NOM :

Quick du 26 octobre 2016 (Durée $\frac{1}{2}$ heure)

Aucun document, ni calculette ou tout objet contenant un processeur n'est autorisé.

Pour tous les exercices, on suppose que l'on dispose d'un générateur aléatoire de nombres entiers **Alea** (n) (resp de nombre réel **Random** ()) à valeur dans $\{0, \cdots, n-1\}$ (resp $[0, 1[$). On suppose qu'une séquence d'appels à **Alea** (.) (resp **Random** ()) se modélise par une suite de variables aléatoires indépendantes, la loi de **Alea** (n) (resp **Random** ()) est supposée uniforme dans $\{0, \cdots, n-1\}$ (resp dans $[0, 1[$).

Problème : mot de passe

En informatique, la plupart des mécanismes assurant la confidentialité reposent sur des mots de passe. Les attaques essayent donc de découvrir les mots de passe en faisant des recherches par dictionnaire, par force brute, par force brute dirigée par des informations sur la façon dont le mot de passe a été construit... Un algorithme de génération de mot de passe (en général public) doit donc assurer un maximum d'imprédictibilité. C'est à dire, qu'a priori, aucun mot de passe ne doit être plus probable qu'un autre.

On souhaite écrire un générateur de mots de passe répondant à ce critère d'imprédictibilité. L'ensemble des caractères disponibles est

$$\mathcal{C} = \{a, b, \cdots, z, 0, 1, \cdots, 9\}.$$

1. Écrire un algorithme de génération uniforme de mot de passe ayant 8 caractères dont 2 ou 3 chiffres.

Solution: La première étape consiste à compter le nombre total de mots de passe. On note $N = 26$ le nombre de lettres $K = 10$ le nombre de chiffres. L'ensemble $\mathcal{M}$ des mots de passe admissibles est l'union de 2 ensembles disjoints : les mots de passe ayant exactement 2 chiffres $\mathcal{M}_2$ et ceux ayant exactement 3 chiffres $\mathcal{M}_3$.

Pour les mots de passe de $\mathcal{M}_2$ (resp $\mathcal{M}_3$) il faut choisir les 2 (resp 3) positions des chiffres, puis les chiffres et les lettres dans leurs emplacements respectifs. Ce qui nous donne

$$|\mathcal{M}_2| = \binom{8}{2} N^6 K^2, \quad |\mathcal{M}_3| = \binom{8}{3} N^5 K^3 \text{ et}$$

$$|\mathcal{M}_2 + \mathcal{M}_3| = \binom{8}{2} N^6 K^2 + \binom{8}{3} N^5 K^3 = \binom{8}{2} N^5 K^2 (N + \frac{6}{3} K) = 28.26^5.10^2.(26+20)$$

$$= 1530321228800, \text{ ce n'était pas nécessaire de le calculer explicitement.}$$

On calcule ensuite

$$p_2 = \frac{\mathcal{M}_2}{\mathcal{M}_2 + \mathcal{M}_3} = \frac{26}{46} \text{ et } p_3 = 1 - p_2 = \frac{20}{46}.$$

Enfin on combine avec l'algorithme de génération d'un vecteur de $n = 8$ bits avec exactement k bits à 1 pour générer un mot de passe avec exactement k chiffres ($k = 2$ ou 3).

Générateur-Passwd-2-3(n)

```
// renvoie un mot de passe de n caractères avec exactement
   2 ou 3 chiffres
if (Random ()⩽ p_2) // il y a 2 chiffres
  T=génère_vecteur (n, 2)
else // il y a 3 chiffres
  T=génère_vecteur (n, 3)
for i = 1 to n
  if T[i]
    M[i] = génère_chiffre () ;
  else
    M[i] = génère_lettre () ;
return M
```

Les appels à **génère_chiffre** () (resp **génère_lettre** ()) génèrent uniformément un chiffre (resp une lettre), ces fonctions sont construites à partir de **Alea** () et du tableaux de caractères C.

génère_chiffre ()

```
// renvoie un chiffre choisi uniformément dans {0,··· ,9}
return C[Alea (10)+26]
```

génère_lettre ()

```
// renvoie une lettre choisie uniformément dans {a,··· ,z}
return C[Alea (26)]
```

génère_vecteur (n, k)

```
// renvoie un vecteur de n bits ayant exactement, k bits à
   1, (voir le cours)
for i = 1 to n
  if (Random ()⩽ k/(n−i+1)) // on choisit la i-ème place
    T[i]=1
    k=k-1
  else
    T[i]=0
return T
```

Un deuxième algorithme consiste à générer uniformément un vecteur de 8 caractères (incluant lettres et chiffres) et de rejeter le résultat s'il ne contient pas 2 ou 3 chiffres.

```
mot_de_passe()
  // renvoie un vecteur de n caractères ayant exactement, 2
     ou 3 chiffres
  repeat// le mot de passe est correct
     chiffre=0
     for i = 1 to 8
        j=Alea (36)
        if j > 25 // c'est un chiffre
           chiffre ++
        M[i]=C[j]
  until (chiffre = 2) ou (chiffre = 3)
  return M
```

2. Démontrer que l'algorithme génère un mot de passe de loi uniforme parmi tous les mots de passe possibles.

Solution: 2 points sont à démontrer : l'uniformité de la valeur produite et l'indépendance des appels successifs. Le deuxième point est facile, il repose sur le fait que les appels à **Alea** sont modélisés par une séquence d'appels indépendants.

Pour démontrer le premier point, on considère un mot de passe arbitraire m et on calcule la probabilité de générer le mot m. Pour que l'algorithme soit correct il faut démontrer que cette probabilité est uniforme, c'est à dire vaut $\frac{1}{|\mathcal{M}_2|+|\mathcal{M}_3|}$.

Considérons le premier algorithme et soit m un mot de passe, calculons $\mathbb{P}(M = m)$. Deux cas se présentent, m contient 2 ou 3 chiffres. Supposons que m contienne 2 chiffres. La probabilité que $(M = m)$ sachant qu'il a 2 chiffres est donc

$$\underbrace{\frac{1}{\binom{n}{2}}}_{(a)} \cdot \underbrace{\frac{1}{N^6K^2}}_{(b)} = \frac{1}{|\mathcal{M}_2|}$$

(a) correspond au choix uniforme dans la position des chiffres dans le mot (cf cours pour la génération uniforme d'une partie de cardinal 2 parmi n éléments.
(b) correspond au choix uniforme et indépendant des 6 lettres et 2 chiffres.

La probabilité de choisir 2 chiffres est $\frac{|\mathcal{M}_2|}{|\mathcal{M}_2|+|\mathcal{M}_3|}$ On en déduit que (cas où m a 2 chiffres)

$$\mathbb{P}(M = m) = \frac{|\mathcal{M}_2|}{|\mathcal{M}_2| + |\mathcal{M}_3|} \frac{1}{|\mathcal{M}_2|} = \frac{1}{|\mathcal{M}_2| + |\mathcal{M}_3|}.$$

La preuve lorsque le nombre de chiffres est 3 est identique (on remplace les 2 par 3 et $\mathcal{M}_2$ par $\mathcal{M}_3$).

Donc le premier algorithme génère uniformément un mot de passe ayant 2 ou 3 chiffres.

L'algorithme 2 est un algorithme a base de rejet, il génère uniformément un mot composé de lettres ou chiffres (ensemble qui contient tous les mots de passes ayant exactement 2 ou 3 chiffres), l'accepte s'il respecte la contrainte, donc il génère selon une loi uniforme un mot de passe ayant exactement 2 ou 3 chiffres (cf cours).

3. Évaluer le coût de votre algorithme en nombre d'appels à **Alea** ou **Random**

Solution: Le coût du premier algorithme est constant, on appelle **Alea** ou **Random** 9 fois.

Pour le deuxième algorithme, la probabilité d'acceptation est

$$p_a = \frac{|\mathcal{M}_2| + |\mathcal{M}_3|}{|\mathcal{M}'|}$$

où M' est l'ensemble tous les mots possibles $|\mathcal{M}'| = (N+K)^8 = 36^8$.

En fait, si on note p la probabilité d'avoir un chiffre dans un tel mot, le nombre de chiffres suit une loi binomiale $\mathcal{B}in(8, \frac{10}{36})$ et

$$p_a = \binom{8}{2}\left(\frac{10}{36}\right)^2\left(\frac{26}{36}\right)^6 + \binom{8}{3}\left(\frac{10}{36}\right)^3\left(\frac{26}{36}\right)^5 \simeq 0.54$$

Donc en moyenne on obtient de l'ordre de 2 tirages.