4. Машина Тюрінга задана наступною таблицею відповідності:

Q \ A	S_0	a	b	c	d
q_0	–	$q_0 a Л$	$q_1 d Л$	$q_0 c Л$	$q_2 d П$
q_1	–	–	–	$q_2 c Л$	$q_2 c П$
q_2	–	$q_4 d П$	–	$q_4 c П$	–
q_3	–	–	$q_2 c П$	–	–
q_4	Зупинка				

Початковий стан машини q_0. Необхідно застосувати дану машину для переробки вихідного слова $acabcd$.

1.6. Генератори псевдовипадкових чисел

1.6.1. Короткі теоретичні відомості

Криптографічні додатки використовують для генерації випадкових чисел спеціальні алгоритми. Якщо вибрати гарний алгоритм, то отримана числова послідовність пройде більшість тестів на випадковість. Такі числа називають псевдовипадковими числами.

Генератор псевдовипадкових чисел (ГПВЧ) – алгоритм, що породжує послідовність чисел, елементи якої майже незалежні один від одного та підпорядковуються заданому розподілу (зазвичай рівномірному).

Лінійний конгруентний метод є одним з найпростіших ГПВЧ, суть його полягає в обчисленні членів лінійної рекурентної послідовності за формулою

$$X_{k+1} = (aX_k + c)\ mod\ m,$$

де a і c – деякі цілочислові коефіцієнти,

m – деяке натуральне число,

X_k – попереднє псевдовипадкове число.

Отримана послідовність залежить від вибору стартового числа, і при різних його значеннях виходять різні послідовності випадкових чисел. Разом з тим багато властивостей цієї послідовності визначаються вибором коефіцієнтів у формулі і не залежать від вибору стартового числа. Для даних констант виписані умови, що гарантують задовільну якість випадкових чисел, що отримуються.

При реалізації алгоритму рекомендується вибирати m^e, де e – число бі-

тів в машинному слові, тому що це дозволяє позбутися відносно повільної операції приведення за модулем.

Один з прикладів вдалих параметрів методу:

$$m = 2^{32},\ a = 69069, c = 5.$$

Особливості розподілу випадкових чисел, що генеруються лінійним конгруентним алгоритмом, роблять неможливим їх використання в статистичних алгоритмах, які потребують високого дозволу.

Один з найбільш поширених датчиків Фібоначчі оснований на такій формулі:

$$X_k = \begin{cases} X_{k-a} - X_{k-b}, \ \textit{якщо}\ X_{k-a} \geq X_{k-b}, \\ X_{k-a} - X_{k-b} + 1, \ \textit{якщо}\ X_{k-a} < X_{k-b}, \end{cases}$$

де X_k – дійсні числа з діапазону $[0, 1)$;

a, b – цілі позитивні числа, які називаються лагами.

При реалізації через цілі числа досить формули $X_k = X_{k-a} - X_{k-b}$.

Для роботи датчика Фібоначчі потрібно знати попередні згенеровані випадкові числа. Для старту датчика Фібоначчі необхідно $\max(a,b)$ випадкових чисел, які можуть бути згенеровані простим конгруентним датчиком.

Найбільшу популярність датчики Фібоначчі отримали у зв'язку з тим, що швидкість виконання арифметичних операцій з дійсними числами зрівнялась зі швидкістю цілочислової арифметики, а фібоначчієві датчики природно реалізуються в арифметиці дійсних чисел.

Отримані випадкові числа мають гарні статистичні властивості, причому всі біти випадкового числа рівнозначні за статистичними властивостями.

Для вибору лагів a і b рекомендуються такі значення:

$$(a, b) = (55{,}24),\ (17{,}5) \text{ чи } (97{,}33).$$

Якість отримуваних випадкових чисел залежить також від значення констант.

Статистичні тести на випадковість дозволяють отримати числову характеристику послідовності і дати відповідь на питання, чи є випадковою послідовність.

Розглянемо деякі з тестів пакета NIST.

Тест 1. Частотний побітовий тест.

Суть даного тесту полягає у визначенні співвідношення між нулями і одиницями у всій двійковій послідовності. Мета – з'ясувати, чи дійсно число нулів і одиниць у послідовності приблизно однакове, як це можна було б припустити у випадку істинно випадкової бінарної послідовності. Всі наступні тести проводяться за умови, що даний тест пройдено.

Вхідні дані для тесту – послідовність бітів $\epsilon = \epsilon_1, \epsilon_2, \ldots, \epsilon_n$.

Тест обчислює статистику $S_{obs} = \frac{\sum_{i=1}^{n} X_i}{\sqrt{n}}$, $X_i = 2\epsilon_1 - 1$, тобто X_i може набувати значень $+1$ та -1. Розподіл даної статистики є напівнормальним для великих n (якщо величина $|z|$ розподілена як нормальна, то розподіл називається напівнормальним). Якщо послідовність випадкова, то плюс і мінус одиниці будуть прямувати до взаємного винищення, а підсумкова статистика прагне до нуля. Якщо ж все-таки є занадто багато одиниць чи занадто багато нулів у початковій послідовності, то статистика буде більшою від нуля.

Покрокове описання тесту.

Крок 1. Перетворити нулі та одиниці вихідної послідовності в -1 і $+1$ та скласти: $S_n = X_1 + X_2 + \cdots + X_n, X_i = 2\,\epsilon_i - 1$.

Крок 2. Обчислити статистику $S_{obs} = \frac{|S_n|}{\sqrt{n}}$.

Крок 3. Обчислити значення $p = erfc\left(\frac{S_{obs}}{\sqrt{2}}\right)$, де $erfc$ – це додаткова функція помилок. Вона визначається як $erfcx = 1 - erfx = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$, де $erfx = \frac{2}{\sqrt{\pi}} \int_0^{x} e^{-t^2} dt$ – функція помилок. Наведені інтеграли неможливо взяти, тому для обчислення даних функцій використовуються розкладення до ряду. Значення додаткової функції помилок доступні у вигляді довідкових таблиць. Є готові реалізації даної функції в бібліотеках C++ (файл *cmath*, функції *erfcf( )* та *erfcl ( )*), Java (метод *Erf.erfc( )* в пакеті org.apache.commons.math3.special), Python (*math.erfc()*).

Крок 4. Інтерпретація результатів. Якщо $p < 0{,}01$, то послідовність вважається невипадковою.

Тест 2. Частотний блочний тест.

Суть тесту – визначення частки одиниць всередині блока довжиною m біт. Мета – з'ясувати, чи дійсно частота повторів одиниць у блоці довжиною M біт приблизно дорівнює $M/2$, як можна було б припустити у випадку абсолютно випадкової послідовності. Якщо взяти $M=1$, даний тест переходить у тест №1 (частотний побітовий тест).

Вхідні дані для тесту – послідовність бітів $\epsilon = \epsilon_1, \epsilon_2, \ldots, \epsilon_n$ і число M – довжина блока.

Тест обчислює статистику $\chi^2(obs)$, яка відображає, наскільки добре пропорція одиниць у даному блоці довжини M відповідає очікуваній пропорції $1/2$. Розподіл даної статистики – χ^2.

Покрокове описання тесту.

Крок 1. Розділити послідовність на $N = \left\lfloor \frac{n}{M} \right\rfloor$ блоків, які не перекриваються. Біти, які не використовуються, відкидаються.

Крок 2. Визначити пропорцію π_i одиниць у кожному блоці довжиною M біт, використовуючи формулу $\pi_i = \frac{\sum_{j=1}^{M} \in_{(i-1)M+j}}{M}$, $1 \le i \le N$.

Крок 3. Обчислити статистику $\chi^2(obs) = 4M \sum_{i=1}^{N} (\pi_i - 0{,}5)^2$.

Крок 4. Обчислити значення $p = Q\left(\frac{N}{2}, \frac{\chi^2(obs)}{2}\right)$, де Q (також позначається як $igamc$) – неповна верхня гамма-функція, яка визначається як $Q(a,x) = \frac{1}{\Gamma(a)} \int_x^\infty e^{-t} t^{a-1} dt$, де $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ – стандартна гамма-функція. Значення даної функції доступні в таблицях, а також у бібліотеках C++ (файл boost/math/special_functions/gamma.hpp, функція *gamma_q*), Java (метод Gamma.regularizedGammaQ з пакета org.apache.commons.math3.special), Python (scipy.special.gammainc і mpmath.gammainc).

Крок 5. Інтерпретація результатів. Якщо $p < 0{,}01$, то послідовність вважається невипадковою.

3. Тест на послідовність однакових бітів.

Суть полягає в підрахунку повного числа рядів у вихідній послідовності, де під словом «ряд» розуміють безперервну підпослідовність однакових бітів. Ряд довжиною k біт складається з k абсолютно ідентичних бітів, починається і закінчується з біта, що містить протилежне значення. Мета даного тесту – зробити висновок про те, чи дійсно кількість рядів, що складаються з одиниць і нулів різної довжини, відповідає їх кількості у випадковій послідовності. Зокрема, визначається, швидко чи повільно чергуються одиниці і нулі у вихідній послідовності. Якщо обчислене в ході тесту значення ймовірності $p < 0{,}01$, то дана двійкова послідовність не є істинно випадковою. В протилежному випадку, вона має випадковий характер.

Вхідні дані для тесту – послідовність бітів $\epsilon = \epsilon_1, \epsilon_2, \ldots, \epsilon_n$.

Тест обчислює статистику $V_n(obs)$ – загальну кількість безперервних відрізків, що складаються лише з нулів чи лише одиниць. Розподіл даної статистики – χ^2.

Покрокове описання тесту.

Крок 1. Обчислити пропорцію одиниць у вихідній послідовності:

$$\pi = \frac{\sum_j \epsilon_j}{n}.$$

Крок 2. Визначити, чи виконана передумова $\left|\pi - \frac{1}{2}\right| < \tau;\ \tau = \frac{2}{\sqrt{n}}$. Якщо вона не виконана, то тест вже не пройдено, а підсумкове значення p кладеться рівним нулю.

Крок 3. Обчислити статистику $V_n(obs) = \sum_{k=1}^{n-1} r(k) + 1$, де $r(k) = 0$, якщо $\epsilon_k = \epsilon_{k+1}$, інакше $r(k) = 1$.

Крок 4. Обчислити $p = erfc\left(\frac{\left|V_n(obs) - 2n\pi(1-\pi)\right|}{2\sqrt{2n}\pi(1-\pi)}\right)$.

Крок 5. Інтерпретація результатів. Якщо $p < 0{,}01$, то послідовність вважається невипадковою.

4. Тест на найдовшу послідовність одиниць у блоці.

У даному тесті визначається найдовший ряд одиниць в середині блока довжиною M біт. Мета – з'ясувати, чи дійсно довжина такого ряду відповідає очікуванням довжини найдовшого ряду одиниць у випадку абсолютно випад-

кової послідовності. Якщо обчислене в ході тесту значення ймовірності $p < 0{,}01$, вважається, що вихідна послідовність не є випадковою. В протилежному випадку роблять висновок про її випадковість. Необхідно зазначити, що з припущення про приблизно однакову частоту появи одиниці і нулів (тест №1) випливає, що такі самі результати даного тесту будуть отримані при розгляді найдовшого ряду нулів. Тому вимірювання можна проводити лише з одиницями.

Вхідні дані для тесту:

– послідовність бітів $\epsilon = \epsilon_1, \epsilon_2, \dots, \epsilon_n$;

– довжина блока M;

– кількість блоків N.

Рекомендовані значення M залежно від довжини послідовності n наведені в табл. 1.5.

Таблиця 1.5 – Рекомендовані значення довжини блока M залежно від довжини послідовності n

Мінімальна довжина послідовності n	Довжина блока M
128	8
6272	128
750 000	10 000

Тест обчислює статистику $\chi^2(obs)$, яка відображає, наскільки спостережувана довжина найдовшої послідовності з одиниць у блоці довжиною M відповідає очікуваній. Розподіл даної статистики – χ^2.

Покрокове описання тесту.

Крок 1. Розділити послідовність на блоки довжиною M.

Крок 2. Розподілити за таблицею частоти v_i найдовших послідовностей з одиниць в кожному блоці, де кожна клітинка містить кількість послідовностей одиниць даної довжини (табл. 1.6).

Крок 3. Обчислити $\chi^2(obs) = \dfrac{\sum_{i=0}^{K}(v_i - N\pi_i)^2}{N\pi_i}$, де K та N знаходять з табл. 1.7.

Теоретичні ймовірності π_i задаються константами. Для $K = 3$, $M = 8$ необхідно взяти $\pi_0 = 0{,}2148$, $\pi_1 = 0{,}3672$, $\pi_2 = 0{,}2305$, $\pi_3 = 0{,}1875$.

Таблиця 1.6 – Визначення частоти v_i

v_i	$M = 8$	$M = 128$	$M = 10\,000$
v_0	≤ 1	≤ 4	≤ 10
v_1	2	5	11
v_2	3	6	12
v_3	≥ 4	7	13
v_4		8	14
v_5		≥ 9	15
v_6			≥ 16

Крок 4. Обчислити $p = Q\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right)$.

Таблиця 1.7 – Визначення величин K та N залежно від M

M	K	N
8	3	16
128	5	49
10 000	6	75

Крок 5. Інтерпретація результатів. Якщо $p < 0{,}01$, то послідовність вважається невипадковою.

1.6.2. Приклади розв'язання задач

Задача 1

Використовуючи лінійний конгруентний метод, згенерувати послідовність з 10 чисел.

Розв'язання

Задамо коефіцієнти для обчислення елементів послідовності $a = 69069$, $c = 5$. Покладемо $m = 2^{32}$. Виберемо довільно стартове число послідовності $x_1 = 56473829$.

Використовуємо формулу лінійного конгруентного методу $X_{k+1} = (aX_k + c) \bmod m$:

$x_2 = (ax_1 + c)\ mod\ m = 760590438;$

$x_3 = (ax_2 + c)\ mod\ m = 1475964851;$

$x_4 = (ax_3 + c)\ mod\ m = 2367523164;$

$x_5 = (ax_4 + c)\ mod\ m = 167553713;$
$x_6 = (ax_5 + c)\ mod\ m = 2125507778;$
$x_7 = (ax_6 + c)\ mod\ m = 419574111;$
$x_8 = (ax_7 + c)\ mod\ m = 1419926552;$
$x_9 = (ax_8 + c)\ mod\ m = 1623783229;$
$x_{10} = (ax_9 + c)\ mod\ m = 2897810654.$

Задача 2

Використовуючи метод Фібоначчі із запізненнями, згенерувати послідовність з 10 чисел.

Розв'язання

Задамо лаги $(a, b) = (5,2)$і початкові елементи $(x_0, x_1, x_2, x_3, x_4) = (0{,}2;\ 0{,}5;\ 0{,}1;\ 0{,}8;\ 0{,}7)$. Використовуємо формулу методу Фібоначчі із запізненнями $X_k = \begin{cases} X_{k-a} - X_{k-b},\ якщо\ X_{k-a} \geq X_{k-b}, \\ X_{k-a} - X_{k-b} + 1,\ якщо\ X_{k-a} < X_{k-b}, \end{cases}$:

$x_5 = x_0 - x_3 + 1 = 0{,}4;$
$x_6 = x_1 - x_4 + 1 = 0{,}8;$
$x_7 = x_2 - x_5 + 1 = 0{,}7;$
$x_8 = x_3 - x_6 = 0;$
$x_9 = x_4 - x_7 + 1 = 1{,}0;$
$x_{10} = x_5 - x_8 = 0{,}4;$
$x_{11} = x_6 - x_9 + 1 = 0{,}8;$
$x_{12} = x_7 - x_{10} = 0{,}3;$
$x_{13} = x_8 - x_{11} + 1 = 0{,}2;$
$x_{14} = x_9 - x_{12} = 0{,}7;$
$x_{15} = x_{10} - x_{13} = 0{,}2.$

Задача 3

Провести перевірку випадковості послідовності $\epsilon = 0100110010$, використовуючи тест на однакові біти, що йдуть підряд.

Розв'язання

Крок 1. Перетворимо нулі та одиниці вихідної послідовності в -1 і $+1$ та складемо:

$S_n = X_1 + X_2 + \cdots + X_n,\ X_i = 2\epsilon_i - 1,$

$S_n = -1 + 1 - 1 - 1 + 1 + 1 - 1 - 1 + 1 - 1 = -2.$

Крок 2. Обчислимо статистику $S_{obs} = \frac{|S_n|}{\sqrt{n}}$:

$$S_{obs} = \frac{|-2|}{\sqrt{10}} = 0{,}632455.$$

Крок 3. Обчислимо значення $p = erfc\left(\frac{S_{obs}}{\sqrt{2}}\right)$:

$$p = erfc\left(\frac{0{,}632455}{\sqrt{2}}\right) = erfc\ 0{,}447213 = 0{,}5270896.$$

Крок 4. Оскільки $p \geq 0{,}01$, то послідовність визнається випадковою.

Задача 4

Провести перевірку випадковості послідовності $\epsilon = 0100110010$, використовуючи частотний блочний тест. Довжина блока $M = 3$.

Розв'язання

Крок 1. Розділимо послідовність на $N = \left\lfloor \frac{n}{M} \right\rfloor = \left\lfloor \frac{10}{3} \right\rfloor = 3$ блоків, що не перекриваються: 010, 011, 001. Останній біт 0, що не використовується, відкидається.

Крок 2. Визначимо пропорцію π_i одиниць у кожному блоці довжиною M біт, використовуючи формулу $\pi_i = \frac{\sum_{j=1}^{M} \in_{(i-1)M+j}}{M}$, $1 \leq i \leq N$:

$\pi_1 = \frac{1}{3}, \pi_2 = \frac{2}{3}, \pi_3 = \frac{1}{3}.$

Крок 3. Обчислимо статистику $\chi^2(obs) = 4M\sum_{i=1}^{N}(\pi_i - 0{,}5)^2$:

$\chi^2(obs) = 4 \cdot 3\left(\left(\frac{1}{3} - 0{,}5\right)^2 + \left(\frac{2}{3} - 0{,}5\right)^2 + \left(\frac{1}{3} - 0{,}5\right)^2\right) = 12(0{,}0278 + +0{,}0278 + 0{,}0278) = 1.$

Крок 4. Обчислимо значення:

$p = Q\left(\frac{N}{2}, \frac{\chi^2(obs)}{2}\right) = Q\left(\frac{3}{2}, \frac{1}{2}\right) = 0{,}801252.$

Крок 5. Оскільки $p \geq 0{,}01$, то послідовність вважається випадковою.

Задача 5

Провести перевірку випадковості послідовності $\epsilon = 0100110010$, використовуючи тест на послідовність однакових бітів.

Розв'язання

Крок 1. Обчислимо пропорцію одиниць у вихідній послідовності:

$$\pi_i = \frac{\sum_j \in_j}{M}, = \frac{4}{10} = 0,4.$$

Крок 2. Перевіримо передумову $\left|\pi - \frac{1}{2}\right| < \tau;\ \tau = \frac{2}{\sqrt{n}}$:

$\left|\pi - \frac{1}{2}\right| = |0,4 - 0,5| = 0,1;$

$\tau = \frac{2}{\sqrt{n}} = \frac{2}{\sqrt{10}} = 0,632455;$

$0,1 < 0,632455.$

Умову виконано, тому тест продовжується.

Крок 3. Обчислимо статистику $V_n(obs) = \sum_{k=1}^{n-1} r(k) + 1,\ \ r(k) = 0,$ якщо $\epsilon_k = \epsilon_{k+1}$, інакше $r(k) = 1.$

$\epsilon = 0100110010$

$V_n(obs) = (1 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 1) + 1 = 6 + 1 = 7.$

Крок 4. Обчислимо:

$p = erfc\left(\frac{|V_n(obs) - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}\right) = erfc\left(\frac{|7 - 2\cdot 10\cdot 0,4(1-0,4)|}{2\sqrt{2\cdot 10}\cdot 0,4(1-0,4)}\right) = erfc\left(\frac{|7-4,8|}{2,1466}\right) =$

$= erfc(1,0249) = 0,1472.$

Крок 5. Оскільки $p \geq 0,01$, то послідовність вважається випадковою.

Задача 6

Провести перевірку випадковості послідовності, використовуючи тест на найдовшу послідовність одиниць у блоці:

11001100 00010101 01101100 01001100 11100000 00000010
01001101 01010001 00010011 11010110 10000000 11010111
11001100 11100110 11011000 10110010.

Розв'язання

Крок 1. Розділяємо послідовність на блоки довжиною $M = 8$, тому що довжина послідовності дорівнює 128.

Крок 2. Розподіляємо за таблицею частоти v_i найдовших послідовностей з одиниць у кожному блоці, де кожна клітинка містить кількість послідовнос-

тей одиниць даної довжини.

Блок	Довжина одиниць	Блок	Довжина одиниць
11001100	2	00010011	2
00010101	1	11010110	2
01101100	2	10000000	1
01001100	2	11010111	3
11100000	3	11001100	2
00000010	1	11100110	3
01001101	2	11011000	2
01010001	1	10110010	2

$v_0 = \{$ кількість блоків з максимальною довжиною $\leq 1\} = 4;$
$v_1 = \{$ кількість блоків з максимальною довжиною $= 2\} = 9;$
$v_2 = \{$ кількість блоків з максимальною довжиною $= 3\} = 3;$
$v_3 = \{$ кількість блоків з максимальною довжиною $\geq 4\} = 0.$

Крок 3. Обчислимо $\chi^2(obs) = \dfrac{\sum_{i=0}^{K}(v_i - N\pi_i)^2}{N\pi_i}$:

$$\chi^2(obs) = \frac{(4-16\cdot 0{,}2148)^2}{16\cdot 0{,}2148} + \frac{(9-16\cdot 0{,}3672)^2}{16\cdot 0{,}3672} + \frac{(3-16\cdot 0{,}2305)^2}{16\cdot 0{,}2305} + \frac{(0-16\cdot 0{,}1875)^2}{16\cdot 0{,}1875} = 4{,}8826.$$

Крок 4. Обчислимо $p = Q\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right) = Q\left(\frac{3}{2}, \frac{4{,}8826}{2}\right) = 0{,}1860.$

Крок 5. Оскільки $p \geq 0{,}01$, то послідовність вважається випадковою.

Задачі для аудиторних занять

Дані завдання передбачається виконувати, не використовуючи комп'ютер, тому їх розмірність є невеликою і не потрібно обчислювати значення функцій $erfc$ і Q.

1. Визначте послідовність з перших десяти чисел, використовуючи лінійний конгруентний метод для різних параметрів a, c та x_0 (при $m = 32$):

1) $a = 2, c = 6, x_0 = 16;$
2) $a = 3, c = 1, x_0 = 25;$
3) $a = 4, c = 3, x_0 = 31;$
4) $a = 5, c = 6, x_0 = 13;$
5) $a = 7, c = 9, x_0 = 27;$

6) $a = 9, c = 11, x_0 = 19$;

7) $a = 10, c = 3, x_0 = 23$;

8) $a = 11, c = 8, x_0 = 7$.

2. Обчисліть послідовність з десяти чисел, що генерується методом Фібоначчі із запізненням, використовуючи такі вихідні дані: пару (a, b) і початкові значення $x_0, \dots, x_k$:

1) (5; 2); (0,29; 0,57; 0,96; 0,90; 0,91);

2) (5;2), (0,07; 0,42; 0,04; 0,88; 0,66);

3) (5;3); (0,13; 0,30; 0,86; 0,63; 0,84);

4) (5;3); (0,13; 0,46; 0,17; 0,62; 0,07);

5) (7;2); (0,21; 0,68; 0,43; 0,75; 0,38; 0,78; 0,61);

6) (7;3); (0,84; 0,41; 0,74; 0,22; 0,05; 0,42; 0,61);

7) (7;4); (0,22; 0,33; 0,88; 0,29; 0,93; 0,50; 0,58);

8) (7;5); (0,46; 0,69; 0,75; 0,59; 0,04; 0,11; 0,36).

3. Обчисліть статистику для перевірки на випадковість заданих послідовностей чисел:

а) частотний побітовий тест – статистика S_{obs};

б) частотний блочний тест (довжина блока $M = 3$) – статистика $\chi^2(obs)$;

в) тест на послідовність однакових бітів – статистика $V_n(obs)$.

Послідовності:

1) 0 0 0 1 1 1 0 0 0 1;

2) 0 1 0 1 0 1 1 0 0 0;

3) 0 1 1 1 1 1 0 0 0 1;

4) 1 0 0 0 1 1 0 0 0 0;

5) 0 0 1 0 1 0 0 1 0 0;

6) 1 1 1 0 1 1 1 0 0 1;

7) 1 0 1 1 0 1 1 1 1 0;

8) 0 1 1 0 1 0 0 1 0 0.

4. Обчисліть статистику для тесту на найдовшу послідовність одиниць в блоці. Довжина кожної вихідної послідовності 128 біт.

1) 00110011 00100000 11001010 01000000 11100101 10000101 10110110 00100110 10010101 10011000 11010010 00101000 10011000 00011101 01001010

00010011;

2) 10100011 11011000 10010110 10000000 10001001 10110010 00010111 10000101 10000111 11000100 11111011 10110100 00101100 10110001 00001000 10000111;

3) 00111111 01001111 10110000 10101001 10111110 11101000 10110001 00011011 00100101 01101001 01101001 00001100 11111100 00100111 00101111 11111000;

4) 10100001 00100000 01000011 01010010 01000101 00000101 11100111 11100011 00010111 10011100 10100110 00010111 01000011 00100100 10100101 10100000;

5) 11110001 11010101 01011101 11111110 01010111 00111001 00000010 10100101 10010111 00001101 11100011 11100000 00000000 01010100 10011100 11111000;

6) 01011110 01110001 11111011 10000110 11110110 00000010 01011110 00010110 00000011 01101101 11000000 00001110 01001000 00110000 01001111 10010000;

7) 10101111 10001111 00001011 11101111 10011100 10011101 01101100 01110010 11100001 10110100 10111110 00010001 11100111 00111111 01100010 01011100;

8) 11000000 10001100 11100001 11001010 10001000 00100100 10111010 11111010 01111101 10101000 01000111 01011011 00010010 10111011 00111111 10111110.

Запитання для самоконтролю

1. Що таке генератор псевдовипадкових чисел?
2. Перелічіть недоліки простих арифметичних генераторів.
3. Назвіть найпоширеніші генератори псевдовипадкових чисел.
4. У чому полягає суть лінійного конгруентного методу?
5. Наведіть ітеративну формулу, на якій базується один із поширених датчиків Фібоначчі.
6. Назвіть дві основні групи, на які поділяються тести на випадковість.
7. У чому полягає відмінність статичних тестів на випадковість від графічних?