

CRC
**COMPÉTITION DE
PROGRAMMATION**



**PROBLÈME
PRÉLIMINAIRE 3**

QUELQUES NOTES

- Les règles complètes sont dans la section 4 du livret des règlements.
- Vous avez jusqu'au **dimanche 15 décembre, 23h59** pour remettre votre code.
- N'hésitez pas à utiliser le forum de programmation sur le discord de la CRC pour poser vos questions et discuter des problèmes. Il est là pour ça!
- **On vous donne des fichiers modèles faciles à utiliser pour votre code et pour faire vos tests. Vous devez les utiliser pour résoudre le problème!**

UTILISATION DU FICHIER MODÈLE

- Le fichier de test appelle la fonction associée avec en paramètre les informations du test et compare sa sortie avec ce qui est attendu pour vous permettre de voir si les tests réussissent. **Tout votre code (sauf fonctions additionnelles que vous créez) devrait être écrit dans la fonction prévue à cet effet.**
- Les points mis dans le document indiquent la difficulté et le pointage attribué pour la réussite pour chaque défi. Ce problème préliminaire aura une valeur globale de 2% du défi principal.

STRUCTURE

Une petite mise en situation comme celle-ci explique les fondements de chaque défi et offre les bases nécessaires pour résoudre celui-ci.

Spécification d'entrée et de sortie:

Contient les caractéristiques des entrées fournies ainsi que les critères attendus pour les sorties du programme.

Exemple d'entrée et de sortie:

Contient un exemple d'entrée, parfois constitué lui-même de plusieurs sous-exemples, pour que vous puissiez tester votre programme. Chaque exemple de sortie donne la réponse attendue pour l'entrée correspondante.

Explication de la première sortie:

Décortique davantage le défi en expliquant comment la première entrée est traitée et en montrant le chemin menant à cette réponse.

Hôpital CRC

Un des aspects les plus stressants de la CRC est de faire des réparations rapides sur le robot dans les quelques minutes avant le prochain match. En qualité d'entrepreneur que vous êtes, vous décidez de fonder l'hôpital CRC. Le tout premier hôpital dédié aux robots!

Faire fonctionner un hôpital vient bien sûr avec son lot de défis et heureusement pour vous, certains d'entre eux peuvent être résolus à l'aide de la programmation. C'est grâce à l'utilisation de la programmation dans votre hôpital moderne que vous pourrez être le premier établissement dédié à la santé et la réparation de robots.

Partie 1: Traits récessifs (40 points)

L'ensemble des gènes d'un organisme constitue son génome. Un gène est un morceau d'ADN qui correspond à un caractère particulier de l'organisme, par exemple la couleur de ses yeux. Chaque gène est présent en double : il possède deux copies, chacune transmise par un parent, qui s'appellent des **allèles**.

Il existe des allèles dominants et récessifs. Un allèle **dominant** masque l'expression de l'autre allèle (c'est-à-dire qu'un allèle dominant s'exprime s'il est présent en une seule copie). Un allèle **récessif** ne s'exprime que s'il est présent en deux copies. On représente un allèle dominant par une lettre majuscule et un allèle récessif par la même lettre en minuscule.

Prenons l'exemple de la couleur de nos yeux. L'allèle brun (B) est dominant vis-à-vis l'allèle bleu (b), alors si une personne possède un allèle brun et un allèle bleu (Bb), elle aura des yeux bruns.

Disons maintenant qu'un des parents possède les allèles Bb et l'autre bb. On peut prédire les gènes résultant d'un croisement entre eux à l'aide d'un **échiquier de Punnett** (ou grille de Punnett, aussi appelé *Punnett square* en anglais):

| | | |
|---|----|----|
| | b | b |
| B | Bb | Bb |
| b | bb | bb |

Leurs enfants auront donc une chance sur deux d'hériter des yeux bleus!

Dans ce problème, vous allez analyser des arbres généalogiques (à l'envers) et des maladies récessives afin de calculer la probabilité P qu'un descendant hérite d'une certaine maladie.

Voici quelques formules pour calculer des probabilités (pour en apprendre plus, [consultez cet article](#)):

$$P_{\text{récessif Parent}} = P_{\text{hériter XX}} \cdot P_{\text{transmettre allèle récessif}} \quad (1)$$

$$P_{\text{récessif}} = P_{\text{récessif Parent 1}} \cdot P_{\text{récessif Parent 2}} \quad (2)$$

Spécification d'entrée et de sortie:

En entrée vous recevrez:

- **tree:** un arbre binaire parfait, dont chaque nœud contient deux lettres qui correspondent aux allèles et les allèles indiqués par "XX" sont inconnus. La racine (elle sera toujours "XX") représente la personne qu'on est en train d'analyser. Les nœuds enfants de la racine sont les parents de cette personne, puis les nœuds enfants des parents sont les grand-parents et ainsi de suite.

En sortie vous devrez fournir:

- une variable *float* avec 3 décimales de précision, correspondant à la probabilité en pourcentage que le nœud racine hérite de la maladie récessive.

Exemple d'entrée:

```
["XX", "XX", "Cc", "Cc", "Cc", "CC", "Cc"]
```

```
["XX", "XX", "XX", "Ss", "ss", "Ss", "SS"]
```

```
["XX", "XX", "XX", "XX", "Tt", "tt", "XX", "TT", "TT", "Tt",  
"Tt",  
"Tt", "XX", "TT", "tt"]
```

Exemple de sortie:

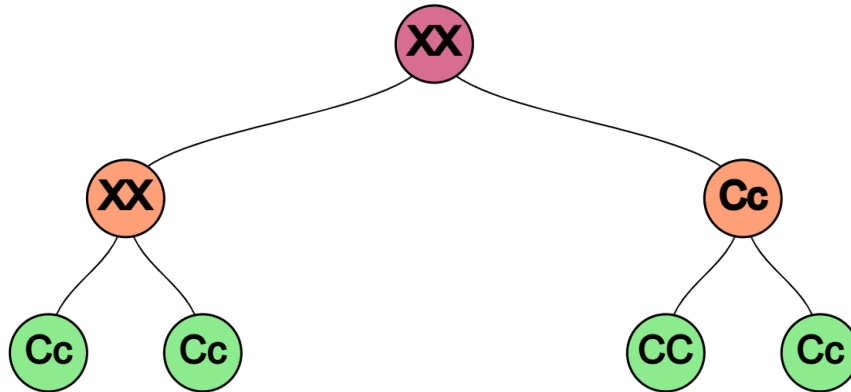
25.0

18.75

18.75

Explication de la première sortie:

Voici l'arbre binaire qu'il faut analyser:



On peut créer un échiquier de Punnett pour trouver les gènes possibles du nœud orange inconnu à gauche (dont les allèles sont "XX"):

| | | |
|---|----|----|
| | C | c |
| C | CC | Cc |
| c | Cc | cc |

Méthode 1: échiquiers de Punnett

Le parent inconnu "XX" pourrait donc avoir les allèles CC, Cc, Cc, ou cc. Voici les 4 échiquiers de Punnett correspondant aux allèles possibles du nœud racine:

| | | |
|---|----|----|
| | C | C |
| C | CC | CC |
| c | Cc | Cc |
| 1 | | |

| | | |
|---|----|----|
| | C | c |
| C | CC | Cc |
| c | Cc | cc |
| 2 | | |

| | | |
|---|----|----|
| | C | c |
| C | CC | Cc |
| c | Cc | cc |
| 3 | | |

| | | |
|---|----|----|
| | c | c |
| C | Cc | Cc |
| c | cc | cc |
| 4 | | |

La racine héritera de la maladie récessive si elle possède la paire d'allèles **cc**, qui correspond à 4 sur 16 des cases. La probabilité est donc de $\frac{4}{16} = 25.0\%$.

Méthode 2: formules

Puisqu'on ne peut pas hériter un allèle récessif *c* de CC, il suffit de considérer les deux autres possibilités, Cc et cc, pour le parent 1 à gauche. On sait déjà que les allèles du 2e parent à droite sont Cc. Il y a donc deux combinaisons possibles: si les parents sont Cc et Cc, ou s'ils sont cc et Cc.

La probabilité d'hériter de la maladie récessive sera la somme de P_1 et P_2 , où P_1 correspond à la probabilité d'hériter de la maladie d'un croisement entre Cc et Cc, et P_2 à la probabilité pour un croisement entre cc et Cc.

Examinons le croisement entre Cc et Cc. On veut calculer la probabilité que chacun des parents transmet un allèle *c*. La formule (1) nous donne:

$$P_{\text{récessif Parent 1}} = P_{\text{récessif Cc (gauche)}} = P_{\text{hériter Cc}} \cdot P_{\text{transmettre allèle c}} = \frac{2}{4} \cdot \frac{1}{2} = \frac{1}{4}$$

(Il y a 2 chances sur 4 que le parent 1 possède Cc, puis 1 chance sur 2 qu'on transmet l'allèle récessif *c* et non l'allèle dominant C.)

$$P_{\text{récessif Parent 2}} = P_{\text{récessif Cc (droite)}} = P_{\text{hériter Cc}} \cdot P_{\text{transmettre allèle c}} = 1 \cdot \frac{1}{2} = \frac{1}{2}$$

(Le parent 2 possède les allèles Cc. Il y a donc 1 chance sur 2 qu'il transmet à la racine l'allèle récessif *c*.)

Ensuite, on utilise la formule (2):

$$P_1 = P_{\text{récessif Parent 1}} \cdot P_{\text{récessif Parent 2}} = \frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$$

Examinons de la même façon le croisement entre cc et Cc:

$$P_{\text{récessif Parent 1}} = P_{\text{récessif cc (gauche)}} = P_{\text{hériter cc}} \cdot P_{\text{transmettre allèle c}} = \frac{1}{4} \cdot \frac{2}{2} = \frac{1}{4}$$

$$P_{\text{récessif Parent 2}} = P_{\text{récessif Cc (droite)}} = P_{\text{hériter Cc}} \cdot P_{\text{transmettre allèle c}} = 1 \cdot \frac{1}{2} = \frac{1}{2}$$

$$P_2 = P_{\text{récessif Parent 1}} \cdot P_{\text{récessif Parent 2}} = \frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$$

Enfin, selon la règle de la somme, la probabilité que le nœud racine hérite de la maladie récessive est la somme de P_1 et P_2 :

$$P = P_1 + P_2 = \frac{1}{8} + \frac{1}{8} = 0.25 = 25.0\%$$

Partie 2: Files d'attentes (20 points)

Vous avez dans votre hôpital un nombre limité de médecin réparateur et vous essayez de savoir selon votre nombre de clients et le temps moyen de réparation combien de temps vos clients vont attendre en moyenne. Vous devrez donc calculer le temps d'attente moyen. Pour vous aider, voici quelques explications et formules utiles sur la théorie des files d'attente.

λ = nombre de clients qui arrivent par heure

μ = nombre de clients qui peuvent être traités par heure par un médecin réparateur

c = nombre de médecins réparateurs

ρ = ratio des arrivées par rapport aux traitements

$$\rho = \frac{\lambda}{c \cdot \mu}$$

Le ratio d'arrivée par rapport aux traitements est une donnée vraiment importante. Si la valeur du ratio est plus grande que 1, cela veut dire que le système n'est pas stable et les arrivées dépassent les traitements. Ainsi, les temps de traitement deviennent infiniment longs plus le temps avance les clients s'accumulent. Dans nos exemples, le ratio sera toujours inférieur à 1.

L = nombre de clients dans l'hôpital

L_q = nombre de clients dans la file d'attente

$P_{full}(\rho, c)$ = Probabilité que tous les médecins réparateurs soient occupés

La probabilité P_{full} est un dictionnaire de valeurs qui dépendent du ratio ρ et du nombre de médecins réparateurs. Vous aurez le dictionnaire de fourni, il vous faudra par contre aller chercher la bonne valeur selon le ρ et le c de la situation. **La table commence à 2 médecins réparateurs et va jusqu'à 7.** On a comme clé de valeurs du dictionnaire la valeur du ratio ρ et tous les exemples données auront un ratio ρ qui est dans la table que nous vous avons donnée. Une fois que vous avez l'array associé au bon ratio, vous avez les probabilités que tous les médecins réparateurs soient occupés en allant de 2 à 7 médecins réparateurs.

```
proba_full = {  
    .10: [.02, .00, .00, .00, .00, .00],  
    .20: [.07, .02, .00, .00, .00, .00],  
    ...  
}
```

$$L_q = P_{full}(\rho, c) \cdot \frac{\rho}{1-\rho}$$

$$L = \frac{\lambda}{\mu} + L_q$$

W = temps moyen passé pour attendre et être traité

$$W = \frac{L}{\lambda}$$

Spécification d'entrée et de sortie:

En entrée vous recevrez:

- arrivals: un *int* du nombre de clients qui arrivent par heure
- mu: un *int* du nombre de clients qu'un médecin réparateur peut traiter par heure
- c: un *int* du nombre de médecin réparateur disponible

En sortie vous devrez donner une variable *float* avec 2 décimales de précision du temps moyen d'attente **en minutes**.

Exemple d'entrée:

arrivals = 80, mu = 50, c = 2

arrivals = 42, mu = 8, c = 7

arrivals = 6, mu = 3, c = 4

Exemple de sortie:

3.33

9.17

21.7

Explication de la première sortie:

Dans le premier exemple, vous avez 80 robots qui arrivent par heure et 2 médecins réparateurs qui travaillent. Chacun des médecins réparateurs peut traiter 50 robots par heure. Nous commençons par trouver le ratio des arrivés et des traitements.

$$\rho = \frac{\lambda}{c \cdot \mu}$$

$$\rho = \frac{80}{2 \cdot 50}$$

$$\rho = 0.8$$

Nous avons donc notre ratio qui va nous permettre d'aller chercher dans notre dictionnaire la probabilité d'avoir les deux médecins réparateurs occupés en même temps.

$$P_{full}(0.8, 2) = 0.71$$

On peut maintenant calculer le nombre de patient en attente par heure L_q et le nombre de patients dans l'hôpital L par heure.

$$L_q = P_{full}(\rho, c) \cdot \frac{\rho}{1-\rho}$$

$$L_q = P_{full}(0.8, 2) \cdot \frac{0.8}{1-0.8}$$

$$L_q = 2.84$$

$$L = \frac{\lambda}{\mu} + L_q$$

$$L = \frac{80}{50} + 2.84$$

$$L = 4.44$$

Finalement, ce que l'on veut trouver c'est le temps d'attentes en minutes donc on trouve le temps d'attente en heures qu'on peut finalement convertir en minutes.

$$W = \frac{L}{\lambda}$$

$$W = \frac{4.44}{80}$$

$$W = 0.555 \text{ heures}$$

$$W = 3.33 \text{ minutes}$$

Partie 3: Moniteur cardiaque (40 points)

Durant les opérations sur les robots, pour vous assurer du bon fonctionnement du circuit électrique, vous voulez implémenter un moniteur cardiaque. Votre moniteur cardiaque va montrer les battements de cœur fait par la batterie pour 4 battements.

Vous recevrez le nombre systolique, le nombre diastolique et la fréquence de battement pour générer votre visuel du moniteur cardiaque. Nous allons faire un visuel simplifié avec ces valeurs qui va représenter les battement électrique des batteries du robot (Oui votre robot va utiliser du AC dans cette situation!). La valeur systolique vous dira à quelle hauteur les pics vont monter alors que la valeur diastolique vous dira à quel point les pics vont descendre. Finalement, la fréquence de battement va vous indiquer l'espacement entre les différents battements.

Chaque tranche de 20 sera l'augmentation d'une ligne par rapport à la neutralité dans l'affichage du pic pour la systolique et chaque tranche de 20 sera une diminution d'une ligne par rapport à la neutralité. Ainsi, un nombre systolique de 95 est une augmentation de 4 lignes et un nombre systolique de 100 est une augmentation de 5 lignes par rapport à la neutralité. On a une logique similaire pour le nombre diastolique de 77 qui serait une diminution de 3 et un nombre diastolique de 85 est une diminution de 4 lignes. Dans un battement, vous partirez de la neutralité et monterez du nombre de lignes selon le nombre systolique, une plateforme en haut puis vous redescendez à la neutralité, une plateforme, la descente selon le nombre diastolique, une plateforme et finalement vous remontez à la neutralité qui va être conservée selon la fréquence cardiaque.

Pour la fréquence cardiaque, chaque tranche de 10 de moins que la valeur 180 sera une ligne de plus à la valeur de neutralité entre deux battements. Donc, une fréquence de 135 donne 4 et une fréquence de 142 donne 3 temps à la valeur neutre.

Vous devrez afficher 4 battements de cœur du robot selon le nombre systolique, le nombre diastolique et la fréquence cardiaque.

Spécification d'entrée et de sortie:

En entrée, vous recevrez

- `systolic`: un `int` qui représente le nombre systolique
- `diastolic`: un `int` qui représente le nombre diastolique
- `heart_rate`: un `int` qui représente la fréquence cardiaque

En sortie, vous donnerez un `array` de `string` qui est le visuel de 4 battements

Exemples d'entrée:

systolic: 120, diastolic: 90, heart_rate: 120

systolic: 105, diastolic: 77, heart_rate: 105

systolic: 100, diastolic: 92, heart_rate: 135

Exemples de sortie:

[

]

[

The diagram illustrates a sequence of four identical rectangular blocks arranged horizontally. Each block is constructed from a 3x3 grid of vertical lines. The top row of each block is capped with a horizontal line, and the bottom row is also capped with a horizontal line. The blocks are connected by horizontal lines between their right and left sides, forming a continuous structure.

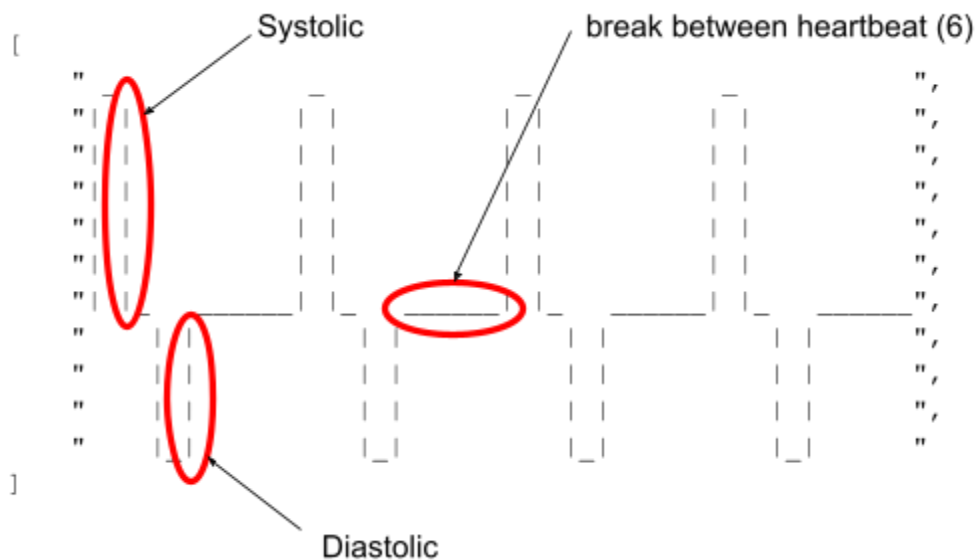
]

[

]

Explication de la première sortie:

Dans le premier exemple, nous avons un nombre systolique de 120. Comme nous opérons par tranche de 20 sur l'affichage, 120 correspond à $120 / 20$ donc 6. Dans la réponse, nous pouvons voir que les battement ont une hauteur de 6 indiqué sur l'image par la flèche Systolic. Pour le nombre diastolic, on a 90 qui lorsqu'on divise par 20 on obtient 4.5 qui lorsqu'on arrondi vers le bas nous donne 4. On peut voir que nous avons un déplacement de 4 vers le bas dans la portion Diastolic de l'image. Pour le temps d'attente entre les battements, nous prenons la fréquence cardiaque qui est de 120. Nous pouvons faire $180 - 120$ qui nous donne 60. Cette fois nous utilisons des tranches de 10. Ainsi nous avons $60 / 10$ qui nous donne 6 temps de pause avant le prochain battement. On répète la forme du battement pour 4 battement pour avoir la réponse finale.



Partie 4: Algorithme de diagnostic (40 points)

Entre les prédispositions génétiques, les conditions affectant déjà le patient, les effets secondaires causés par de la médication, et tellement d'autres facteurs externes comme l'environnement du patient, le diagnostic médical est un travail assez difficile, surtout lorsqu'il s'agit de maladies plus rares. En se débarrassant de plusieurs facteurs plus complexes comme l'âge et les prédispositions génétiques, voyons voir s'il est possible de construire la base d'un algorithme qui pourrait diagnostiquer un patient à partir d'une simple série d'informations et de bases de données.

Votre programme aura accès à deux bases de données assez petites pour construire des bases solides. La première s'appelle *diagnostics*, et contient le nom et les symptômes des maladies que vous devrez diagnostiquer. La deuxième s'appelle *médicaments*, et contient quelques prescriptions communes pouvant concerner vos patients ainsi qu'une liste des principaux effets secondaires associés à chaque prescription.

Pour un patient, vous aurez la liste de ses symptômes apparents ainsi que la liste de ses prescriptions. **Il est possible qu'un patient soit sous aucune prescription, ou qu'il soit sous certaines prescriptions absentes de la base de données.** Dans le dernier cas, il faudra ignorer une telle prescription.

La logique de diagnostic que vous devrez utiliser est assez simple: on suppose qu'une maladie ne montre que des symptômes présents dans sa liste dans la base de données. **Si un symptôme est présenté par le patient, mais n'est pas dans la liste donnée pour une maladie, on assume que ce ne peut raisonnablement pas être cette maladie.** Attention! **Certains symptômes** peuvent se retrouver dans la liste d'**effets indésirables produits par une prescription du patient.** Comme on ne peut prouver si ces symptômes viennent de la condition du patient ou de sa prescription, **ils doivent être ignorés.** Dans cette partie, on considère que chaque patient donné sera affecté **d'une et d'une seule maladie** provenant de la base de données diagnostique.

Spécification d'entrée et de sortie:

En entrée, vous recevrez:

- symptômes: une liste de *string* contenant les symptômes du patient
- prescriptions: une liste de *string* contenant les prescriptions du patient
- diagnostics: une liste en 3D de *string*, soit la base de données diagnostique
- médicaments: une liste en 3D de *string*, soit la base de données des prescriptions

En sortie, vous devez fournir une *string*, soit le diagnostic tel qu'écrit dans la base de données pour ce patient.

Exemple d'entrée:

```
symptomes: ["Fatigue", "Faiblesse", "Toux"], prescriptions:  
["Ibuprofène", "Morphine"], diagnostics, médicaments
```

```
symptomes: ["Pâleur"], prescriptions: [], diagnostics, médicaments
```

```
symptomes: ["Troubles de vision", "Étourdissements", "Maux de tête",  
"Perte de poids"], prescriptions: ["Acétaminophène"], diagnostics,  
médicaments
```

```
symptomes: ["Douleurs abdominales", "Diarrhée", "Constipation",  
"Fatigue", "Dépression", "Troubles d'équilibre", "Fièvre",  
"Somnolence"], prescriptions: ["Antibiotiques", "Fluvoxamine"],  
diagnostics, médicaments
```

```
symptomes: ["Éternuements", "Diarrhée", "Nausées", "Maux de gorge",  
"Congestion nasale", "Fatigue", "Toux"], prescriptions:  
["Antibiotiques", "Diphénhydramine"], diagnostics, médicaments
```

Exemple de sortie:

```
"Tuberculose"  
"Anémie"  
"Diabète"  
"Sclérose en plaques"  
"Rhume"
```

Explication de la première sortie:

Pour voir *diagnostics* et *médicaments*, allez voir les fichiers modèles de test pour cette partie. Dans le premier exemple, on trouve d'abord les prescriptions "Ibuprofène" et "Morphine". Comme *médicaments* ne contient rien pour "Morphine", on l'ignore. Cependant, au niveau de "Ibuprofène", on retrouve les effets secondaires suivants: ["Démangeaisons", "Insomnie", "Nausées", "Vomissements", "Douleurs/Crampes d'estomac", "Diarrhée", "Fatigue", "Indigestion", "Constipation", "Ballonnements"]. Nous devons donc ignorer "Fatigue" dans les symptômes du patient. Il nous reste les symptômes "Faiblesse" et "Toux" et, en cherchant dans *diagnostics*, on trouve que le seul diagnostic exhibant ces deux symptômes est la "Tuberculose": ["Toux", "Douleurs thoraciques", "Faiblesse", "Fatigue", "Perte de poids", "Perte d'appétit", "Frissons", "Fièvre", "Sueurs nocturnes"].

Bonus sur la partie 4

Ce cas-ci est pas mal plus complexe et ne vaudra donc pas de points. Il est cependant très intéressant pour ceux qui désirent pousser leur code de la partie 4 un peu plus loin.

Nous considérons maintenant le cas où le patients présente des symptômes qui ne sont pas tous présents dans une seule condition de la base de données. Dans ce cas-ci, on jugera que le cas le plus probable est que le patient souffre de deux conditions dans la base de données. Ainsi, si un patient est affecté par deux conditions, n'importe laquelle de celles-ci peut contribuer à la liste des symptômes et il suffit donc de trouver chaque symptômes dans la liste d'au moins une des deux conditions.

Voici un exemple de symptômes: ["Faiblesse", "Toux", "Fatigue", "Écoulement nasal", "Perte d'appétit", "Maux de tête", "Frissons", "Fièvre", "Étourdissements"]. À prime abord, il pourrait s'agir de la tuberculose, mais on se rend compte que l'écoulement nasal, les maux de tête et les étourdissements sont de trop. On pourrait croire que c'est la tuberculose et une autre condition, mais on se rend bien compte qu'aucune condition en réunit ces trois symptômes manquants. En cherchant d'autres combinaisons possibles, on se rend bien compte que la grippe a comme symptômes la fatigue, la toux, l'écoulement nasal, la fièvre, les frissons, la perte d'appétit et les maux de tête. L'anémie a les deux symptômes restants, soit la faiblesse et les étourdissements en plus de d'autres symptômes en commun avec la grippe. Dans ce cas-ci, il s'agirait donc d'une grippe et de l'anémie. La logique de votre code doit être pas mal plus robuste pour prendre en compte de telles possibilités!