

## Compte rendu de l'entretien du 22/01/2018

Un bon manager est une personne qui sait écouter et échanger avec son équipe. Il parvient alors à exploiter les meilleures compétences de chacun des membres pour rendre un résultat correct en un minimum de temps. Par ailleurs, en cas de conflit, il arrive à les gérer en discutant avec les membres de l'équipe.

Ainsi, c'est aussi de cette façon que j'aimerais être managé. Je saurais alors sur qui m'appuyer en cas de question ou de problème. Le manager est avant tout humain et sait s'adapter à tout à chacun. Je crois que la discussion est la clé.

Concernant mes motivations, j'aime apprendre de nouvelles choses. Enrichir ma culture est un point essentiel et c'est ce qui me permet d'avancer dans la vie de tous les jours. La connaissance est aujourd'hui primordiale pour aborder le monde qui nous entoure. C'est donc ma motivation principale.

De ce fait, entrer chez Valwin me permettra d'acquérir des compétences techniques poussées mais aussi de créer de nouvelles relations avec les membres de l'équipe. Evidemment, en plus de la bonne humeur, il me sera possible d'apporter un nouveau point de vue sur un projet en cours, une autre façon de penser voire même des connaissances juridiques du fait de mon activité dans la réserve.

Enfin, entrer dans l'entreprise nécessitera un léger temps d'adaptation d'un point de vue matériel et organisationnel (disposition des locaux) mais aussi comprendre l'articulation des projets déjà existant et qui sont susceptibles d'être améliorés.

La santé est un domaine important qui doit être accessible à tous. C'est la raison pour laquelle je souhaite rejoindre la start-up. Beaucoup de personnes ne prennent pas soin d'eux. Cela se ressent dans les catégories sociales plutôt populaires. Dès lors, faciliter cet accès aux soins permet à tout le monde d'être en bonne santé. A hauteur de mes compétences, j'aimerais pouvoir contribuer à cette avancée.

En parlant de start-up, je crois qu'il y a une différence entre celle-ci et une entreprise classique. En effet, au sein d'une entreprise classique, on y trouve le patron qui va avoir un travail de gestion de ses salariés et qui va déléguer ses pouvoirs à ses subordonnés. On peut alors retrouver une hiérarchie sous forme d'arbres. A contrario, la start-up se veut innovante et chacun travaille sur une partie d'un projet. Il y a évidemment une hiérarchie mais elle reste moins marquée que dans une entreprise classique. Enfin, la start-up sera plus petite qu'une entreprise. On pourra trouver une dizaine de personnes pour une start-up contre plus d'une centaine pour une entreprise classique.

Aimant que le travail soit bien fait, j'ai tendance à vouloir diriger afin de m'assurer que chacun d'entre nous travaille selon ses compétences. Au sein de mon groupe de projet, j'essaye à ce que chacun fasse sa part de travail. Pour autant, on ne m'a jamais demandé de faire de concession sur mon intégrité.

A l'issue de mon DUT je compte m'engager au sein de la gendarmerie nationale. Il existe beaucoup de spécialités dont celle de l'informatique (Système d'Information et de Communication). J'aimerais donc allier rêve d'enfant et passion pour l'informatique.

Enfin, si je gagnais 5 millions d'euros, j'en donnerais à ma famille, mes amis, je partirais voyager autour du monde et je m'achèterais une voiture et une maison. Le reste sera placé sur un compte en banque.

Je peux alors affirmer que j'aime les gens.

### ***Exercice : Récursivité***

Plusieurs approches du même exercice ont pu être faites. Le but étant de calculer la somme des valeurs dans une liste d'entier.

On a alors les valeurs suivantes : 2,8,16,32,5

Un premier raisonnement consiste à initialiser une variable somme à laquelle on ajoute toutes les valeurs de la liste dans une boucle.

```
List<Integer> liste = new ArrayList<>();
liste.add(2);
liste.add(8);
liste.add(16);
liste.add(32);
liste.add(5);
int somme = 0;

pour i allant de 0 à liste.size{
    somme += liste.get(i);
}

return somme;
```

Un second raisonnement consiste à utiliser la récursivité. Le principe de la récursivité nécessite une condition d'arrêt sinon on bouclera à l'infini.

Considérons les méthodes suivantes à notre dispositions :

```
ArrayList<Integer> tail() // renvoie la queue
Integer head() // renvoie la tête
boolean isEmpty()
```

Il faut maintenant créer la méthode calculant la somme de la liste

```
List<Integer> sum (ArrayList<Integer> liste){
    si liste.isEmpty(){return 0;}
    sinon {
        int tete = liste.head();
        return tete + sum(liste.tail());
    }
}
```

```
    }  
}
```

**Un troisième raisonnement permet d'éviter le traitement de l'addition dans le return. Notre programme manie trois types : Les entiers, les listes d'entier et les booléens. L'idée, c'est de passer un second paramètre à notre méthode. Le second paramètre sera forcément de type Integer puisque la liste des entiers et un booléen qui vérifie la présence d'un entier est inutile du fait du type de la liste.**

```
private Integer sum(List<Integer> liste, Integer i){  
    si liste.isEmpty(){return i;}  
    sinon{  
        i = i+liste.head();  
        return sum(liste, i);  
    }  
}
```

**Ce raisonnement implique l'initialisation de i. On peut alors créer une méthode qui sera publique à laquelle i est égal à 0 de la façon suivante :**

```
public sum(List<Integer> liste){  
    return sum(liste, 0);  
}
```