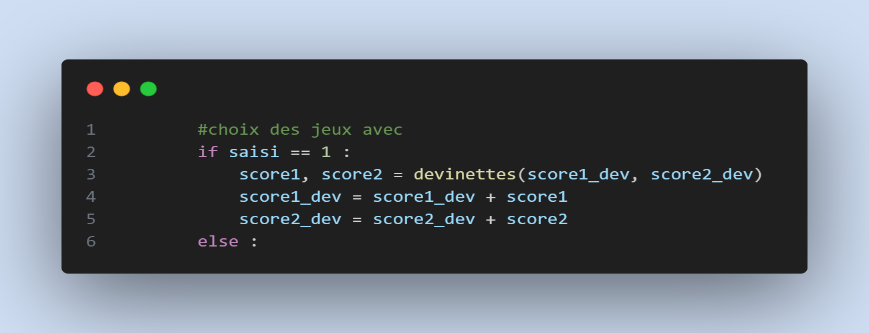
*Sacha Coulombel – Michel Wang*

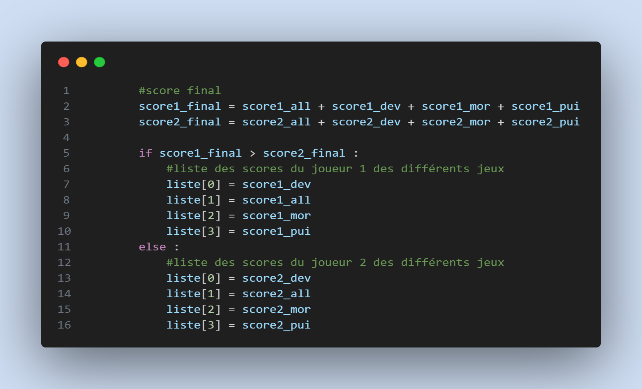
***SAE R1\_01 :***

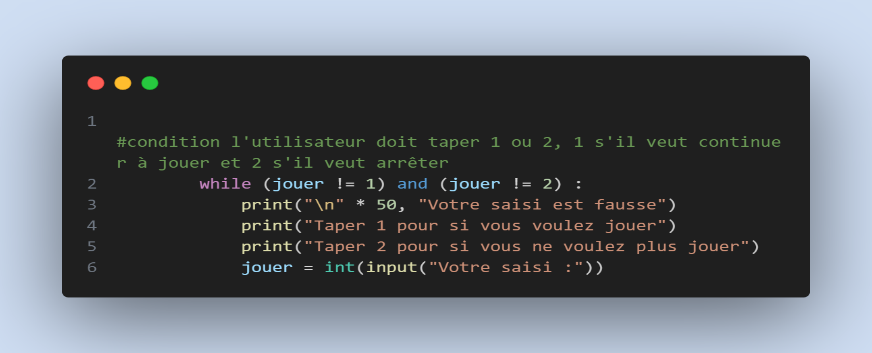
*Voici 4 Codes avec leurs jeux d’essais :*

*Programme Menu :*

*Programme python qui va faire office de Menu. Début avec le programme qui va demander à l’utilisateur de saisir un entier, 1 pour jouer au Devinette, 2 pour Allumette, 3 pour Morpion et 4 pour Puissance4.*

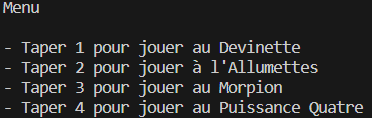
* *Ensuite, la vérification par le programme pour voir si la saisie correspond à 1 ou 2 ou 3 ou 4.*
* *Ensuite, l’utilisation des fonctions en fonction de la sais*ie. Avec le fonctionnement de*s scores*
* *Attribution des scores* à une variable *« score\_final ». Pour ensuite attribuer les différents scores des différents jeux, à une liste (seulement celui du meilleur joueur)*
* *Tous ceci dans une boucle, dans laquelle le programme va demander à l’utilisateur s’il veut rejouer ou non à un jeu. Ensuite la vérification de la saisie : soit 1 pour continuer, soit 2 pour arrêter*
* *Ensuite on sort de la boucle si l’utilisateur saisi 2*
* *On tri la liste avec une fonction tri à bulle*
* *Et puis on affiche le classement des différents jeux du meilleure joueur*

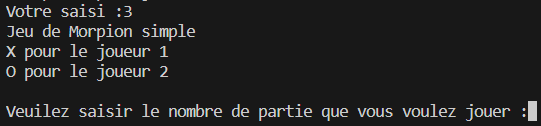
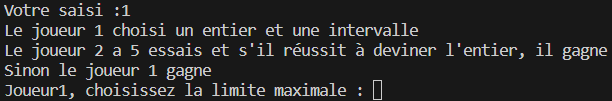
**

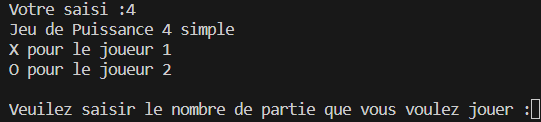
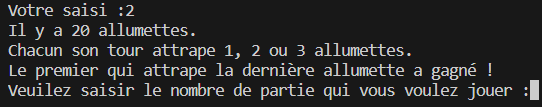
**

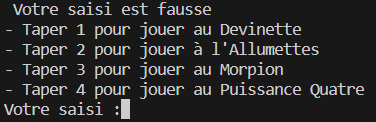
**

*Jeux d’essai :*

**

**

**

**

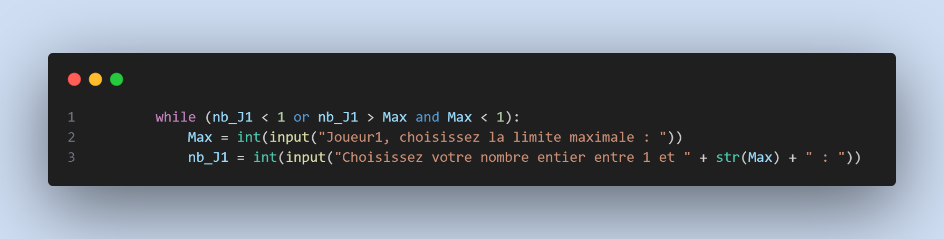
*Une image contenant texte, Police, capture d’écran, typographie

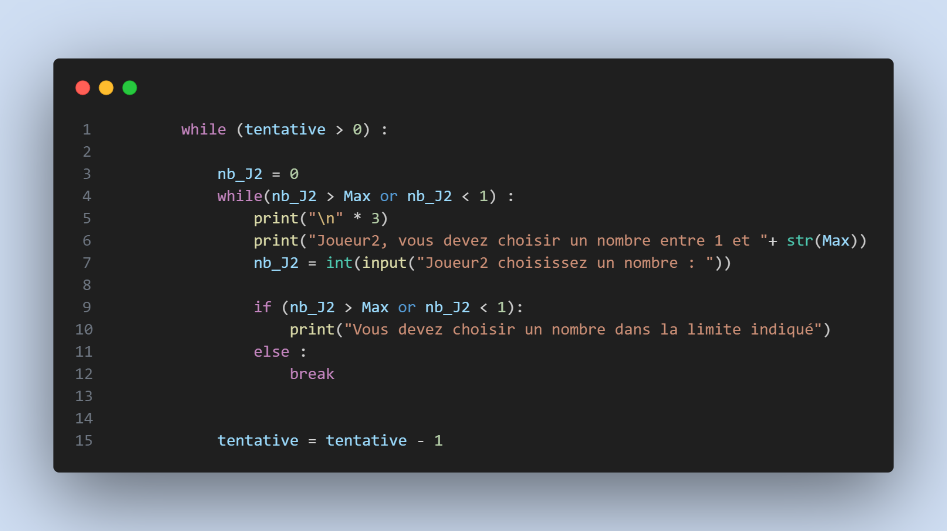
Description générée automatiquement*

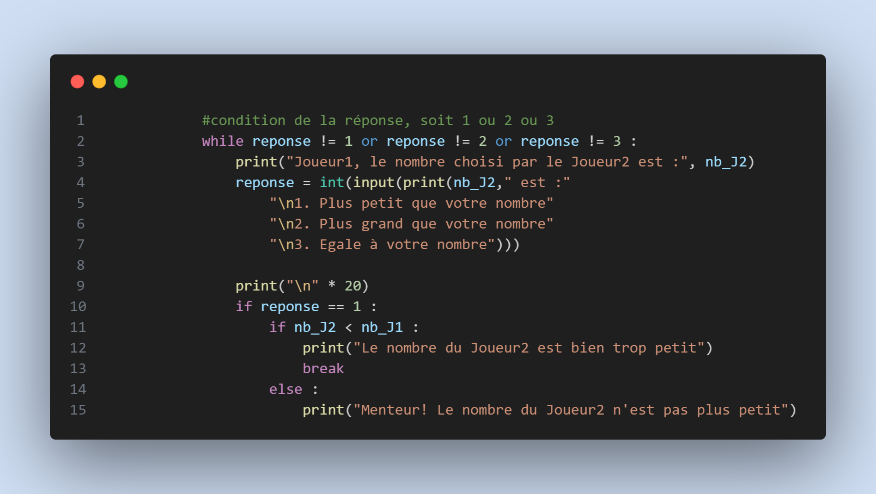
*Programme Devinettes :*

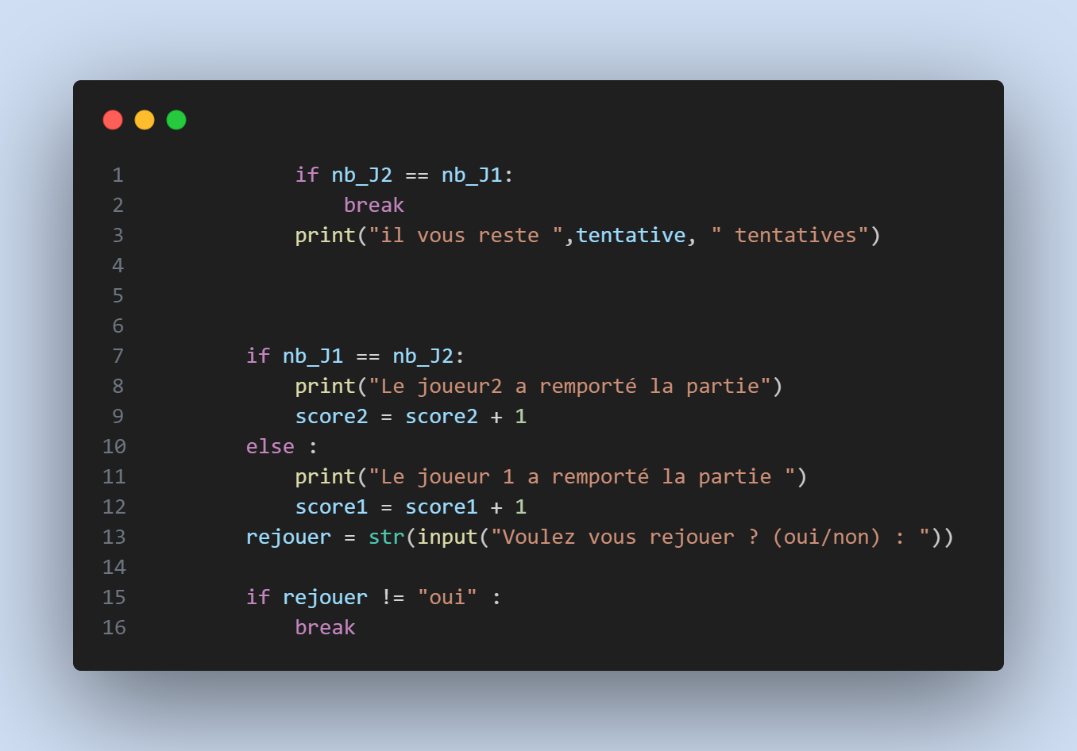
*Fonction qui va prendre en paramètre d’entrée le score du joueur 1 et du joueur 2. Et va renvoyer les mêmes scores mais avec des différentes valeurs en fonction de qui a gagné :*

* *Le joueur 1 qui saisi l’entier à deviner. Et va ensuite saisir un intervalle entre 1 et un nombre maximal qu’il va également saisir.*
* *Le joueur 2 qui aura 5 tentatives de deviner ce nombre. Pour cela, une boucle tentative qui baisse de 1 à chaque itération. A 0, le joueur 2 perd et le joueur 1 gagne*
* *Contrôle de la saisie du joueur 2, que sa saisie est bien dans l’intervalle*
* *Après chaque réponse du joueur 2, le joueur 1 donne son avis.*
* *Attribution des scores en fonction du gagnant et la renvoie au programme menu*

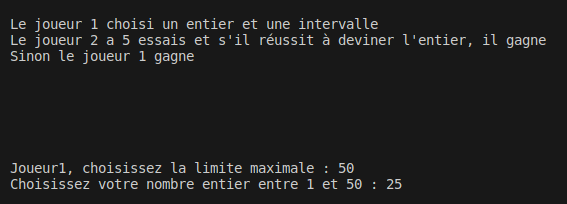


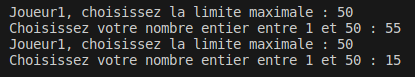
**

**

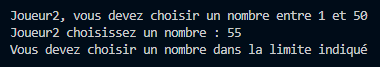
**

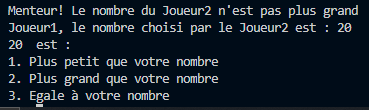
*Jeux d’essais :*



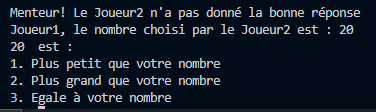


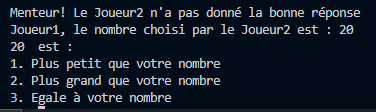


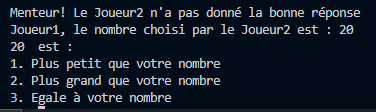






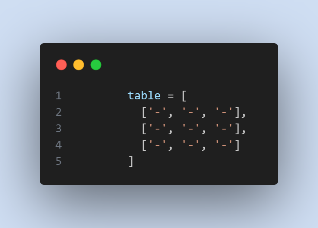


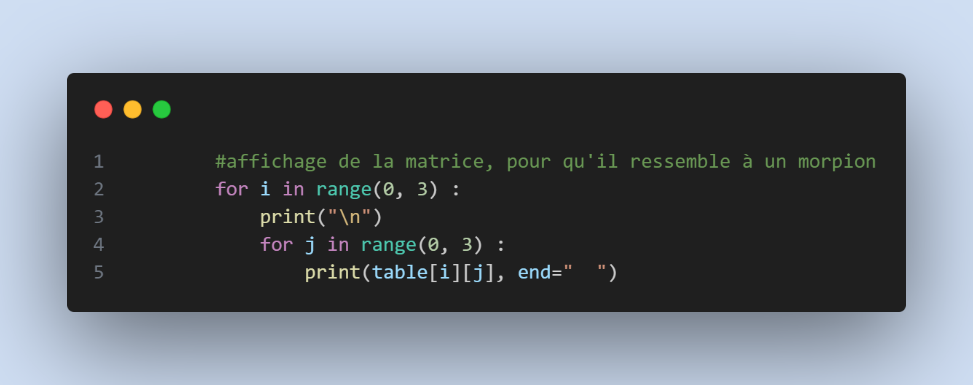


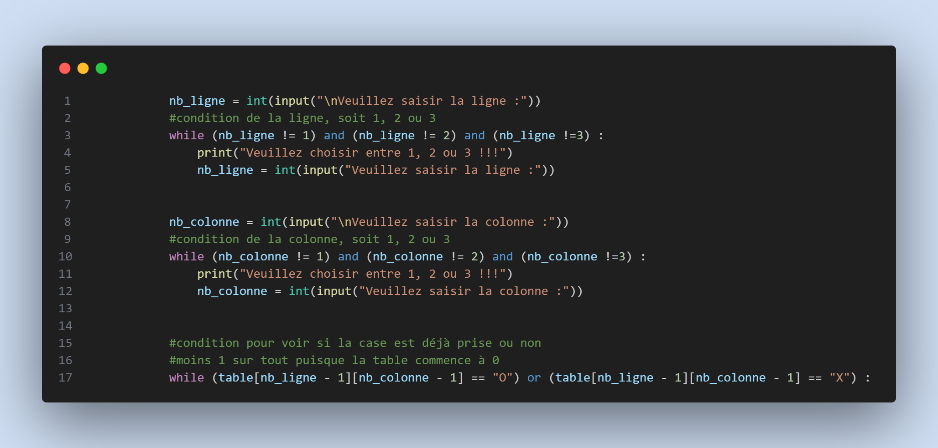


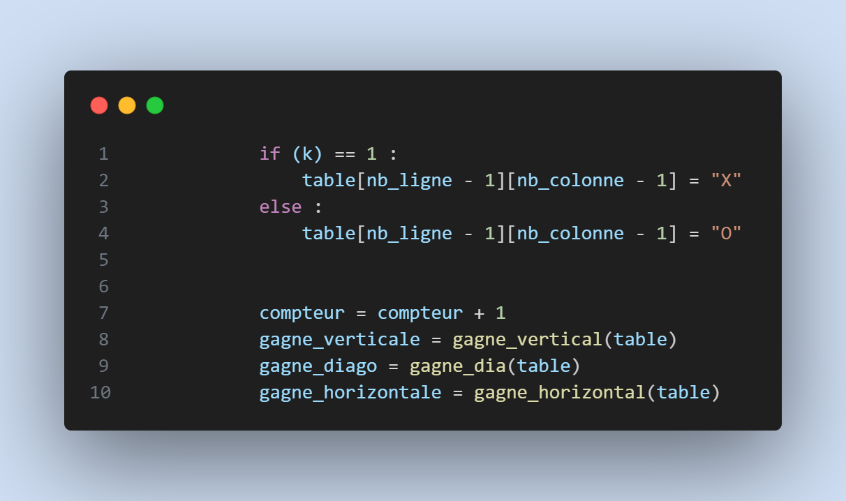
*Programme Morpion :*

*Fonction qui va prendre en paramètre d’entrée le score du joueur 1 et du joueur 2. Et va renvoyer les mêmes scores mais avec des différentes valeurs en fonction de qui a gagné :*

* *Initialisation d’une liste à double dimension. Soit 3 sous-listes de 3 cases. Afin d’avoir une matrice de 3 lignes et 3 colonnes pour le jeu de morpion*
* *Boucle « for » pour afficher une matrice qui ressemble au morpion*
* *Entrée dans une boucle pour saisir le nombre de partie souhaité*
* *Entrée dans une autre boucle qui vérifie si toute les cases sont remplies ou si l’un des joueurs a gagné ou non*
* *Dans cette boucle, demande à l’utilisateur de saisir la colonne et ligne pour jouer. Vérification de sa réponse (si la case est déjà prise ou non, et s’il a bien saisi un nombre entre 1 et 3)*
* *Si pas de problème, entrée de 0 ou X en fonction du joueur à la case choisie*
* *Sort de cette boucle si l’une des conditions de la boucle est remplie*
* *Attribution du score en fonction du gagnant*
* *Renvoie au programme Menu à la toute fin*

**

**

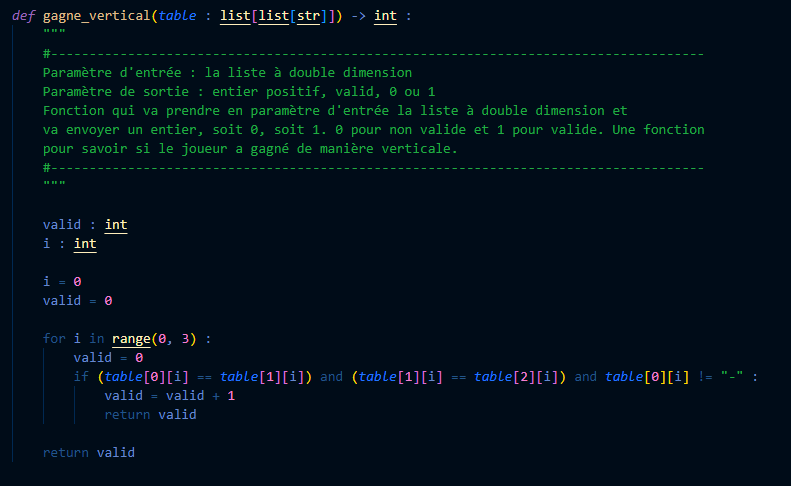
**

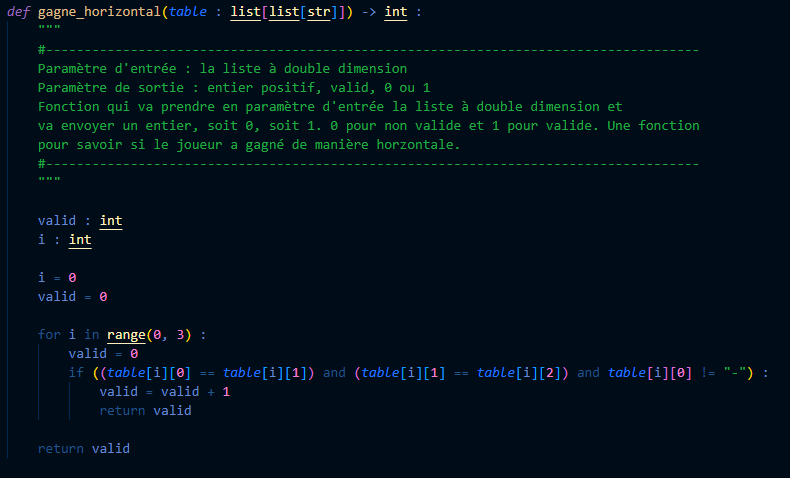
**

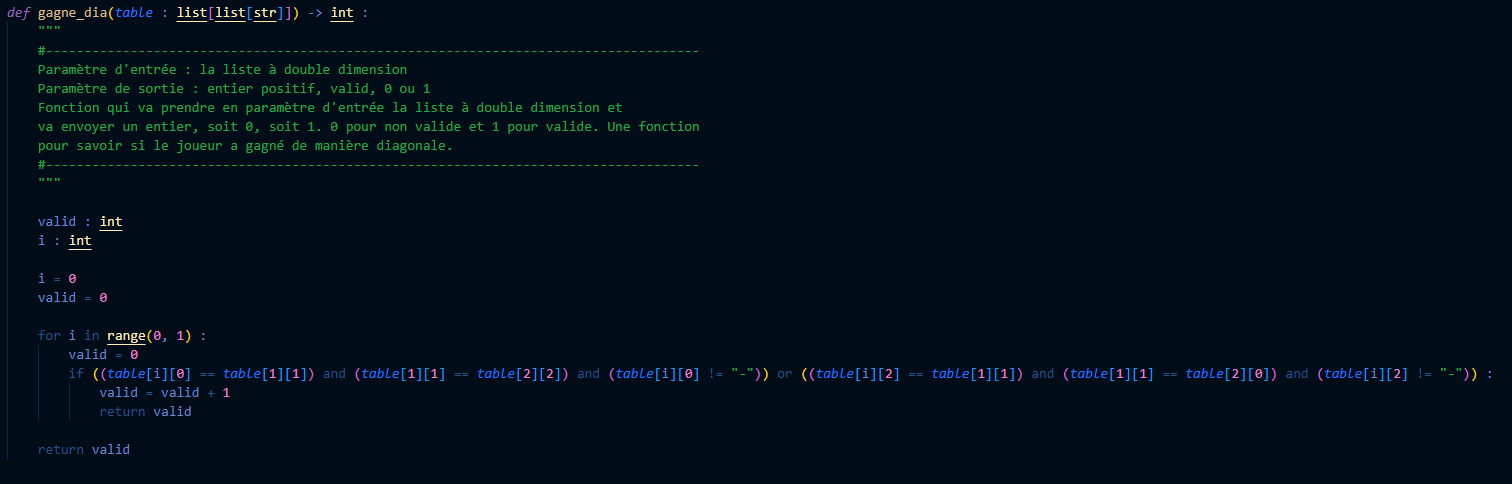
*3 fonctions complémentaire pour la fonction principale de morpion. Une fonction pour gagner verticalement, une autre pour gagner horizontalement et une dernière pour gagner de manière diagonale : (dans tous ces cas, il faut aligner 3 cases).*

*Les 3 fonctions prennent en paramètre d’entrée la liste qui représente le morpion. Et vont renvoyer un entier. L’entier sera égale à 1 si la condition pour gagner est valide. Et puis va renvoyer 0 si la condition n’est pas vérifiée :*

* *Pour la fonction gagne\_vertical. On utilise une boucle « for » pour parcourir toute la liste et vérifie si les valeurs de la liste d’une même colonne, sont similaires*
* *Pour la fonction gagne\_horizontal. On fait la même chose mais on vérifie cette fois s’ils ont de la même ligne*
* *Pour la fonction gagne\_dia. On utilise une boucle « for » pour parcourir toute la liste et vérifie si les valeurs de la liste de même ligne et même colonne sont similaires*

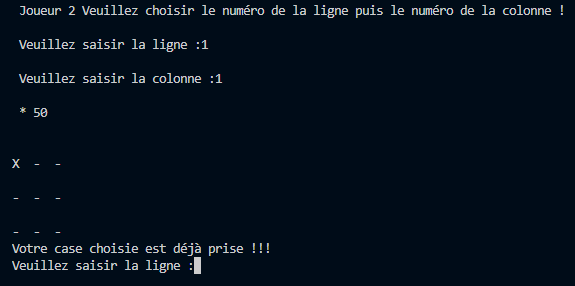
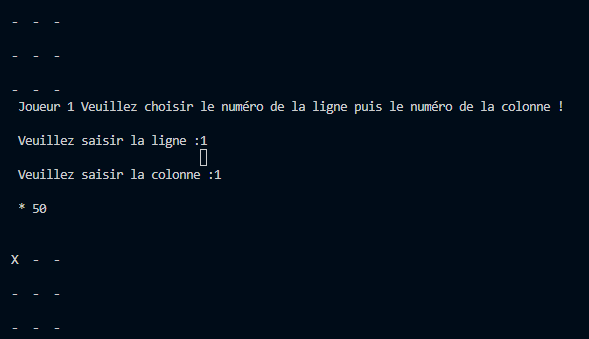


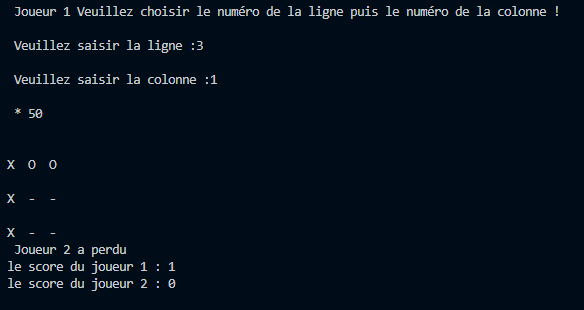


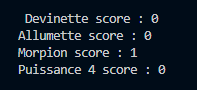


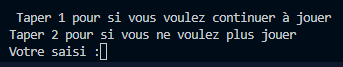
*Jeu d’essai :*









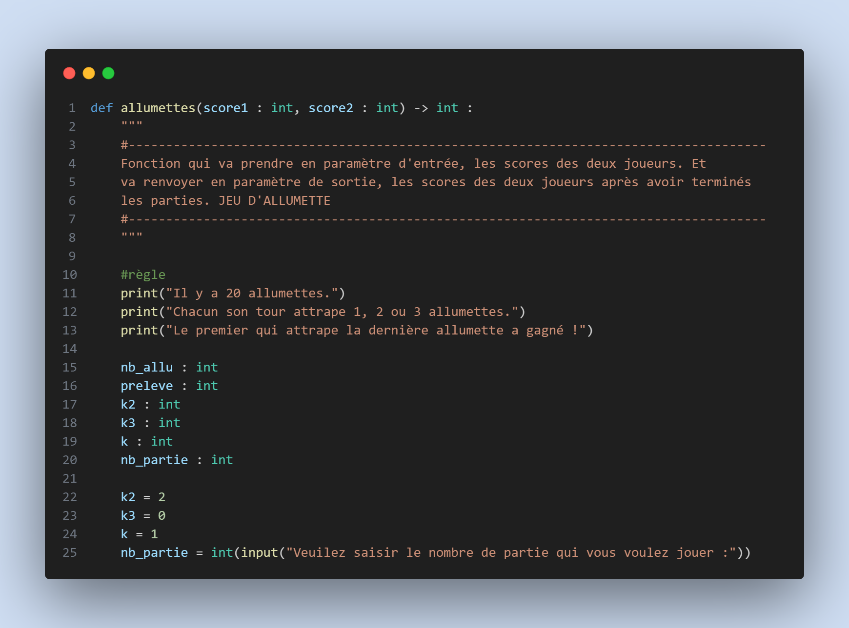


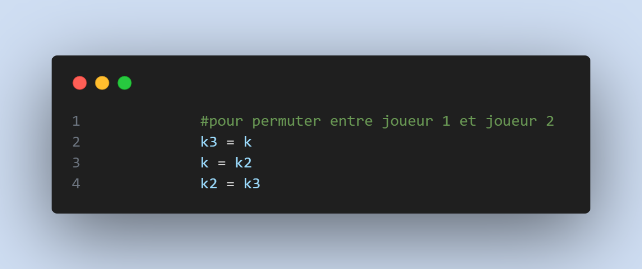
*Programme Allumettes :*

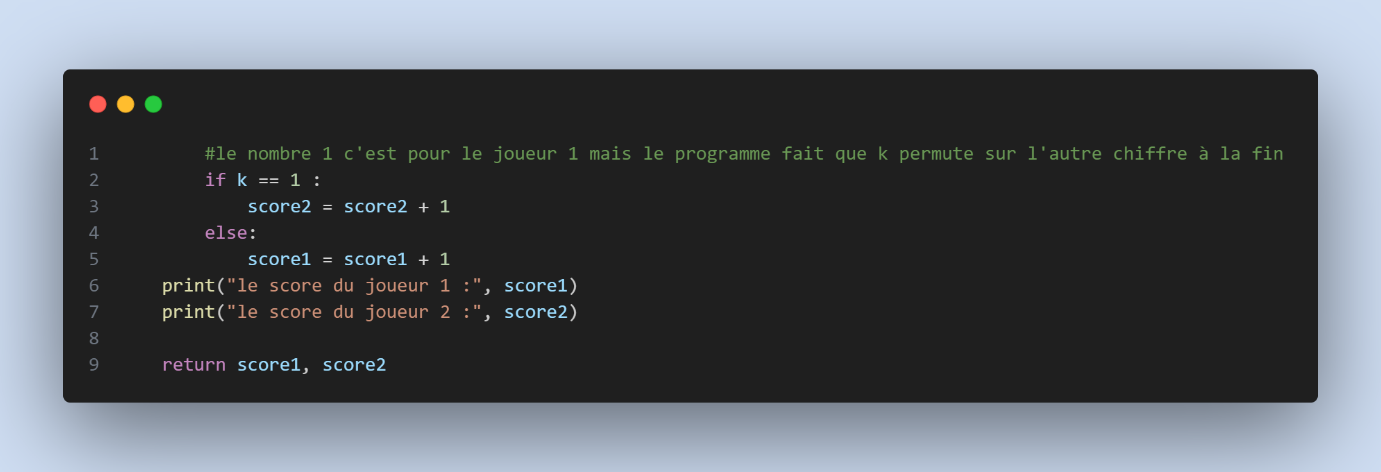
*Fonction qui va prendre en paramètre d’entrée le score du joueur 1 et du joueur 2. Et va renvoyer les mêmes scores mais avec des différentes valeurs en fonction de qui a gagné :*

* *Commence avec une boucle qui va permettre à l’utilisateur de choisir le nombre de partie qu’il voudra jouer*
* *Ensuite demande à l’utilisateur de saisir un entier entre 1 et 3. Vérification de la saisie si c’est bien 1 ou 2 ou 3. De plus la vérification si le joueur peut bien enlever n nombre d’allumette en sachant qu’il faut qu’il en reste un seul*
* *Le nombre total d’allumette qui baisse en fonction de la saisie de l’utilisateur*
* *La permutation de la variable k pour permuter entre joueur 1 et joueur 2*
* *A la fin lorsque le nombre total d’allumette atteint 1, on attribue plus un au score en fonction du joueur gagnant*

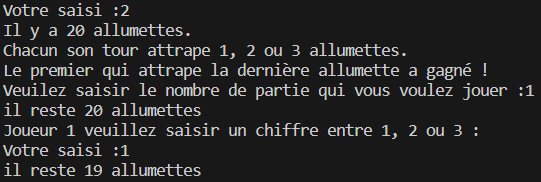
*Une image contenant texte, capture d’écran, Police, conception

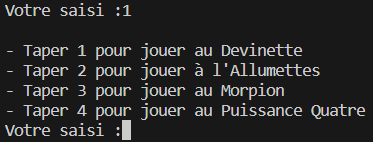
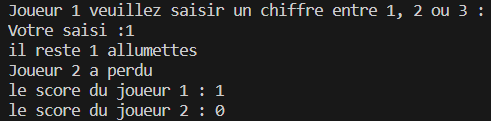
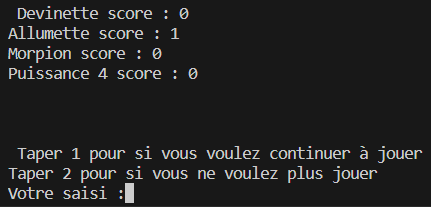
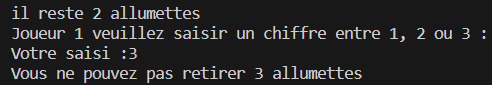
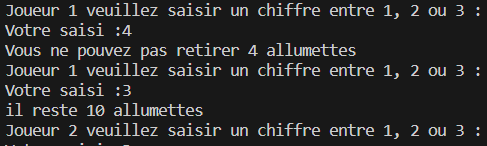
Description générée automatiquement*

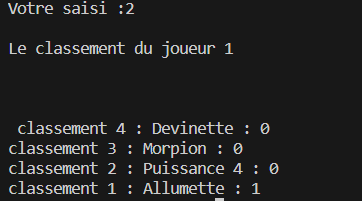
**

**

*Jeux d’essai :*

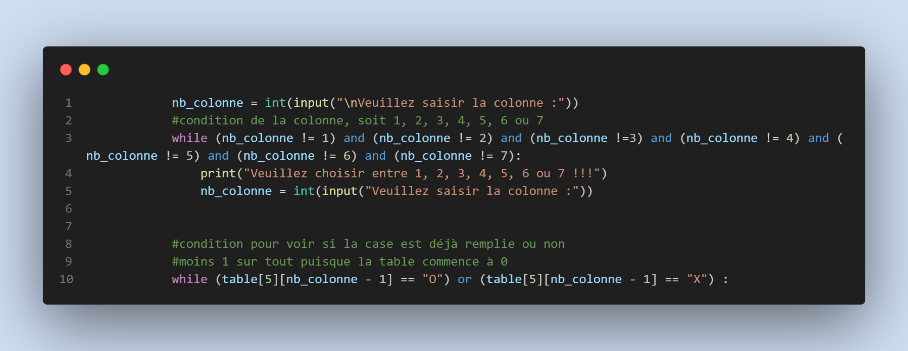
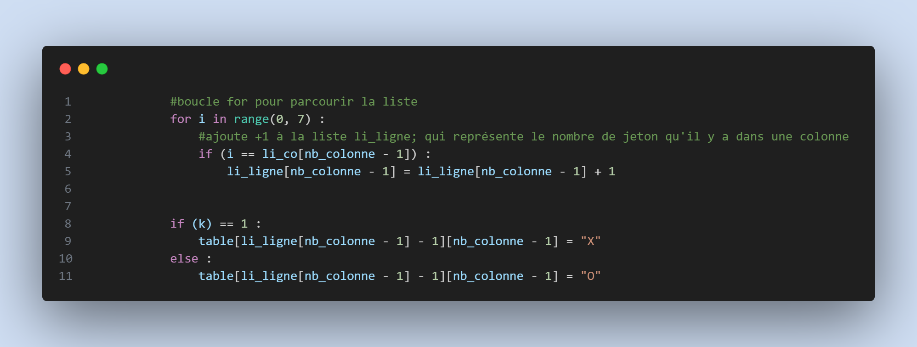
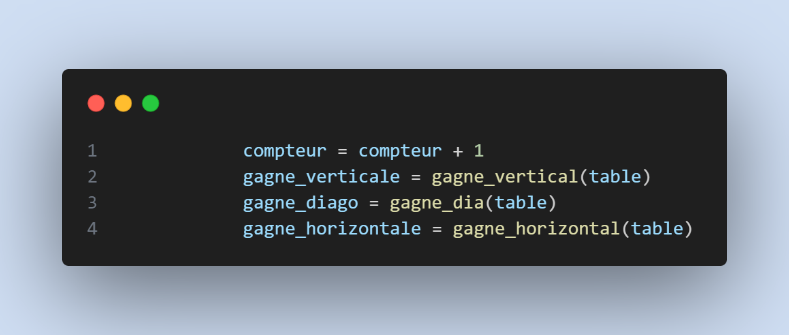
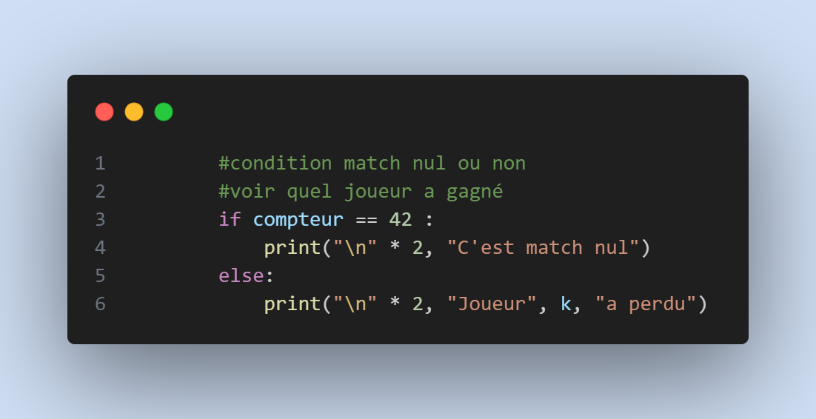
**

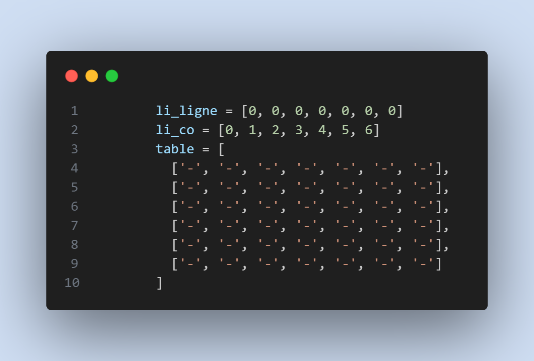
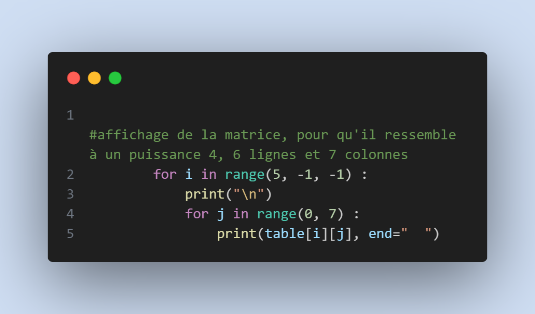
**

**

*Programme Puissance 4 :*

*Fonction qui va prendre en paramètre d’entrée le score du joueur 1 et du joueur 2. Et va renvoyer les mêmes scores mais avec des différentes valeurs en fonction de qui a gagné :*

* *Initialisation d’une liste à double dimension. Soit 6 sous-listes de 7 cases. Afin d’avoir une matrice de 6 lignes et 7 colonnes*
* *Boucle « for » pour afficher une matrice qui ressemble au jeu de puissance 4*
* *Entrée dans une boucle pour saisir le nombre de partie souhaité*
* *Entrée dans une autre boucle qui vérifie si toute les cases sont remplies ou si l’un des joueurs a gagné ou non*
* *Dans cette boucle, demande à l’utilisateur de saisir seulement la colonne. Vérification de sa réponse (si la case est déjà prise ou non, et s’il a bien saisi un nombre entre 1 et 7)*
* *Si pas de problème, entrée de 0 ou X en fonction du joueur à la case choisie*
* *Sort de cette boucle si l’une des conditions de la boucle est remplie*
* *Attribution du score en fonction du gagnant*
* Une image contenant texte, carte de visite, capture d’écran, Police

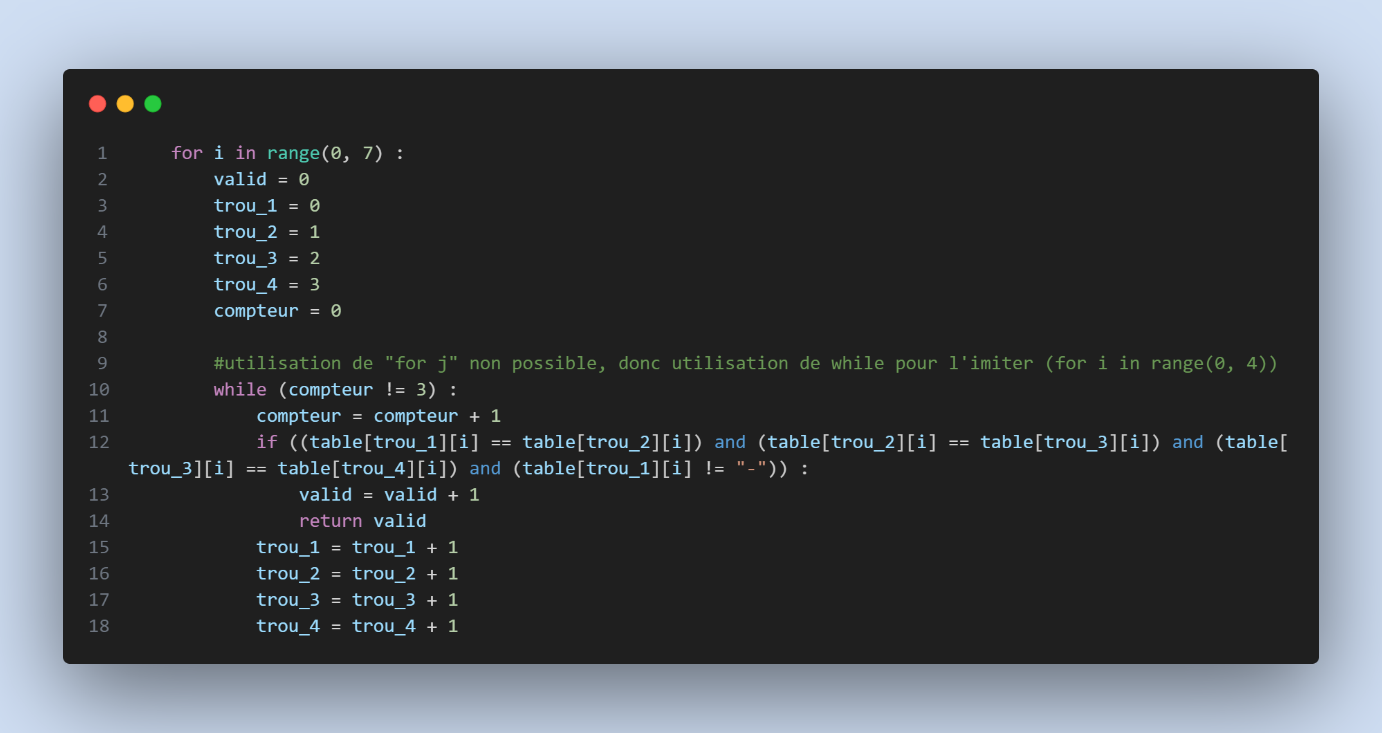
  Description générée automatiquement*Renvoie au programme Menu à la toute fin*

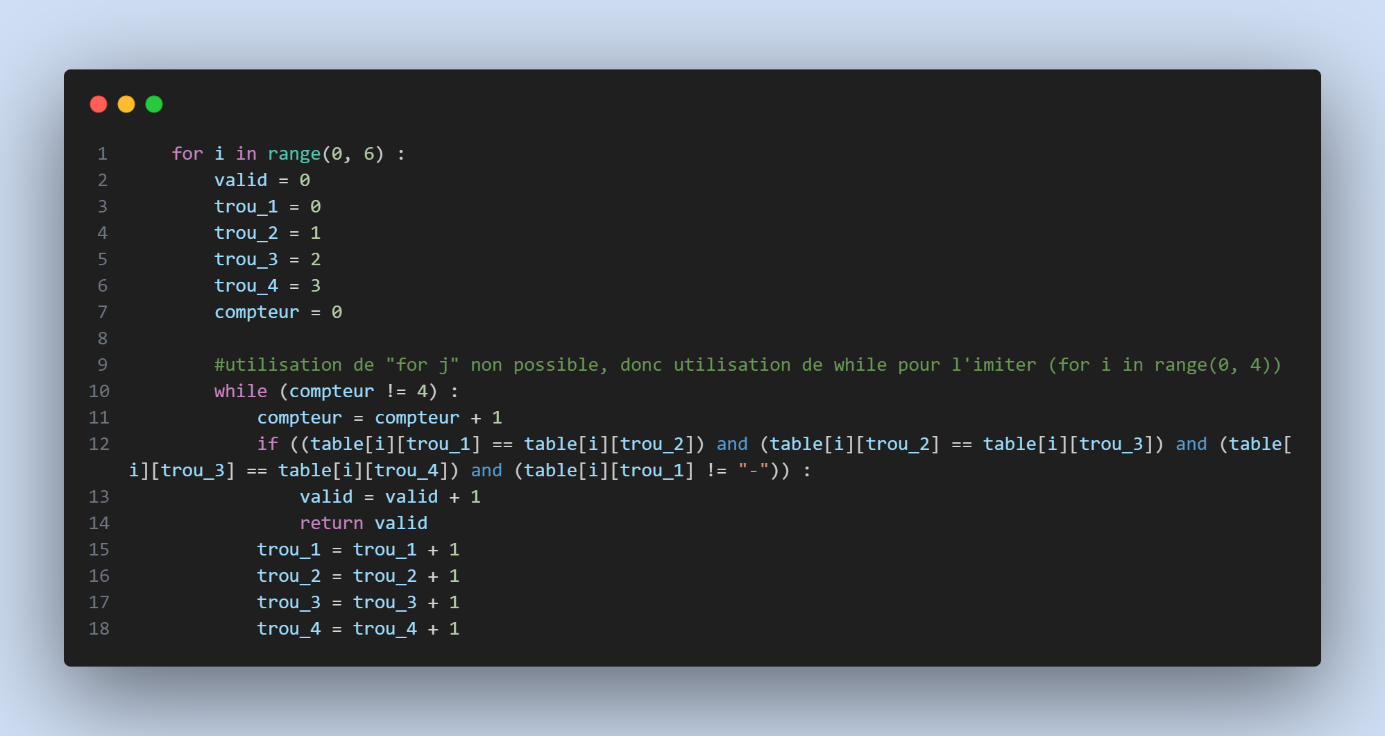
*3 fonctions complémentaires, pour la fonction principale du jeu de Puissance 4. Une fonction pour gagner verticalement, une autre pour gagner horizontalement et une dernière pour gagner de manière diagonale : (Dans tous les cas, il faut aligner 4 cases)*

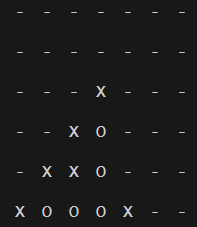
*Les explications sont quasi les même que le morpion. La différence c’est qu’on teste sur une plus grande liste. Et donc les conditions sont plus grandes*

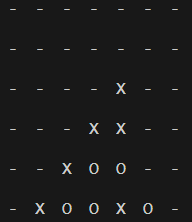
*Pour cela, on utilise cette fois, une boucle « while » dans une autre boucle « for » (pour la fonction\_vertical et fonction\_horizontal)*

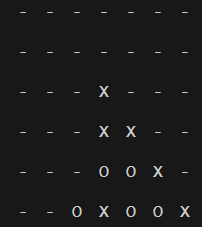
*Pour la fonction\_dia, on utilisera deux boucle « while » afin d’imiter la boucle « for ». A cause des problèmes d’intervalle de la boucle « for »*

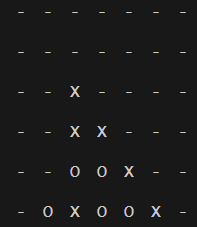
**

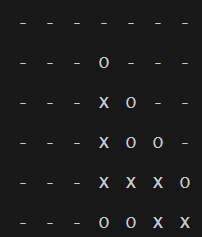
**

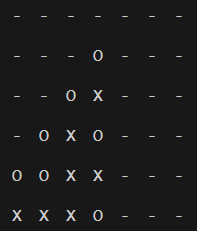
**

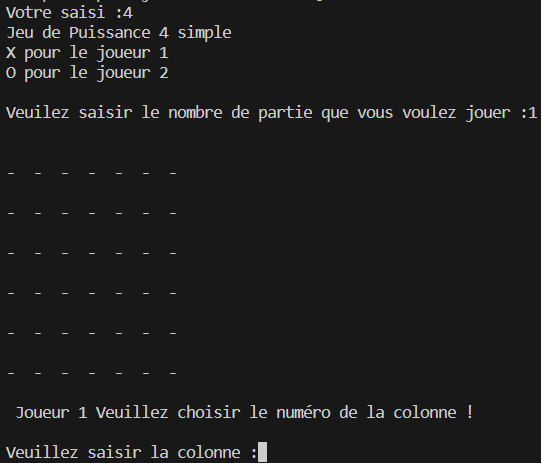
**

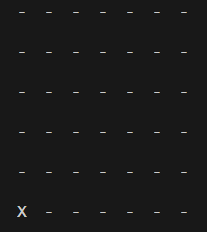
**

**

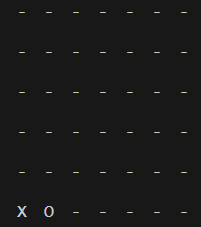
**

**

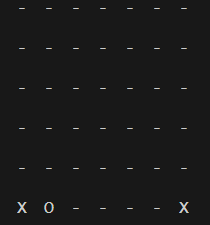
*Jeux d’essai :*

*Une image contenant texte, Police, capture d’écran

Description générée automatiquement*

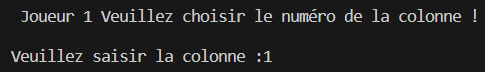
*Une image contenant texte, Police, capture d’écran, typographie

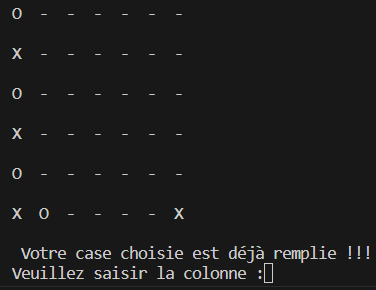
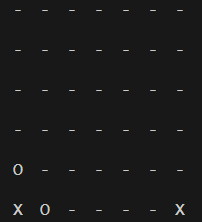
Description générée automatiquement*

**

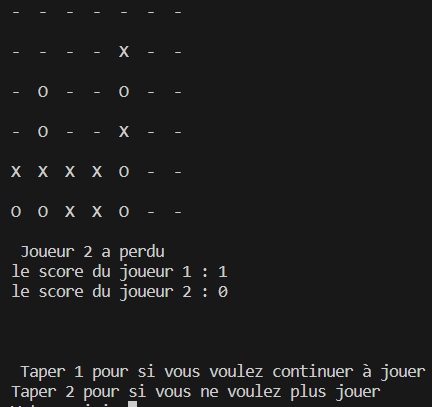
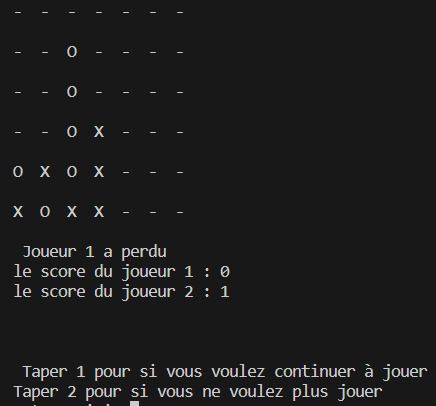
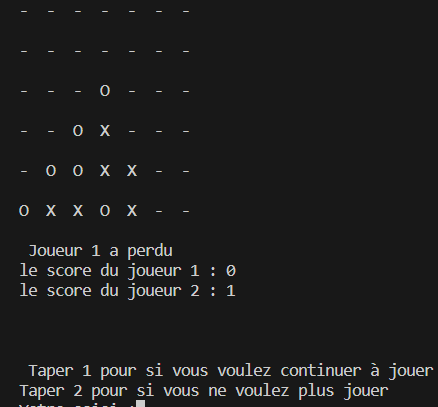
*Si on essaye avec 8*

*Si on essaye avec une colonne déjà remplie*

**

**

*3 manières de gagner :*

**

*Imaginons que le joueur 1 a gagné deux fois et le joueur 2 a gagné une fois. Et que ce soit la fin du programme :*

*Une image contenant texte, capture d’écran, Police

Description générée automatiquement*