INFORMATYKA

Poziom rozszerzony

Egzamin maturalny – Formuła 2023 · arkusz próbny nr 1

Imię i nazwisko: .....

WYBRANE:

.....
(system operacyjny)

.....
(program użytkowy)

.....
(środowisko programistyczne)

CZAS TRWANIA:	210 minut
LICZBA PUNKTÓW DO UZYSKANIA:	50
ZAŁĄCZNIK:	katalog DANE (archiwum matura-informatyka-pr-arkusz-1-dane.zip)

Arkusz przygotowany według zasad egzaminu maturalnego z informatyki w Formule 2023 (budowa arkusza, rodzaje zadań i sposób oddawania rozwiązań jak w arkuszach CKE 2023–2026; oprogramowanie zgodnie z komunikatem dyrektora CKE z 20 sierpnia 2026 r.). Zadania sprawdzają wymagania podstawy programowej w brzmieniu z Dz.U. 2024 poz. 1019. To nie jest arkusz CKE; treści, dane i historie są oryginalne. Materiał nie jest związany z CKE ani z producentami oprogramowania – nazwy programów, systemów i instytucji podano wyłącznie informacyjnie.

Instrukcja dla zdającego

1. Zanim zaczniesz, sprawdź, czy arkusz ma 19 stron (zadania 1–8) i czy masz katalog **DANE** z plikami do zadań (archiwum **matura-informatyka-pr-arkusz-1-dane.zip**). Jeśli czegoś brakuje, powiedz o tym nauczycielowi.
2. Na pierwszej stronie wpisz imię i nazwisko oraz wybrane przez siebie (na egzaminie – zadeklarowane) oprogramowanie: system operacyjny, program użytkowy i środowisko programistyczne. Według komunikatu dyrektora CKE z 20 sierpnia 2026 r. w 2027 r. do wyboru są: system **Windows** z pakietem Microsoft Office 2007 lub nowszym (Word, Excel, Access) albo z LibreOffice 7.0 lub nowszym (Writer, Calc, Base) lub OpenOffice/Apache OpenOffice 4.1 lub nowszym, albo system **Linux** z LibreOffice 7.0 lub nowszym lub OpenOffice/Apache OpenOffice 4.1 lub nowszym – zawsze razem z pakietem XAMPP (Apache, MySQL/MariaDB, PHP, phpMyAdmin) – oraz jeden język programowania: C/C++, Java (Java SE 8 lub nowsza) albo Python (3.10 lub nowszy).
3. Zadanie z symbolem ✎ w nagłówku rozwiązujesz bez komputera – odpowiedź wpisz wyłącznie w arkuszu, w miejscu na nią przeznaczonym.
4. Programy pisz w wybranym języku programowania. Wszystkie utworzone przez siebie pliki źródłowe zapisz w katalogu nazwanym Twoim imieniem i nazwiskiem (na egzaminie katalog oznacza się numerem PESEL).
5. Jeżeli algorytm zapisywany w arkuszu przedstawiasz w języku programowania, a nie w pseudokodzie, użyj języka zadeklarowanego na egzamin (C++, Java lub Python).
6. Przy zadaniu z bazą danych w MySQL lub MariaDB zapisz w katalogu treści zapytań SQL, a przed końcem pracy wyeksportuj całą bazę do pliku *.sql. W programie Access lub Base oddajesz plik bazy z kwerendami nazwanymi tak, jak podano w zadaniu.
7. Każdy plik do oceny nazwij dokładnie tak, jak podano w zadaniu (rozszerzenie zależy od używanego programu) – pliki nazwane inaczej nie są sprawdzane. Zadanie praktyczne bez komputerowego rozwiązania otrzymuje 0 punktów.
8. Przed końcem czasu pracy zapisz w katalogu ostateczne wersje plików z rozwiązaniami.
9. Pisz czytelnie, długopisem lub piórem z czarnym tuszem (atramentem). Nie używaj korektora – błędny zapis wyraźnie przekreśl. Zapisów w brudnopisie nikt nie ocenia.
10. Kratki z punktacją przy zadaniach wypełnia egzaminator – nic w nich nie wpisuj.
11. Na egzaminie wtyczki AI generujące kod lub treść (np. Copilot) są w środowiskach programistycznych i w pakiecie biurowym odinstalowane albo wyłączone – rozwiązuj arkusz bez nich.
12. Programy do zadań 3. i 4. możesz dodatkowo sprawdzić na stronie **pierwszalawka.pl/kod-na-maturze** – wpisz kod zadania podany przy poleceniu. Narzędzie sprawdza wynik danego zadania (z numerem zadania albo bez), wypisany na ekran albo zapisany w pliku wyniki*.txt, na danych z arkusza i na dwóch innych zestawach danych.

PIERWSZA ŁAWKA

Zadanie 1. Funkcja rekurencyjna

Dana jest funkcja rekurencyjna B, której argumentami są dodatnie liczby całkowite *a* i *b*. Instrukcja **zwróć** kończy działanie funkcji i podaje jej wynik; *div* oznacza dzielenie całkowite, a *mod* – resztę z dzielenia.

```
funkcja B(a, b)
  jeżeli a = b, to zwróć a
  jeżeli a mod 2 = 0 oraz b mod 2 = 0, to zwróć 2 · B(a div 2, b div 2)
  jeżeli a mod 2 = 0, to zwróć B(a div 2, b)
  jeżeli b mod 2 = 0, to zwróć B(a, b div 2)
  jeżeli a > b, to zwróć B(a – b, b)
  zwróć B(a, b – a)
```

Na przykład obliczenie B(12, 18) przebiega tak: B(12, 18) = 2 · B(6, 9) = 2 · B(3, 9) = 2 · B(3, 6) = 2 · B(3, 3) = 2 · 3 = 6. Funkcja B została wywołana 5 razy (łącznie z pierwszym wywołaniem).

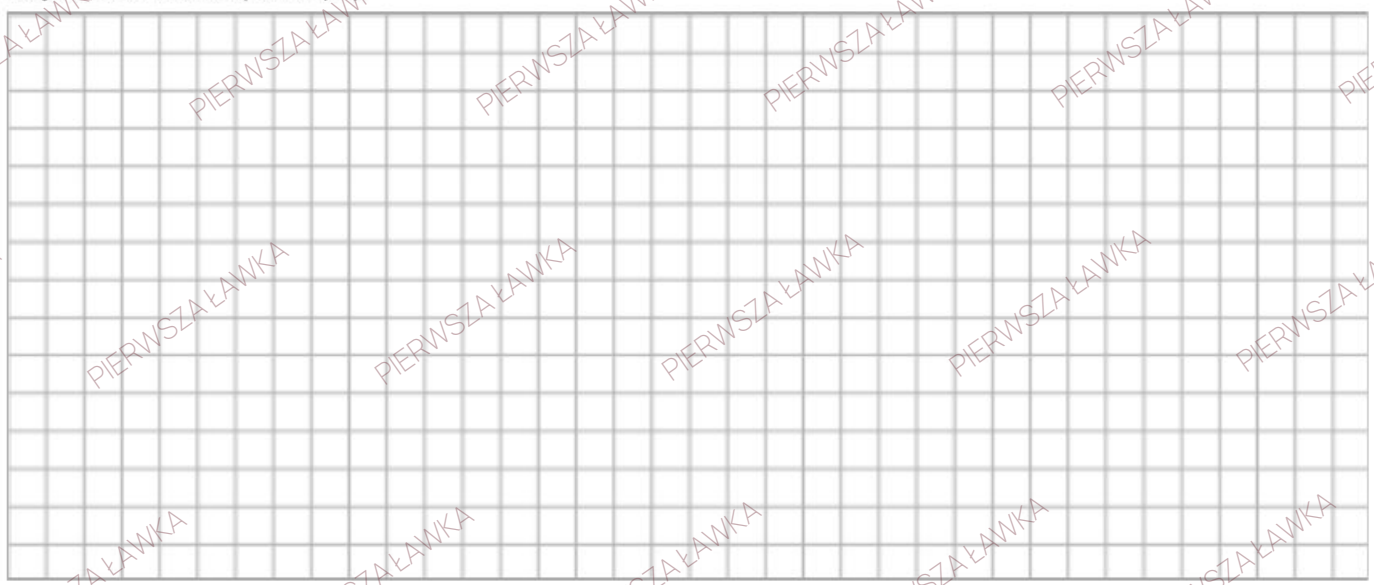
Zadanie 1.1. (0–2)

1.1.
0–1–2

Uzupełnij tabelę. Wpisz wynik funkcji B dla podanych argumentów oraz liczbę wywołań funkcji B (łącznie z pierwszym wywołaniem).

a	*b*	wynik B(*a*, *b*)	liczba wywołań funkcji B
12	18	6	5
40	24		
27	45		
1	64		

Miejsce na obliczenia (brudnopis)



Zadanie 2.2. (0–4)

2.2.
0–1–2–3–4

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który dla danych ciągów *A* i *B* obliczy liczbę liczb występujących w obu ciągach.

Uwaga: Twój algorytm może przejrzeć każdy z ciągów tylko raz – łączna liczba wykonań pętli (wszystkich pętli razem) nie może przekroczyć $n + m$; nie wolno więc porównywać każdego elementu ciągu *A* z każdym elementem ciągu *B*. W zapisie algorytmu możesz korzystać tylko z operacji arytmetycznych, porównań, instrukcji sterujących, przypisania i odwołań do elementów tablic. Zabronione jest używanie funkcji wbudowanych (np. wyszukujących, sortujących), zbiorów i słowników.

Specyfikacja:

Dane:

n, m – dodatnie liczby całkowite

A[1..n] – ciąg liczb całkowitych uporządkowany ściśle rosnąco

B[1..m] – ciąg liczb całkowitych uporządkowany ściśle rosnąco

Wynik:

w – liczba liczb, które występują zarówno w ciągu *A*, jak i w ciągu *B*

Miejsce na zapis algorytmu

Zadanie 3. Liczby

W pliku **liczby.txt** znajduje się 1000 różnych liczb naturalnych z przedziału [2, 1 000 000], każda w osobnym wierszu.

Pierwsze wiersze pliku liczby.txt:

```
826504
315496
958243
715395
257376
```

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku **wyniki3.txt**. Każdą odpowiedź poprzedź numerem oznaczającym zadanie.

Do Twojej dyspozycji jest plik **liczby_przyklad.txt** w identycznym formacie. Odpowiedzi dla danych z tego pliku podano pod treściami zadań – możesz na nich sprawdzać działanie swojego programu.

Uwaga: Pamiętaj, że Twój program musi ostatecznie działać dla pliku **liczby.txt**.

Zadanie 3.1. (0–2)

3.1.
0–1–2

Podaj, ile liczb z pliku **liczby.txt** jest liczbami pierwszymi, oraz największą z tych liczb pierwszych.

Postać odpowiedzi: Dwie liczby w jednym wierszu: liczba liczb pierwszych i największa z nich.

Dla pliku **liczby_przyklad.txt** poprawna odpowiedź to:

```
2 39791
```

[kod zadania: inf-1-3a]

Zadanie 3.2. (0–4)

3.2.
0–1–2–3–4

Dla każdej liczby z pliku wyznacz liczbę jej różnych dzielników pierwszych (np. 360 = 2 · 2 · 2 · 3 · 3 · 5 ma trzy różne dzielniki pierwsze: 2, 3 i 5). Podaj największą z wyznaczonych wartości, a po niej – wszystkie liczby z pliku, które mają tyle różnych dzielników pierwszych, w kolejności ich występowania w pliku.

Postać odpowiedzi: Jeden wiersz: największa liczba różnych dzielników pierwszych, a po niej – liczby z pliku, oddzielone spacjami.

Dla pliku **liczby_przyklad.txt** poprawna odpowiedź to:

```
6 84630 72930 46410
```

[kod zadania: inf-1-3b]

Zadanie 3.3. (0–4)

3.3.
0–1–2–3–4

Dwie liczby są względnie pierwsze, jeżeli ich największy wspólny dzielnik jest równy 1. Znajdź najdłuższy spójny fragment ciągu liczb zapisanych w pliku (ciąg liczb z kolejnych wierszy), w którym każde dwie sąsiednie liczby są względnie pierwsze. Podaj długość tego fragmentu (liczbę jego elementów) oraz jego pierwszą liczbę. Jest jeden taki najdłuższy fragment.

Postać odpowiedzi: Dwie liczby w jednym wierszu: długość fragmentu i jego pierwsza liczba.

Przykład: w ciągu 14, 35, 12, 25, 9, 10, 4, 21 najdłuższy fragment o podanej własności to 35, 12, 25, 9, 10 – ma długość 5, a jego pierwszą liczbą jest 35 (NWD(14, 35) = 7, NWD(10, 4) = 2).

Dla pliku **liczby_przyklad.txt** poprawna odpowiedź to:

```
7 24579
```

[kod zadania: inf-1-3c]

Do oceny oddajesz:

- plik tekstowy **wyniki3.txt** zawierający odpowiedzi do zadań 3.1.–3.3. (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- pliki zawierające kody źródłowe Twoich programów o nazwach: **zad3_1** (zadanie 3.1.), **zad3_2** (zadanie 3.2.), **zad3_3** (zadanie 3.3.) – z rozszerzeniem zgodnym z językiem programowania (.py, .cpp lub .java); jeżeli jeden program rozwiązuje kilka zadań, zapisz go pod nazwą zawierającą numery tych zadań, np. **zad3_1_2** (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania).

PIERWSZA ŁAWKA

Zadanie 4. Notowania

W pliku **kurs.txt** zapisano kursy zamknięcia akcji pewnej spółki z 500 kolejnych dni, od 1 stycznia 2025 r. W każdym wierszu znajduje się data w formacie rrrr-mm-dd i kurs w groszach (liczba całkowita), oddzielone spacją. Dane są uporządkowane chronologicznie; każda data występuje w pliku dokładnie raz.

Pierwsze wiersze pliku kurs.txt:

```
2025-01-01 4203
2025-01-02 4152
2025-01-03 4272
2025-01-04 4427
2025-01-05 4291
```

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku **wyniki4.txt**. Każdą odpowiedź poprzedź numerem oznaczającym zadanie.

Do Twojej dyspozycji jest plik **kurs_przyklad.txt** w identycznym formacie. Odpowiedzi dla danych z tego pliku podano pod treściami zadań – możesz na nich sprawdzać działanie swojego programu.

Uwaga: Pamiętaj, że Twój program musi ostatecznie działać dla pliku **kurs.txt**.

Zadanie 4.1. (0–2)

4.1.
0–1–2

Podaj najniższy kurs i datę dnia, w którym wystąpił on po raz pierwszy, oraz najwyższy kurs i datę dnia, w którym wystąpił on po raz pierwszy.

Postać odpowiedzi: Jeden wiersz: najniższy kurs, jego data, najwyższy kurs, jego data – oddzielone spacjami.

Dla pliku **kurs_przyklad.txt** poprawna odpowiedź to:

```
3648 2026-03-16 4199 2026-03-01
```

[kod zadania: inf-1-4a]

Zadanie 4.2. (0–4)

4.2.
0–1–2–3–4

Okresem bez spadku nazywamy ciąg kolejnych dni, w którym kurs każdego dnia (poza pierwszym) jest większy od kursu z dnia poprzedniego lub mu równy – kursy tworzą spójny podciąg niemalejący. Znajdź najdłuższy okres bez spadku. Podaj liczbę dni tego okresu oraz datę jego pierwszego i ostatniego dnia. Jest jeden taki najdłuższy okres.

Postać odpowiedzi: Jeden wiersz: liczba dni, data pierwszego dnia, data ostatniego dnia.

Przykład: dla kursów z ośmiu kolejnych dni: 100, 102, 102, 101, 105, 107, 107, 99 najdłuższy okres bez spadku obejmuje dni od czwartego do siódmego (kursy 101, 105, 107, 107) i ma 4 dni.

Dla pliku **kurs_przyklad.txt** poprawna odpowiedź to:

```
5 2026-03-16 2026-03-20
```

[kod zadania: inf-1-4b]

Zadanie 4.3. (0–4)

4.3.
0–1–2–3–4

Inwestor kupuje akcję jednego dnia (po kursie z tego dnia) i sprzedaje ją jednego z późniejszych dni (po kursie z dnia sprzedaży). Podaj największy możliwy zysk na jednej akcji (w groszach) oraz datę kupna i datę sprzedaży, przy których ten zysk jest osiągany. Jest jedna taka para dni.

Postać odpowiedzi: Jeden wiersz: zysk w groszach, data kupna, data sprzedaży.

Przykład: dla kursów z sześciu kolejnych dni: 50, 42, 49, 40, 45, 44 największy zysk wynosi 7 – kupno drugiego dnia (po kursie 42), sprzedaż trzeciego dnia (po kursie 49).

Dla pliku **kurs_przyklad.txt** poprawna odpowiedź to:

```
284 2026-03-16 2026-03-20
```

[kod zadania: inf-1-4c]

Do oceny oddajesz:

- plik tekstowy **wyniki4.txt** zawierający odpowiedzi do zadań 4.1.–4.3. (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- pliki zawierające kody źródłowe Twoich programów o nazwach: **zad4_1** (zadanie 4.1.), **zad4_2** (zadanie 4.2.), **zad4_3** (zadanie 4.3.) – z rozszerzeniem zgodnym z językiem programowania (.py, .cpp lub .java); jeżeli jeden program rozwiązuje kilka zadań, zapisz go pod nazwą zawierającą numery tych zadań, np. **zad4_1_2** (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania).

Zadanie 7. Rower miejski

W pliku **rowery.txt** zapisano dane o 2400 wypożyczeniach rowerów miejskich w okresie od 1 kwietnia do 30 września 2026 r. Pierwszy wiersz pliku jest wierszem nagłówkowym, a dane w wierszach rozdzielono znakami tabulacji. W każdym wierszu zapisano:

- ***IdKlienta*** – numer klienta (liczba całkowita)
- ***Data*** – data wypożyczenia (rrrr-mm-dd)
- ***Godzina*** – godzina wypożyczenia (gg:mm)
- ***StacjaStart*** – nazwa stacji, na której wypożyczono rower
- ***StacjaKoniec*** – nazwa stacji, na której zwrócono rower
- ***CzasMin*** – czas wypożyczenia w minutach (liczba całkowita)

Przykład:

IdKlienta	Data	Godzina	StacjaStart	StacjaKoniec	CzasMin
34	2026-04-01	08:07	Rynek	Uniwersytet	26
73	2026-04-01	09:16	Dworzec	Stadion	12
73	2026-04-01	15:09	Park	Dworzec	24
73	2026-04-01	17:24	Dworzec	Dworzec	133

Z wykorzystaniem danych zawartych w pliku oraz dostępnych narzędzi informatycznych wykonaj podane zadania. Odpowiedzi zapisz w pliku **wyniki7.txt**. Odpowiedź do każdego zadania poprzedź numerem tego zadania.

Zadanie 7.1. (0–2)

7.1.
0–1–2

Podaj nazwę stacji, na której rozpoczęło się najwięcej wypożyczeń, i nazwę stacji, na której zakończyło się najwięcej wypożyczeń, wraz z odpowiednimi liczbami wypożyczeń.

Zadanie 7.2. (0–3)

7.2.
0–1–2–3

Wykonaj zestawienie liczby wypożyczeń w poszczególnych miesiącach. Na podstawie zestawienia utwórz wykres kolumnowy. Pamiętaj o czytelnym opisie wykresu: tytuł wykresu, opisy osi, nazwy miesięcy na osi X.

Zadanie 7.3. (0–2)

7.3.
0–1–2

Opłata za wypożyczenie jest naliczana według zasady: pierwsze 20 minut jest bezpłatne, a za każde rozpoczęte kolejne 15 minut płaci się 2 zł. Na przykład wypożyczenie trwające 20 minut jest bezpłatne, 21 minut i 35 minut – kosztuje 2 zł, 36 minut – 4 zł. Podaj łączną kwotę opłat za wszystkie wypożyczenia oraz liczbę wypożyczeń bezpłatnych.

Zadanie 7.4. (0–3)

7.4.
0–1–2–3

Operator rozważa wprowadzenie abonamentu miesięcznego w cenie 25 zł, z którym wszystkie wypożyczenia w danym miesiącu kalendarzowym są bezpłatne. Dla każdego miesiąca podaj liczbę klientów, dla których suma opłat za wypożyczenia w tym miesiącu (naliczonych jak w zadaniu 7.3.) była **większa niż** 25 zł. Podaj też łączną kwotę, jaką zaoszczędziliby ci klienci w całym okresie,

IdKursu – identyfikator kursu

DataZapisu – data zapisu (rrrr-mm-dd)

LiczbaMiesiecy – liczba opłaconych miesięcy nauki

Przykład (pierwsze wiersze pliku):

IdZapisu	IdKursanta	IdKursu	DataZapisu	LiczbaMiesiecy
1	47	38	2025-09-01	9
2	126	5	2025-09-01	2
3	198	3	2025-09-01	8

Z wykorzystaniem danych zawartych w plikach oraz dostępnych narzędzi informatycznych podaj odpowiedzi do zadań 8.1.–8.5. Odpowiedzi zapisz w pliku **wyniki8.txt**, a każdą z nich poprzedź numerem odpowiedniego zadania. Przychodem z zapisu nazywamy iloczyn ceny miesięcznej kursu i liczby opłaconych miesięcy.

Zadanie 8.1. (0–1)

8.1.
0–1

Podaj język, na którego kursy zapisano się najwięcej razy, oraz liczbę tych zapisów.

Zadanie 8.2. (0–2)

8.2.
0–1–2

Podaj imię i nazwisko lektora, którego kursy przyniosły szkole największy łączny przychód, oraz ten przychód.

Zadanie 8.3. (0–2)

8.3.
0–1–2

Podaj imiona i nazwiska kursantów, którzy nie zapisali się na żaden kurs. Zestawienie uporządkuj alfabetycznie według nazwisk, a przy takich samych nazwiskach – według imion.

Zadanie 8.4. (0–2)

8.4.
0–1–2

Podaj imiona i nazwiska kursantów, którzy zapisali się na kursy co najmniej trzech różnych języków. Zestawienie uporządkuj jak w zadaniu 8.3.

Zadanie 8.5. (0–2)

8.5.
0–1–2

Szkoła podniosła o 20 zł cenę miesięczną wszystkich kursów na poziomach B2 i C1. Zapisz w języku SQL zapytanie, które wprowadzi tę zmianę w tabeli z danymi z pliku kursy.txt, a następnie podaj średnią cenę miesięczną kursu (wszystkich kursów) po zmianie, z dokładnością do 0,01 zł.

Do oceny oddajesz:

- plik tekstowy **wyniki8.txt** zawierający odpowiedzi do zadań 8.1.–8.5. (odpowiedź do każdego zadania poprzedzona jego numerem)
- plik **zad8.sql** z treścią zapytań SQL do zadań 8.1.–8.5. (każde zapytanie poprzedzone komentarzem z numerem zadania, np. **-- 8.1**) oraz wyeksportowana cała baza **baza8.sql** (MySQL/MariaDB) albo plik bazy **zad8** (.accdb/.odb) z kwerendami o nazwach zad8_1 … zad8_5
- (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania)

CO JESZCZE JEST W ŚRODKU

Matura 2027 z informatyki – poziom rozszerzony. Arkusz próbny 1 z kluczem

Podgląd pokazał 10 z 33 stron. Pełna wersja: 33 stron A4, PDF do druku i DOCX do edycji.

- Arkusz próbny: Matura 2027 z informatyki – poziom rozszerzony, zestaw nr 1.
- Poziom rozszerzony · Formuła 2023 · arkusz próbny na maturę 2027 · rekurencyjne NWD, wspólne elementy ciągów, liczby pierwsze i dzielniki, notowania giełdowe, systemy liczbowe, sie
- Czas pracy: 210 minut · 8 zadań · 50 pkt · 33 strony A4.
- W pakiecie plików: arkusz w PDF (do druku), ten sam arkusz w DOCX (do edycji w Wordzie) oraz archiwum DANE z plikami .txt do zadań przy komputerze (z plikami *_przyklad.txt do spra
- Na końcu: zasady oceniania, klucz odpowiedzi i przykładowe realizacje zadań otwartych.
- Arkusz jest zbudowany jak matura CKE 2023–2026: 8 zadań, około 20% punktów bez komputera, odpowiedzi zapisywane w plikach wynikiN.txt, wiązka arkusza kalkulacyjnego z wykresem i wi
- Programy z zadań 3 i 4 możesz sprawdzić automatycznie na pierwszalawka.pl/kod-na-maturze – przy każdym poleceniu jest kod zadania.
- „Kod na maturze" uruchamia Twój program w Pythonie, C++ albo Javie na danych z arkusza i na dwóch dodatkowych zestawach.
- To nie jest arkusz CKE ani kopia arkusza z poprzednich lat – wszystkie teksty i zadania są oryginalne, przygotowane zgodnie z wymaganiami egzaminacyjnymi na rok 2027.
- Dzięki temu uczeń nie zna zadań z internetu, a wynik rzetelnie pokazuje stan przygotowania.

Kup pełną wersję – plik od razu na e-mail

pierwszalawka.pl/material/matura-informatyka-pr-2027-arkusz-probny-1