

Compte rendu : The River

Lilian Hiault

11 avril 2019

Table des matières

1	The River I	1
1.1	Comment incrémenter un rivière ?	1
1.2	Que faut-il calculer ?	2
2	The River II	3
2.1	Calculer le terme précédent	3
2.2	trouver une deuxième rivière correspondant aux critères	3

Introduction

« The River I » et « The River II » sont deux exercices posés par ng201 sur Coding Games (<https://www.codingame.com/training/easy/the-river-i->).

Une rivière est une suite de nombres dont le terme suivant est lui même auquel on ajoute la somme de chaque chiffre qui le compose. Par exemple, si on prend la rivière 123, on a le terme suivant égal à $123 + 1 + 2 + 3 = 129$.

1 The River I

Dans ce premier exercice, on nous donne deux rivières et il faut trouver à quel nombre les deux rivières se retrouvent.

1.1 Comment incrémenter un rivière ?

Pour trouver ce point de rencontre j'ai créé deux rivières que j'incrémente jusqu'à ce qu'elles soient égales. J'ai donc créé une fonction pour calculer le terme suivant.

```

long long incrementerRiviere(long long r)
{
    long long riviere = r;
    long long somme = 0;
    while(riviere != 0)
    {
        somme += riviere%10;
        riviere /= 10;
    }
    return r+somme;
}

```

Dans cet exercice, on utilise le type « long long », c'est un type de données qui existe en C99 et qui nécessite 64 bits. Il peut stocker des nombres plus grands que la capacité des « int ». On crée d'abord une copie de la rivière donnée en paramètre pour pouvoir faire la somme plus tard. On va ensuite extraire chaque chiffre pour en faire la somme. On prend le reste de la division euclidienne par 10 ce qui nous donne le chiffre des unités qu'on peut ajouter à notre somme. Une fois fait, on fait une division par 10 d'entiers sur la rivière, on ne garde donc que le quotient. Cela revient à supprimer le chiffre des unités et à décaler le nombre. Le nouveau chiffre des unités correspond à l'ancien chiffre de dizaines. En continuant jusqu'à ce que la rivière soit nulle, on ajoute chaque chiffre puis on retourne la somme de ces chiffres avec la rivière donnée, on a calculé le terme suivant.

1.2 Que faut-il calculer ?

```

int main()
{
    long long r1;
    scanf("%lld", &r1);
    long long r2;
    scanf("%lld", &r2);
    while (r1 != r2)
    {
        while (r1 < r2)
        {
            r1 = incrementerRiviere(r1);
        }
        while (r1 > r2)
        {
            r2 = incrementerRiviere(r2);
        }
    }
    printf("%lld", r1);
    return 0;
}

```

Après avoir récupéré les deux rivières données, il suffit d'incrémenter celle qui est la plus petite jusqu'à ce que les deux soient identiques et afficher alors à quel nombre elles le deviennent.

2 The River II

Ici, on nous donne un nombre et il faut trouver deux rivières qui se rencontrent à cet endroit.

2.1 Calculer le terme précédent

Je garde la même fonction d'incréméntation que dans l'exercice précédent (à ceci près qu'on manipule des entiers cette fois).

```
int incrementerRiviere(int r)
{
    int riviere = r;
    int somme = 0;
    while(riviere != 0)
    {
        somme += riviere%10;
        riviere /= 10;
    }
    return r+somme;
}
```

2.2 trouver une deuxième rivière correspondant aux critères

On récupère l'entier donné par l'utilisateur. On considère cet entier comme une rivière qui commence à ce nombre. Il ne reste plus qu'à trouver une autre rivière qui atteint ce nombre.

```

int main()
{
    int r1;
    scanf("%d", &r1);
    int i = r1-1;
    int trouve = 0;
    while ((trouve==0) && (i>0))
    {
        int r2 = i;
        r2 = incrementerRiviere(r2);
        if (r1==r2)
        {
            trouve = 1;
        }
        i--;
    }
    if (trouve==1)
    {
        printf("YES\n");
    }
    else
    {
        printf("NO\n");
    }
    return 0;
}

```

On teste si le nombre précédant la première rivière k (donnée) retrouve l'intersection à son prochain terme, si c'est le cas, on a trouvé une rivière $k - 1$ qui satisfait les conditions. Sinon, on réessaie avec la rivière $k - 2$ et ainsi de suite. Si aucune rivière n'est trouvée lorsqu'on arrive à 0 alors il n'existe pas deux rivières qui se rencontrent au point donné.

Conclusion

Ces deux problèmes ne m'ont pas posé de problèmes particulier. Le seul que j'ai eu a été mon temps d'exécution du deuxième programme. Au début je re-calculais la seconde rivière jusqu'à ce qu'elle soit plus grande que la première alors qu'une seule itération suffit. En effet si on calcule une deuxième incrémentation de la rivière c'est qu'elle est inférieure à la rivière 1. Or puisqu'on parcourt les solutions dans l'ordre décroissant on a donc déjà calculé le nouveau terme.