Examen du 12 janvier 2007 (Durée 2 heure)

Les documents ne sont pas autorisés mais les calculettes et les tables statistiques sont autorisées.

Dans tout le sujet, la fonction *random()* renvoie une séquence de nombres pseudo-aléatoires indépendants de loi uniforme sur l'intervalle $[0, 1[$.

Dés et pièces

On jette un dé, puis autant de pièces de monnaie que le nombre obtenu sur le dé et on compte le nombre de piles obtenus.

Question 1.1 : Modélisation

Modéliser ce problème à l'aide de variables aléatoires, on justifiera soigneusement les hypothèses.

Question 1.2 : Loi de probabilité

Calculer la loi du nombre de piles obtenus, sa moyenne et sa variance.

Lois continues

On considère l'algorithme de génération suivant :

répéter
  $u = random()$;
  $v = random()$;
jusqu'à $u^2 + v^2 \leqslant 1$
Return u ;

Question 2.1 : Génération

A l'aide de la table de nombres aléatoires donnée en annexe donner les résultats de 5 exécutions consécutives de l'algorithme. On précisera le choix de parcours de la table.

Question 2.2 : Loi de probabilité

Calculer la loi de X, valeur retournée par l'algorithme. Calculer sa moyenne.

Question 2.3 : Validation

On observe $n = 1000$ tirages selon cet algorithme. On observe $\sum_{i=1}^{n} x_i = 417$ et $\sum_{i=1}^{n} x_i^2 = 224$. Donner un estimateur de $\mathbb{E}X$ avec un intervalle de confiance à 90%. Commenter votre résultat.

Annexe :Réels (float) pseudo-aléatoires

0.327010	0.057128	0.994553	0.214157	0.825574	0.795653	0.068671	0.667426	0.755272	0.461837
0.788446	0.411315	0.905150	0.781532	0.794132	0.095405	0.647180	0.548351	0.271737	0.638842
0.723094	0.464648	0.332958	0.886690	0.764691	0.604677	0.390348	0.213932	0.135788	0.528952
0.155550	0.462798	0.586080	0.150103	0.676956	0.411654	0.945757	0.745627	0.079080	0.701028
0.207464	0.867526	0.112343	0.112614	0.649058	0.906475	0.208019	0.296238	0.454826	0.479756
0.935080	0.177919	0.944403	0.268038	0.064609	0.709094	0.872715	0.454958	0.923026	0.008503
0.983909	0.078576	0.471301	0.569990	0.228680	0.148257	0.981644	0.174436	0.893884	0.060724
0.875465	0.101348	0.928250	0.987808	0.213961	0.577309	0.894283	0.421980	0.873546	0.349109
0.901736	0.808627	0.527028	0.846139	0.076665	0.591637	0.555233	0.949380	0.046595	0.478259
0.957883	0.030504	0.556835	0.429184	0.600494	0.785515	0.577441	0.582138	0.959951	0.471325

Pas touche

On considère un espace mémoire constitué de N blocs contigus. Un algorithme de gestion de table place des données dans ces blocs. Pour un bon fonctionnement de l'algorithme deux blocks contigus ne doivent pas être occupés par des données. On modélise la configuration de l'espace mémoire par un vecteur de N bits, le i-ème bit est à 1 si le bloc i est occupé par une donnée. Un vecteur de configuration de la mémoire sera dit admissible si et seulement si 2 bits à 1 du vecteur ne sont pas voisins.

Question 3.1 : Exemples pour de petites tailles

Donner les vecteurs de configuration admissibles pour $N = 1$, $N = 2$, $N = 3$.

Question 3.2 : Génération

Ecrire un algorithme de génération uniforme de configuration admissible en vous basant sur la méthode du rejet.

On note F_N le nombre de configurations admissibles pour un vecteur de taille N.

Question 3.3 : Combinatoire

Démontrer que les F_N vérifient l'équation de récurrence

$$F_{N+2} = F_{N+1} + F_N \text{ pour } N \geqslant 1.$$

Question 3.4 : Taux de rejet

Calculer la probabilité de rejet de l'algorithme précédent en fonction de F_N. En déduire le coût de l'algorithme pour $N = 1$ à 10. Commenter votre résultat.

Question 3.5 : Loi du dernier bit

En supposant que $X = [X_1, \cdots, X_N]$ soit un vecteur admissible tiré selon la loi uniforme, calculer la probabilité que le dernier bit X_N soit à 1.

Question 3.6 : Algorithme glouton

En déduire un algorithme récursif de génération et donner sa preuve. Commenter votre résultat.