**Polytechnique Montréal**

Département de génie informatique et de génie logiciel

INF1005D - Hiver 2022

**Travail pratique nº5**

FONCTIONS ET FICHIERS

**Date limite de remise :** Lundi 4 avril, avant 23h30

**Répartition des points du TP (20 points)**

|  |  |
| --- | --- |
| **EXERCICE** | **POINTS** |
| Exercice 1 | 10 points |
| Exercice 2 | 10 points |
| Boni | 1 point |

# Objectifs du TP

Appliquer concrètement les connaissances des fonctions et de la manipulation de fichiers (.txt, .json et .csv) en Python pour résoudre des mandats d’obfuscation et planifier la retraite d’un directeur affluent.

# Durée

2 séances (6 heures en classe)

# Fichier à remettre

Un fichier ZIP nommé matricule1\_matricule2.zip contenant tous les fichiers .py Assurez-vous qu’à l’ouverture de l’archive, on retrouve directement les fichiers sans passer par aucun dossier.

Exercice 1 – Dans la brume

Étant expert en sécurité informatique, la firme Polytech[[1]](#footnote-1) vous engage en tant que consultant pour déceler l’impact des plus récentes attaques sur leur serveur Jira. Ingénieur en devenir, vous usez de vos plus récentes connaissances *pythoniques* en manipulation de fonctions et de fichiers pour trouver la cible terminale des cybercriminels.

Dans leur plus récent rapport, Polytech a affirmé que les pirates n’ont pas pu effacer complètement les traces de leur méfait. À vrai dire, les administrateurs réseaux ont pu intercepter un des programmes utilisés par les pirates pendant leur attaque, qui vous est remis sous le nom de programme-brumeux.txt.

Néanmoins, les pirates ont prévu qu’une telle fuite pourrait survenir. Ainsi, le code source a été [obfusqué](https://en.wikipedia.org/wiki/Obfuscation) ; toutes les fonctions, structures de contrôles et noms de variables ont été substitués et obscurcis par des caractères incompréhensibles. Heureusement, grâce à une analyse de fréquence, l’équipe Polytech a pu dresser ce tableau de correspondance, dont la majorité des concordances figurent en assembleur et en [ASCII](https://www.alpharithms.com/ascii-table-512119/) :

**Tableau 1 : équivalences compréhensibles des structures de contrôles et divers caractères obfusqués.**

|  |  |
| --- | --- |
| **Code obfusqué** | **Équivalence en Python** |
| \_ZN4crtl1iC1Ev | if |
| \_ZN4crtl1eC1Ev | else |
| \_ZN4crtl1wC1Ev | while |
| \_ZN4crtl1fC1Ev | for |
| \_ZN4crtl1rC1Ev | range |
| \_ZN4crtl1tC1Ev | try |
| \_ZN4crtl2exC1Ev | except |
| \_ZN4util3impC1Ev | import |
| //[0x28] | ( |
| //[0x23] | # |
| //[0x29] | ) |
| //[0x2e] | . |
| //[0x34] | “ |
| //[0x27] | ‘ |
| //[0x20] | ESPACE |
| //[0x57] | / |
| //[0x3d] | = |
| //[0x3a] | : |
| //[0x3b] | ; |
| //[0x3c] | < |
| //[0x3e] | > |
| //[0x5f] | \_ |
| //[0x2d] | - |
| //[0x5c] | \ |
| //[0x2b] | + |
| //[0x2c] | , |
| //[0x61] | a |
| //[0x62] | b |
| //[0x63] | c |
| //[0x64] | d |
| //[0x65] | e |
| //[0x66] | f |
| //[0x67] | g |
| //[0x68] | h |
| //[0x69] | i |
| //[0x6a] | j |
| //[0x6b] | k |
| //[0x6c] | l |
| //[0x6d] | m |
| //[0x6e] | n |
| //[0x6f] | o |
| //[0x70] | p |
| //[0x71] | q |
| //[0x72] | r |
| //[0x73] | s |
| //[0x74] | t |
| //[0x75] | u |
| //[0x76] | v |
| //[0x77] | w |
| //[0x78] | x |
| //[0x79] | y |
| //[0x7a] | z |
| //[0x41] | A |
| //[0x42] | B |
| //[0x43] | C |
| //[0x44] | D |
| //[0x45] | E |
| //[0x46] | F |
| //[0x47] | G |
| //[0x48] | H |
| //[0x49] | I |
| //[0x4a] | J |
| //[0x4b] | K |
| //[0x4c] | L |
| //[0x4d] | M |
| //[0x4e] | N |
| //[0x4f] | O |
| //[0x50] | P |
| //[0x51] | Q |
| //[0x52] | R |
| //[0x53] | S |
| //[0x54] | T |
| //[0x55] | U |
| //[0x56] | V |
| //[0x57] | W |
| //[0x58] | X |
| //[0x59] | Y |
| //[0x5a] | Z |
| //[0x7f] | CODE MORT (À REMPLACER PAR LA CHAÎNE VIDE) |
| //[0x00] | CODE MORT (À REMPLACER PAR LA CHAÎNE VIDE) |
| //[0x07] | CODE MORT (À REMPLACER PAR LA CHAÎNE VIDE) |
| //[0x1e] | CODE MORT (À REMPLACER PAR LA CHAÎNE VIDE) |
| //[0x30] | 0 |
| //[0x31] | 1 |
| //[0x32] | 2 |
| //[0x33] | 3 |
| //[0x34] | 4 |
| //[0x35] | 5 |
| //[0x36] | 6 |
| //[0x37] | 7 |
| //[0x38] | 8 |
| //[0x39] | 9 |
| //[0x7b] | { |
| //[0x7d] | } |
| //[0x5b] | [ |
| //[0x5c] | ] |
| //[0x09] | \n |
| //[0x2a] | \* |

Partie 1 du mandat :

Convertir le fichier programme-brumeux.txt en fichier compréhensible pour y comprendre le fonctionnement derrière l’attaque des cybercriminels. Votre conversion devrait créer un second fichier, nommé programme-decouvert.txt, qui contiendra le code transpilé à partir du Tableau 1. Votre programme de transpilation (à édifier dans exercice1.py) devrait fonctionner pour tous les autres virus qui seront potentiellement découverts par Polytech, et non seulement pour le programme-brumeux.txt. Laissez-vous guider par les instructions incluses au sein du fichier exercice1.py pour arriver à vos fins.

**Boni (1 point) :** écrivez une description précise de l’exécution du programme utilisé par les pirates lors de l’attaque , selon le code transpilé à l'exercice 1. Cette description devrait figurer au sein du fichier boni.txt.

Partie 2 du mandat :

La firme Polytech en a assez de se faire larguer des virus par des cybercriminels ; elle décide ainsi de contre-attaquer ! Quoi de mieux que de combattre le feu par le feu ! Vous devez développer une fonction qui permet d’obfusquer n’importe quel programme de la firme dans le but de combattre les malfaiteurs. Laissez-vous guider par les instructions incluses au sein du fichier exercice1.py pour arriver à vos fins. Dans le cadre de cet exercice, vous devrez obfusquer le programme-a-obfusquer.txt au sein du fichier programme-obfusque.txt.

Barème pour l’exercice 1

|  |  |
| --- | --- |
| **Fonctions à compléter de l’exercice 1** | **Pondération (/10 points)** |
| ● charger\_contenu | 1 |
| ● creer\_dictionnaire | 3  (1,5/3 pour l’ingéniosité) |
| ● transpilation | 3 |
| ● obfuscation | 3 |
| **Total** | **10 points** |

|  |  |
| --- | --- |
| **Difficulté** | **Légende** |
| facile | ● |
| moyen | ● |
| difficile | ● |

Exercice 2 – Hommage à Arruda

Vous êtes mandaté par le ministère de la Santé et des Services sociaux du Québec[[2]](#footnote-2) pour planifier le départ à la retraite du Dr Arruda. L’équipe ministérielle vous mandate avec le choix de la destination de voyage. Toutefois, vous savez bel et bien que le Dr Arruda redoute les pays non vaccinés.

Le fichier vaccinations.csv présente des données mondiales sur les vaccins administrés contre la COVID-19 à travers le monde entier :

**Tableau 2 : données présentées au sein du fichier vaccinations.csv.**

|  |  |  |
| --- | --- | --- |
| **Index** | **Nom des colonnes** | **Descriptif français** |
| 0 | location | pays de la récolte de données |
| 1 | iso\_code | norme ISO 3166-1 du pays |
| 2 | date | date de la récolte de données |
| 3 | total\_vaccinations | nombre total de vaccinations |
| 4 | people\_vaccinated | nombre total de personnes vaccinées |
| 5 | people\_fully\_vaccinated | nombre total de personnes complètement vaccinées (ayant reçu **2** doses) |
| 6 | total\_boosters | nombre total de vaccins en extra administrés (hors période de vaccination) |
| 7 | daily\_vaccinations\_raw | nombre quotidien brut de vaccinations (donnée parfois absente) |
| 8 | daily\_vaccinations | nombre quotidien vaccinations |
| 9 | total\_vaccinations\_per\_hundred | nombre total de vaccinations (en %) |
| 10 | people\_vaccinated\_per\_hundred | nombre total de personnes vaccinées (en %) |
| 11 | people\_fully\_vaccinated\_per\_hundred | nombre total de personnes complètement vaccinées (ayant reçu **2** doses) (en %) |
| 12 | total\_boosters\_per\_hundred | nombre total de vaccins en extra administrés (hors période de vaccination) (en %) |
| 13 | daily\_vaccinations\_per\_million | nombre total de vaccinations quotidiennes par million d’habitants |
| 14 | daily\_people\_vaccinated | nombre total de personnes vaccinées quotidiennement |
| 15 | daily\_people\_vaccinated\_per\_hundred | nombre total de personnes vaccinées quotidiennement (en %) |

Partie 1 du mandat :

Trouver tous les pays potentiels de la destination-retraite du docteur à partir de la récolte de données. Pour ce faire, le gouvernement juge qu’un pays détenant une population dont **au moins 75%** des habitants sont vaccinés peut être considéré comme destination-retraite[[3]](#footnote-3). Pour arriver à vos fins, laissez-vous guider par les instructions incluses au sein du fichier exercice2.py. Entre autres, vous devez charger les données de vaccinations, estimer le nombre total d’habitants du pays considéré, trouver le pourcentage de la population vaccinée et le comparer avec le seuil de 75%, puis terminer en faisant un rapport en format json au sein du fichier rapport.json. Les détails de votre rapport suivent.

Partie 2 du mandat :

Votre rapport (rapport.json) doit arborer cette nomenclature précise :

{ “<pays\_considere\_comme\_destination\_retraite>”: {

“pourcentage\_population\_vaccinee”: “<10.00>”,

“nombre\_habitants”: “<12345678>”,

“nombre\_personnes\_vaccinees”: “<1234567>”

}

}, { “<autre\_pays\_considere\_comme\_destination\_retraite>”:

…

}

…

Les noms entre chevrons doivent être remplacés par vos valeurs obtenues suite à l’analyse faite à la partie 1 du mandat. Un exemple de rapport attendu vous est fourni au sein de exemple-rapport.json.

Barème pour l’exercice 2

|  |  |
| --- | --- |
| **Fonctions à compléter de l’exercice 1** | **Pondération (/10 points)** |
| ● charger\_donnees | 1 |
| ● donnees\_pays | 1 |
| ● nombre\_habitants\_maximal | 2 |
| ● pourcentage\_population\_vaccinee | 3 |
| ● destinations\_retraite | 3 |
| **Total** | **10 points** |

|  |  |
| --- | --- |
| **Difficulté** | **Légende** |
| facile | ● |
| moyen | ● |
| difficile | ● |

|  |  |
| --- | --- |
| **Critère** | **Valeur** |
| Le code répond aux exigences de l’énoncé et s’exécute sans erreurs | 0,25 point |
| Les tests unitaires fonctionnent | 0,50 point |
| Les variables sont bien nommées et il y a des commentaires lorsque nécessaire. | 0,25 point |
| Total | 1 point |

Critères de correction de chaque exercice :

Si un exercice vaut deux points, le total sera multiplié par 2, et ainsi de suite.

1. Bien que fictive, est inspirée de faits réels. [↑](#footnote-ref-1)
2. Fictif. [↑](#footnote-ref-2)
3. *Ibid.* [↑](#footnote-ref-3)