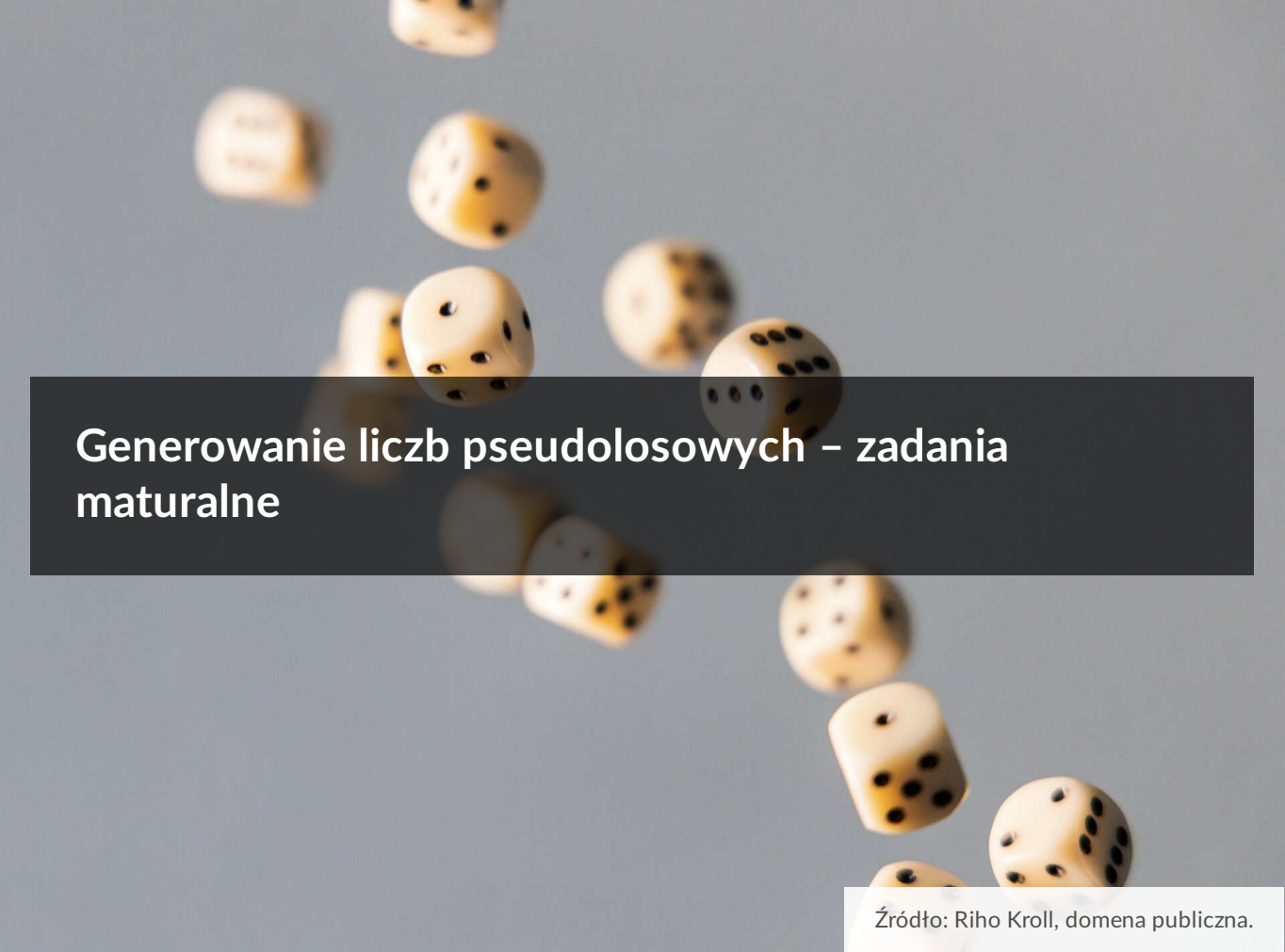




## Generowanie liczb pseudolosowych – zadania maturalne

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Prezentacja multimedialna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



## Generowanie liczb pseudolosowych – zadania maturalne

Źródło: Riho Kroll, domena publiczna.

Liczby losowe oraz pseudolosowe mają wiele zastosowań praktycznych, poznaliśmy je w e-materiale [Liczby pseudolosowe](#). Programy służące do generowania takich liczb mogą być wykorzystywane w firmach zajmujących się grami losowymi.

W tym e-materiale zapoznamy się z przykładowymi zadaniami maturalnymi dotyczącymi tego zagadnienia.

Implementacje algorytmu generowania liczb pseudolosowych w poszczególnych językach programowania przedstawiamy w e-materiałach:

- [Generowanie liczb pseudolosowych w języku C++](#),
- [Generowanie liczb pseudolosowych w języku Java](#),
- [Generowanie liczb pseudolosowych w języku Python](#).

### Twoje cele

- Wymienisz algorytmy wykorzystywane do generowania liczb pseudolosowych.
- Przeanalizujesz rozwiązania zadań typu maturalnego.
- Samodzielnie napiszesz generator liczb pseudolosowych.

# Przeczytaj

---

## Generator liczb pseudolosowych LCG

Generator LCG jest jednym z algorytmów służących do obliczania wartości pseudolosowych. Wykorzystywane są w nim **funkcje modularne** zwane **liniowymi generatorami kongruencyjnymi liczb pseudolosowych** (ang. *pseudorandom number linear congruential generator* – w skrócie LCG).

Można wyróżnić dwa podstawowe wzory do obliczania liczb pseudolosowych z wykorzystaniem generatora LCG.

### Addytywny LCG:

$$X_n = (a \cdot X_{n-1} + c) \bmod m$$

### Multiplikatywny LCG:

$$X_n = a \cdot X_{n-1} \bmod m$$

Gdzie:

- $X_n$  - n-ta liczba pseudolosowa,
- $X_{n-1}$  - poprzednia liczba pseudolosowa,
- $a$  - mnożnik,
- $c$  - parametr, dla generatora multiplikatywnego  $c = 0$ ,
- $m$  - ilość liczb generowanych.

Liczyby wygenerowane przez addytywny LCG mogą przyjmować wartości z przedziału od 0 do  $m-1$ , zaś multiplikatywny z przedziału od 1 do  $m-1$ . Po  $m$  wykonaniach wartości zaczynają się powtarzać, wynika to z własności działania modulo.

Konieczne jest podanie wartości oznaczającej ziarno (ang. *seed*) będące początkową wartością  $X_0$ .

Aby generator działał poprawnie, wartości  $a$ ,  $c$  i  $m$  nie mogą być przypadkowe. Muszą zostać dobrane zgodnie z poniższymi zasadami:

- wartość zmiennej  $m$  jest o jeden większa od największej wartości losowej, jaką algorytm będzie mógł wygenerować;
- parametr  $c$  musi być względnie pierwszy z  $m$ ;

- wartość wyrażenia  $a - 1$  jest wielokrotnością wszystkich dzielników zmiennej  $m$ .

## Zadanie 1.

Wypisz kolejnych 18 wartości, jakie będzie generował addytywny generator LCG dla podanych danych:

- $a = 4$ ,
- $c = 2$ ,
- $m = 9$ ,
- $X_0 = 0$ .

## Przykładowe rozwiązanie

Aby wykonać poprawnie to zadanie, należy podstawić dane wartości  $a$ ,  $c$ ,  $m$  i  $X_0$  do wzoru. Otrzymamy w ten sposób wyrażenie:

$$X_n = (4 \cdot X_{n-1} + 2) \bmod 9$$

Następnie wystarczy tylko obliczać kolejne wartości zgodnie z podanym wzorem. Łatwo można zaobserwować, że gdy zostanie wypisana wartość 0 algorytm „zapętla się” i wraca.

```

1 Xn = (4 * Xn-1 + 2) mod 9
2
3 X0 = (4 * 0 + 2) mod 9 = 2 mod 9 = 2
4 X1 = (4 * 2 + 2) mod 9 = 10 mod 9 = 1
5 X2 = (4 * 1 + 2) mod 9 = 6 mod 9 = 6
6 X3 = (4 * 6 + 2) mod 9 = 26 mod 9 = 8
7 X4 = (4 * 8 + 2) mod 9 = 34 mod 9 = 7
8 X5 = (4 * 7 + 2) mod 9 = 30 mod 9 = 3
9 X6 = (4 * 3 + 2) mod 9 = 14 mod 9 = 5
10 X7 = (4 * 5 + 2) mod 9 = 22 mod 9 = 4
11 X8 = (4 * 4 + 2) mod 9 = 18 mod 9 = 0
12 X9 = (4 * 0 + 2) mod 9 = 2 mod 9 = 2
13 X10 = (4 * 2 + 2) mod 9 = 10 mod 9 = 1
14 X11 = (4 * 1 + 2) mod 9 = 6 mod 9 = 6
15 X12 = (4 * 6 + 2) mod 9 = 26 mod 9 = 8
16 X13 = (4 * 8 + 2) mod 9 = 34 mod 9 = 7
17 X14 = (4 * 7 + 2) mod 9 = 30 mod 9 = 3
18 X15 = (4 * 3 + 2) mod 9 = 14 mod 9 = 5
19 X16 = (4 * 5 + 2) mod 9 = 22 mod 9 = 4

```

$$20 \quad X_{17} = (4 * 4 + 2) \bmod 9 = 18 \bmod 9 = 0$$

## Zadanie 2.

Uzupełnij tabelę tak, aby przy pomocy podanych danych algorytm działał poprawnie. Wypisz współczynniki o następujących właściwościach:

- największa liczba, jaka może zostać wylosowana -  $X_{\max}$ ,
- współczynnik  $m$  o najwyższej możliwej wartości,
- minimalna wartość współczynnika  $a$ ,
- wartości, jakich współczynnik  $c$  nie może przyjąć.

| $X_{\max}$ | maksymalna wartość zmiennej $m$ | minimalna wartość zmiennej $a$ | wartości jakich nie może przyjąć zmienna $c$ |
|------------|---------------------------------|--------------------------------|----------------------------------------------|
| 10         |                                 |                                |                                              |
|            | 30                              |                                |                                              |
|            |                                 | 51                             |                                              |
| 209        |                                 |                                |                                              |
|            |                                 |                                | 5                                            |

## Przykładowe rozwiązanie

a)  $X_{\max} = 10$

$$1 \quad m = X_{\max} + 1 = 10 + 1 = 11$$

$$2 \quad a - 1 = 11$$

3

4 Naturalnie  $a - 1$  może być inną liczbą podzielną przez 11, ale w z

5

$$6 \quad a = 11 + 1 = 12$$

7

8 Wartość  $m = 11$  jest liczbą pierwszą, więc wartość zmiennej  $c$  może

b)  $m = 30$

$$1 \quad X_{\max} = m - 1 = 30 - 1 = 29$$

2

$$3 \quad m = 2 * 3 * 5$$

```

4
5 a - 1 = 30
6 a = 30 + 1 = 31
7
8 c nie może być równa 2, 3 ani 5

```

**c) a = 51**

```

1 a - 1 = 50
2 50 = 2 * 5 * 5
3
4 Liczba 50 musi być wielokrotnością liczby m, więc m może być równ
5
6 m = 50
7
8 Xmax = m - 1 = 50 - 1 = 49
9
10 c może przyjąć każdą wartość poza 2 i 5

```

**d) Xmax = 209**

```

1 m = Xmax + 1 = 210
2 210 = 2 * 3 * 5 * 7
3
4 a = 211
5
6 c nie może być równe 2, 3, 5 ani 7

```

Rozwiązanie zadania:

| <b>Xmax</b> | <b>maksymalna wartość<br/>zmiennej m</b> | <b>minimalna wartość<br/>zmiennej a</b> | <b>wartości jakich nie może<br/>przyjąć zmienna c</b> |
|-------------|------------------------------------------|-----------------------------------------|-------------------------------------------------------|
| 10          | 11                                       | 12                                      | 11                                                    |
| 29          | 30                                       | 31                                      | 2, 3, 5                                               |
| 49          | 50                                       | 51                                      | 2, 5                                                  |
| 209         | 210                                      | 211                                     | 2, 3, 5, 7                                            |

# Słownik

arytmetyka modularna

działanie polegające na obliczeniu reszty z dzielenia danej liczby przez inną wartość  
**liczby względnie pierwsze**

liczby, które w rozkładzie na czynniki pierwsze nie mają żadnego wspólnego dzielnika;  
przykładem liczb względnie pierwszych jest 15 i 8.

**ziarno (ang. *seed*)**

wartość inicjująca generator liczb pseudolosowych

# Prezentacja multimedialna

---

## Generator LFG

Opóźniony generator Fibonacciego, zwany generatorem LFG ma nazwę wynikającą z podobieństwa wzoru, który jest wykorzystywany w algorytmie LFG do definicji ciągu Fibonacciego.

**Ciąg Fibonacciego:**

$$x_n = x_{n-1} + x_{n-2}$$

**LFG:**

$$x_n = (x_{n-j} @ x_{n-k}) \bmod m, \quad 0 < j < k$$

Gdzie:

- $x_i$  – kolejna wygenerowana liczba,
- $m$  – zakres generowanych liczb od 0 do  $m - 1$ ,
- $@$  – dowolne działanie np. dodawanie, odejmowanie.

Algorytm wykorzystuje do działania tablicę liczb o długości  $k$ . Wartości  $j$  i  $k$  są tak ustalone, by wynikiem generatora był najdłuższy możliwy cykl wartości.

Obliczenie kolejnej wartości  $x_i$  polega na dodaniu do siebie wartości z tablicy  $x[ ]$  o indeksach  $j - 1$  i  $k - 1$ . Następnie indeks elementu tablicy przesuwany jest w lewo o jedno miejsce, a na ostatniej pozycji umieszczana jest wyliczona wartość pseudolosowa.

*Dane:*

- $j$  i  $k$  – współczynniki,
- $m$  – zakres generowanych liczb od 0 do  $m - 1$ ,
- $x[k]$  –  $k$  – elementowa tablica,
- $n$  – ilość liczb do wygenerowania.

```
1 ile = 0
2 dopóki ile < n wykonuj:
3     dla i = 0, 1, 2, ..., k - 1 wykonuj:
4         jeżeli x[0] == 0 wykonaj:
5             wynik = (x[j - 1] @ x[k - 1]) mod m
```



```
6   jeżeli  $i < k - 1$  wykonaj:  
7        $x[i] = x[i + 1]$   
8   w przeciwnym wypadku wykonaj:  
9        $x[i] = \text{wynik}$   
10      wypisz wynik
```

## Polecenie 1

Wykorzystując algorytm LFG, wygeneruj 7 wartości pseudolosowych z przedziału od 0 do 14 dla podanej tablicy  $x[ ]$  oraz współczynników  $j$  oraz  $k$ .

Swój program przetestuj dla następujących wartości:

```
1 j = 3
2 k = 7
3 x[k] = {4, 2, 9, 1, 10, 7, 3}
```

### Specyfikacja problemu:

*Dane:*

- $j$  – liczba naturalna; pierwszy ze współczynników
- $k$  – liczba naturalna; drugi ze współczynników
- $x[k]$  –  $k$ -elementowa tablica liczb naturalnych

*Wynik:*

Program wyświetla liczby wygenerowane za pomocą algorytmu LFG.

## Polecenie 2

Porównaj swoje rozwiązanie z prezentacją.



W arkuszu egzaminacyjnym znajdą się zarówno zadania zamknięte, jak i otwarte, a także praktyczne wymagające użycia komputera i

zapisania komputerowej realizacji rozwiązania, np. programu w wybranym języku programowania.

Źródło: Pixabay, CC 0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P14NpkVog>

Naszym zadaniem będzie wypisanie kolejnych wygenerowanych 7 liczb pseudolosowych z zakresu od 0 do 14. Użyjemy do tego celu algorytmu LFG.

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P14NpkVog>

Ponieważ przedział wartości, jakie mamy wylosować, wynosi od 0 do 14, oznacza to, że wartość współczynnika  $m$  wynosi 15.

3

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P14NpkVog>

Obliczmy teraz pierwszą wartość wygenerowanego ciągu:

|

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P14NpkVog>

Obliczmy teraz drugą wartość wygenerowanego ciągu:

|

5

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P14NpkVog>

Obliczmy teraz trzecią wartość  
wygenerowanego ciągu:

|

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P14NpkVog>

Obliczmy teraz czwartą wartość  
wygenerowanego ciągu:

|

7

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P14NpkVog>

Obliczmy teraz piątą wartość wygenerowanego  
ciągu:

|

8

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P14NpkVog>

Obliczmy teraz szóstą wartość  
wygenerowanego ciągu:

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P14NpkVog>

Obliczmy teraz siódmą wartość  
wygenerowanego ciągu:



Więcej zadań znajdziesz w zbiorze dostępnym na  
stronie internetowej Okręgowej Komisji  
Egzaminacyjnej w Warszawie.

Źródło: OKE, [oke.waw.pl](http://oke.waw.pl), tylko do użytku edukacyjnego.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P14NpkVog>

Ponieważ zmienna  $i_l e$  jest większa od  
zmiennej  $n$ , wystarczy wypisać otrzymane  
wartości.

Wynik: 12, 13, 8, 0, 3, 0, 13

### Polecenie 3

Dodaj do swojego programu komentarze tak, żeby był zrozumiały dla osoby, która nie potrafi programować.

# Sprawdź się

---

**Materiał źródłowy do ćwiczeń nr 1-3.**

Algorytm LCG:

- $a = 102$ ,
- $x_0 = 0$ .

Algorytm LFG:

- $j = 2$ ,
- $k = 15$ .

*Dane:*

- $a$  – mnożnik,
- $x_0$  – ziarno,
- $j, k$  – współczynniki.

Pokaż ćwiczenia:   

## Ćwiczenie 1



### C++

Napisz program, który na początku wygeneruje  $n_1$ -elementową tablicę wartości pseudolosowych z zakresu od 1 do  $m_1$  włącznie, wykorzystując do tego multiplikatywny algorytm LCG. Na podstawie wcześniej stworzonej tablicy wygeneruje  $n_2$  liczb pseudolosowych z zakresu od 0 do  $m_2$  włącznie, wykorzystując algorytm LFG z odejmowaniem. Wypisz uzyskane  $n_2$  liczb, oddzielając je spacjami.

Przetestuj jego działanie dla następujących wartości:

```
1 n1 = 15
2 m1 = 50
3 n2 = 20
4 m2 = 41
```

### Specyfikacja:

*Dane:*

- $n_1$  – liczba naturalna; liczba liczb pseudolosowych wygenerowanych przy pomocy multiplikatywnego algorytmu LCG
- $m_1$  – liczba naturalna; największa liczba wygenerowana przy pomocy multiplikatywnego algorytmu LCG; włącza się do zakresu
- $n_2$  – liczba naturalna; liczba liczb pseudolosowych wygenerowanych przy pomocy algorytmu LFG z odejmowaniem
- $m_2$  – liczba naturalna; największa liczba wygenerowana przy pomocy algorytmu LFG z odejmowaniem; włącza się do zakresu

*Wynik:*

Program na standardowym wyjściu wypisuje wygenerowane liczby rozdzielając je spacjami.



## Twoje zadania

1. Napisz w języku C++ program generujący ciąg liczb pseudolosowych za pomocą algorytmów LCG i LFG.

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     //parametry LCG
5     int a = 102, x0 = 1;
6
7     //parametry LFG
8     int j = 2, k = 15;
9 }
```

1



### Java

Napisz program, który na początku wygeneruje  $n_1$ -elementową tablicę wartości pseudolosowych z zakresu od 1 do  $m_1$  włącznie, wykorzystując do tego multiplikatywny algorytm LCG. Na podstawie wcześniej stworzonej tablicy wygeneruje  $n_2$  liczb pseudolosowych z zakresu od 0 do  $m_2$  włącznie, wykorzystując algorytm LFG z odejmowaniem. Wypisz uzyskane  $n_2$  liczb, oddzielając je spacjami.

Przetestuj jego działanie dla następujących wartości:

```
1 n1 = 15
2 m1 = 50
3 n2 = 20
4 m2 = 41
```

### Specyfikacja:

*Dane:*

- $n_1$  – liczba naturalna; liczba liczb pseudolosowych wygenerowanych przy pomocy multiplikatywnego algorytmu LCG
- $m_1$  – liczba naturalna; największa liczba wygenerowana przy pomocy multiplikatywnego algorytmu LCG; włącza się do zakresu
- $n_2$  – liczba naturalna; liczba liczb pseudolosowych wygenerowanych przy pomocy algorytmu LFG z odejmowaniem
- $m_2$  – liczba naturalna; największa liczba wygenerowana przy pomocy algorytmu LFG z odejmowaniem; włącza się do zakresu

*Wynik:*

Program na standardowym wyjściu wypisuje wygenerowane liczby rozdzielając je spacjami.

## Twoje zadania

1. Napisz w języku Java program generujący ciąg liczb pseudolosowych za pomocą algorytmów LCG i LFG.

```
1 public class main {  
2     public static void main(String[] args) {  
3  
4         //parametry LCG  
5         int a = 102, x0 = 1;  
6  
7         //parametry LFG  
8         int j = 2, k = 15;  
9  
10    }
```

```
1
```



### Python

Napisz program, który na początku wygeneruje  $n_1$ -elementową tablicę wartości pseudolosowych z zakresu od 1 do  $m_1$  włącznie, wykorzystując do tego multiplikatywny algorytm LCG. Na podstawie wcześniej stworzonej tablicy wygeneruje  $n_2$  liczb pseudolosowych z zakresu od 0 do  $m_2$  włącznie, wykorzystując algorytm LFG z odejmowaniem. Wypisz uzyskane  $n_2$  liczb, oddzielając je spacjami.

Przetestuj jego działanie dla następujących wartości:

```
1 n1 = 15
2 m1 = 50
3 n2 = 20
4 m2 = 41
```

### Specyfikacja:

*Dane:*

- $n_1$  – liczba naturalna; liczba liczb pseudolosowych wygenerowanych przy pomocy multiplikatywnego algorytmu LCG
- $m_1$  – liczba naturalna; największa liczba wygenerowana przy pomocy multiplikatywnego algorytmu LCG; włącza się do zakresu
- $n_2$  – liczba naturalna; liczba liczb pseudolosowych wygenerowanych przy pomocy algorytmu LFG z odejmowaniem
- $m_2$  – liczba naturalna; największa liczba wygenerowana przy pomocy algorytmu LFG z odejmowaniem; włącza się do zakresu

*Wynik:*

Program na standardowym wyjściu wypisuje wygenerowane liczby rozdzielając je spacjami.

## Twoje zadania

1. Napisz w języku Python program generujący ciąg liczb pseudolosowych za pomocą algorytmów LCG i LFG.

```
1 # parametry LCG
2 a = 102
3 x0 = 1
4
5 # parametry LFG
6 j = 2
7 k = 15
```

```
1
```

# Dla nauczyciela

---

**Autor:** Maurycy Gast

**Przedmiot:** Informatyka

**Temat:** Generowanie liczb pseudolosowych – zadania maturalne

**Grupa docelowa:**

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy i rozszerzony

**Podstawa programowa:**

Cele kształcenia – wymagania ogólne

- I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.
- III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi, w tym: znajomość zasad działania urządzeń cyfrowych i sieci komputerowych oraz wykonywania obliczeń i programów.

Treści nauczania – wymagania szczegółowe

- I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

- 1) planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania).
- 3) wyróżnia w problemie podproblemy i charakteryzuje: metodę połowienia, stosuje podejście zachłanne i rekurencję;
- 4) porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji;
- 5) sprawdza poprawność działania algorytmów dla przykładowych danych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

- 1) w zależności od problemu rozwiązuje go, stosując metodę wstępującą lub zstępującą;

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres podstawowy. Uczeń:

- 1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);
- 2) do realizacji rozwiązań problemów prawidłowo dobiera środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki;

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

- 1) projektuje i tworzy rozbudowane programy w procesie rozwiązywania problemów, wykorzystuje w programach dobrane do algorytmów struktury danych, w tym struktury dynamiczne i korzysta z dostępnych bibliotek dla tych struktur;
- 3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

### **Kształtowane kompetencje kluczowe:**

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

### **Cele operacyjne (językiem ucznia):**

- Wymienisz algorytmy wykorzystywane do generowania liczb pseudolosowych.
- Przeanalizujesz rozwiązania zadań typu maturalnego.
- Samodzielnie napiszesz generator liczb pseudolosowych.

### **Strategie nauczania:**

- konstruktywizm;

- konektywizm.

### **Metody i techniki nauczania:**

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych.

### **Formy pracy:**

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

### **Środki dydaktyczne:**

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji);
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

### **Przebieg lekcji**

#### **Przed lekcją:**

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Generowanie liczb pseudolosowych – zadania maturalne”. Nauczyciel prosi uczniów o zapoznanie się z multimediu w sekcji „Prezentacja multimedialna” w odniesieniu do programowania.

#### **Faza wstępna:**

1. Przedstawienie tematu zajęć oraz wspólne z uczniami ustalenie kryteriów sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji oraz w odniesieniu do poznanych języków programowania.

#### **Faza realizacyjna:**

1. **Praca z tekstem.** Jeżeli przygotowanie uczniów do lekcji jest niewystarczające, nauczyciel prosi o indywidualne zapoznanie się z treścią zawartą w sekcji „Przeczytaj”.



Każdy uczestnik zajęć podczas cichego czytania wynotowuje najważniejsze kwestie poruszane w tekście.

2. **Praca z multimediami.** Chętny lub wybrany uczeń przedstawia swoje rozwiązanie zadania z sekcji „Prezentacja multimedialna”. Następnie uczniowie oglądają prezentację i wspólnie sprawdzają poprawność rozwiązania.
3. **Ćwiczenia umiejętności.** Uczniowie wykonują ćwiczenia z sekcji „Sprawdź się”. Ich zadaniem jest napisanie programu, który na początku wygeneruje 15-elementową tablicę wartości pseudolosowych z zakresu od 1 do 49, wykorzystując do tego multiplikatywny algorytm LCG. Na podstawie wcześniej stworzonej tablicy wygeneruje 20 liczb pseudolosowych, z zakresu od 0 do 40 wykorzystując algorytm LFG z odejmowaniem.  
Ćwiczenie realizują w jednym z dostępnych języków programowania.

#### **Faza podsumowująca:**

1. Nauczyciel prosi uczniów o podsumowanie zgromadzonej wiedzy w zakresie programowania.

#### **Praca domowa:**

1. Uczniowie dodają do swoich rozwiązań z sekcji „Sprawdź się” komentarze tak, by nawet osoba, która nie zna programowania mogła zrozumieć zachodzące procesy.
2. Uczniowie wykonują polecenie 3 z sekcji „Prezentacja multimedialna”.

#### **Materiały pomocnicze:**

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.
- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

#### **Wskazówki metodyczne:**

- Uczniowie mogą wykorzystać treści w sekcjach: „Przeczytaj”, „Prezentacja multimedialna”, „Sprawdź się” jako materiał do lekcji powtórkowej.