**ПЗ №1 з теми:**

**­­**

**Генерація послідовності випадкових чисел**

**Голомозик Кирило**

**МІТ-21**

**Хід роботи**

1. Опрацював розділ 3 з джерела Haunts Stephen. Applied Cryptography in .NET and Azure Key Vault (pp. 24-30).

2. Створив рішення (Visual Studio solution), у якому містяться проекти, що реалізують наступні задачі.

3. Написав програму, яка забезпечує генерацію та виведення на екран послідовності псевдовипадкових чисел. Порівняти елементи послідовності для однакових та різних початкових значень. Зробити висновки.

№ 1

Маємо два випадки, коли послідовність псевдовипадкових чисел чисел генерується через константне значення, у данній реалізації таке число рівне 1234.

class PsevdoNumbers

{

private Random rnd = new Random();

private int[] array = new int[10];

public void genPsevdo()

{

for (int i = 0; i < 10; i++)

{

array[i] = rnd.Next(0, 1001);

}

}

public void printAll()

{

foreach (int el in array)

{

Console.Write(el + " ");

}

}

}

Програма виводить 10 рандомних чисел від 1 до 1000, при кожному запуску програми ці числа змінюються.  
А якщо ми змінимо цей блок коду:

for (int i = 0; i < 10; i++)

{

array[i] = rnd.Next(0, 1001);

}

Результат: 

Змінивши червоне число, ми змінимо кількість чисел нашої псевдо-випадкової послідовності.  
А якщо змінимо зелене число, то наші числа будуть в іншому діапазоні, наприклад від 0 до 10:

for (int i = 0; i < 10; i++)

{

array[i] = rnd.Next(0, 10);

}

Результат: 

№2

Написати програму, яка забезпечує генерацію та виведення на екран криптографічно стійкої послідовності випадкових чисел. Порівняти елементи послідовності для декількох послідовних спроб. Зробити **висновки.**

**class CryptoNumbers**

**{**

**private RNGCryptoServiceProvider rndNumberGenerator = new RNGCryptoServiceProvider();**

**private byte[] randomNumber = new byte[12];//Ініціалізуємо створення криптографічно стійкої послідовності знаків**

**12-кількість символів нашої послідовності.**

**public void print()//Виводимо нашу послідовність на екран**

**{**

**for (int i = 0; i < 12; i++)(12-кількість послідовностей, які програма генерує.)**

**{**

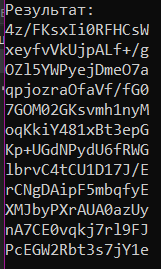
**rndNumberGenerator.GetBytes(randomNumber);**

**Console.WriteLine(Convert.ToBase64String(randomNumber));**

**}**

**}**

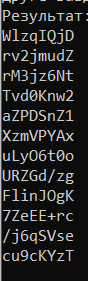
**}**

**Результат:  
**

**Змінимо кількість символів:  
 private byte[] randomNumber = new byte[12]**

**На таке:** **private byte[] randomNumber = new byte[5];**

**Результат:**

****

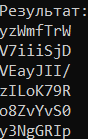
**Змінимо:**

**for (int i = 0; i < 12; i++)(12-кількість послідовностей, які програма генерує.)**

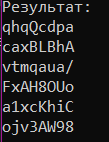
**На:**

**for (int i = 0; i < 6; i++)**

**Результат:**



**При кожному запуску програми послідовність змінюється**

****

**Висновок:  
1. Псевдовипадкова послідовність видає задану кількість чисел, при кожному запуску програми послідовність змінюється, ми можемо задавати параметри цієї послідовності, такі як кількість символів та чисел.**

**2.Криптографічно стійка послідовність- це послідовність символів, які можуть повторюватись, при кожному запуску програми ця послідовність змінюється.**