CENTRALNA KOMISJA EGZAMINACYJNA

Arkusz zawiera informacje prawnie chronione do momentu rozpoczęcia egzaminu.

WYPEŁNIA ZDAJĄCY

KOD | **PESEL**

Miejsce na naklejkę.
Sprawdź, czy kod na naklejce to **E-100**.
Jeżeli tak – przyklej naklejkę.
Jeżeli nie – zgłoś to nauczycielowi.

Egzamin maturalny ***Formuła 2015***

INFORMATYKA

Poziom rozszerzony

Część I

WYPEŁNIA ZDAJĄCY

WYBRANE:

.....
(system operacyjny)

.....
(program użytkowy)

.....
(środowisko programistyczne)

Symbol arkusza
EINP-R1-**100**-2605

DATA: **14 maja 2026 r.**

GODZINA ROZPOCZĘCIA: **9:00**

CZAS TRWANIA: **60 minut**

LICZBA PUNKTÓW DO UZYSKANIA: **15**

Przed rozpoczęciem pracy z arkuszem egzaminacyjnym

1. Sprawdź, czy nauczyciel przekazał Ci **właściwy arkusz egzaminacyjny**, tj. arkusz we **właściwej formule**, z **właściwego przedmiotu** na **właściwym poziomie**.
2. Jeżeli przekazano Ci **niewłaściwy** arkusz – natychmiast zgłoś to nauczycielowi. Nie rozrywaj banderol.
3. Jeżeli przekazano Ci **właściwy** arkusz – rozerwij banderole po otrzymaniu takiego polecenia od nauczyciela. Zapoznaj się z instrukcją na stronie 2.

Układ graficzny
© CKE 2022

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 11 stron (zadania 1–3). Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
2. Na pierwszej stronie arkusza oraz na karcie odpowiedzi wpisz swój numer PESEL i przyklej naklejkę z kodem.
3. Wpisz zadeklarowane (wybrane) przez Ciebie na egzamin: system operacyjny, program użytkowy oraz środowisko programistyczne.
4. Odpowiedzi i rozwiązania zapisz w miejscu na to przeznaczonym przy każdym zadaniu.
5. Pisz czytelnie. Używaj długopisu/pióra tylko z czarnym tuszem/atramentem.
6. Nie używaj korektora, a błędne zapisy wyraźnie przekreśl.
7. Pamiętaj, że zapisy w brudnopisie nie będą oceniane.
8. Nie wpisuj żadnych znaków w części przeznaczonej dla egzaminatora.
9. Możesz korzystać z kalkulatora prostego.

Zadania egzaminacyjne są wydrukowane na następnych stronach.

Zadanie 1. Rekurencja

Dana jest zdefiniowana rekurencyjnie funkcja $A(m, n)$, gdzie m i n są dodatnimi liczbami całkowitymi.

$$A(m, n) = \begin{cases} m & \text{gdy } n = 1 \\ A\left(2 \cdot m, \frac{n}{2}\right) & \text{gdy } n > 1 \text{ oraz } n \text{ jest podzielne przez 2} \\ 2 \cdot A\left(m, \frac{n-1}{2}\right) + m & \text{gdy } n > 1 \text{ oraz } n \text{ nie jest podzielne przez 2} \end{cases}$$

Zadanie 1.1. (0–4)

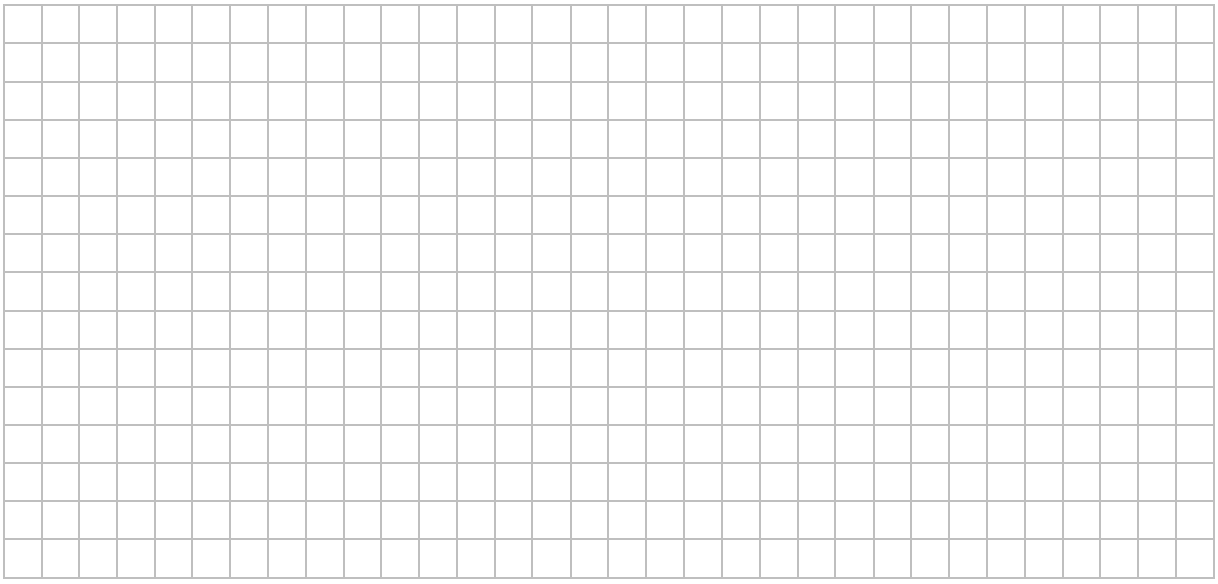
Obliczenie wartości funkcji $A(3, 9)$ wprost z definicji wymaga trzech wywołań rekurencyjnych: $A(3, 4)$, $A(6, 2)$, $A(12, 1)$, ponieważ:

$$A(3, 9) = 2 \cdot A(3, 4) + 3 = 2 \cdot A(6, 2) + 3 = 2 \cdot A(12, 1) + 3 = 2 \cdot 12 + 3 = 27$$

Uzupełnij poniższą tabelę. Podaj liczbę wywołań rekurencyjnych funkcji A oraz wypisz wywołania rekurencyjne wraz z ich argumentami (w ostatnim wierszu podaj tylko liczbę wywołań rekurencyjnych).

m	***n***	**liczba wywołań rekurencyjnych funkcji *A***	**wywołania rekurencyjne funkcji *A***
3	9	3	$A(3, 4)$, $A(6, 2)$, $A(12, 1)$
32	17		
2^5	2^5		
10	15		
1	$2^{100} + 1$		

Miejsce na obliczenia (brudnopis)



Zadanie 1.2. (0–2)

Uzupełnij poniższą tabelę. Podaj wartości funkcji $A(m, n)$ dla zadanych argumentów m i n:

m	n	$A(m, n)$
1	777	
$2 \cdot 10^9$	$256 \cdot 10^9$	

Uwaga: W swoich odpowiedziach możesz zapisać wynik podobnie jak wartości w pierwszych dwóch kolumnach (z wykorzystaniem operatorów mnożenia, potęgowania i dodawania).

Miejsce na obliczenia (brudnopis)

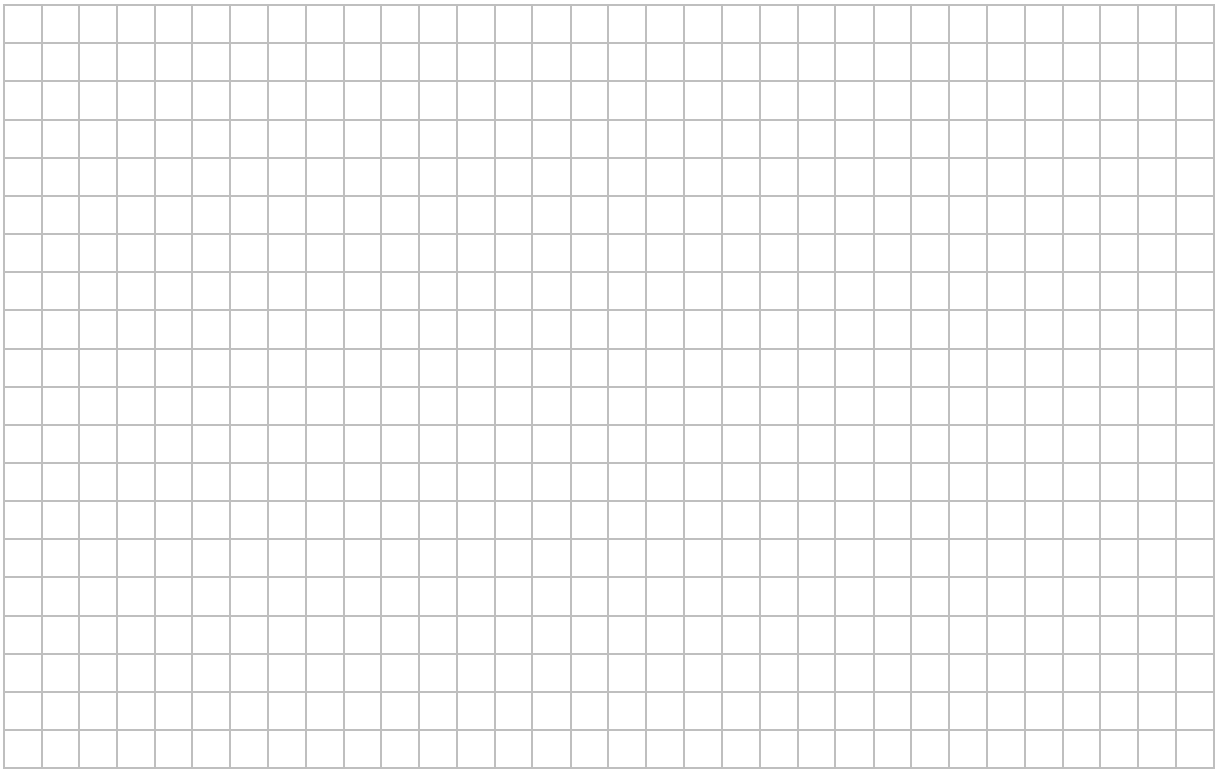
Zadanie 1.3. (0–3)

Uzupełnij tabelę. W drugiej kolumnie podaj liczbę wywołań rekurencyjnych funkcji *A* dla każdej wartości *n* podanej w tabeli (drugiego argumentu wywołania funkcji, pierwszy jest nieistotny w tym zadaniu). W trzeciej kolumnie podaj wyrażenie, którego wartość jest równa drugiemu argumentowi funkcji w *i*-tym wywołaniu rekurencyjnym dla wszystkich wartości *i* większych bądź równych 1 i mniejszych bądź równych całkowitej liczbie wywołań.

***n* – drugi argument wywołania funkcji**	**liczba wywołań rekurencyjnych**	**wartość drugiego argumentu *A* w *i*-tym wywołaniu rekurencyjnym**
8	3	$\frac{8}{2^i}$ (lub 2^{3-i})
2^k		
$2^k - 1$		

gdzie *k* jest pewną liczbą całkowitą dodatnią większą od 2.

Miejsce na obliczenia (brudnopis)



Zadanie 2. Dodawanie

Rozważamy dodawanie pisemne dwóch liczb zapisanych w systemie dziesiętnym, zilustrowane na przykładzie.

Przeniesienie:		***1***	***1***	***1***		***1***	
Liczba *a*:			2	7	7	3	2
Liczba *b*:	+		7	2	6	1	9
		1	0	0	3	5	1

W tym przykładzie mamy 4 przeniesienia.

Zadanie 2.1. (0–1)

Dla danych dwóch liczb: *a* i *b*, podaj, ile razy pojawiają się przeniesienia podczas ich dodawania.

Liczba *a*	**Liczba *b***	**Liczba przeniesień**
37932	12528	3
88765	11111	
456789	222222	

Miejsce na obliczenia (brudnopis)

Zadanie 2.2. (0–4)

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który dla danych dwóch liczb całkowitych dodatnich *a* i *b*, o tej samej liczbie cyfr w zapisie dziesiętnym, obliczy liczbę przeniesień otrzymanych w trakcie ich dodawania pisemnego.

Przykład: dla liczb 27732 i 72619 wynik to 4.

Uwaga: Twój algorytm może operować **wyłącznie na liczbach całkowitych** i używać tylko zmiennych przechowujących **pojedyncze** liczby całkowite. W zapisie algorytmu możesz korzystać tylko z operatorów arytmetycznych: dodawania, odejmowania, mnożenia, dzielenia, dzielenia całkowitego i reszty z dzielenia; z operatorów logicznych, porównań, instrukcji sterujących, instrukcji przypisania lub samodzielnie napisanych funkcji i procedur wykorzystujących powyższe operacje. **Zabronione** jest używanie funkcji wbudowanych oraz operatorów innych niż wymienione, dostępnych w językach programowania, nie wolno zwłaszcza korzystać z tablic/list oraz żadnych funkcji zamiany z typu znakowego lub napisowego na liczbowy – i odwrotnie.

Specyfikacja:

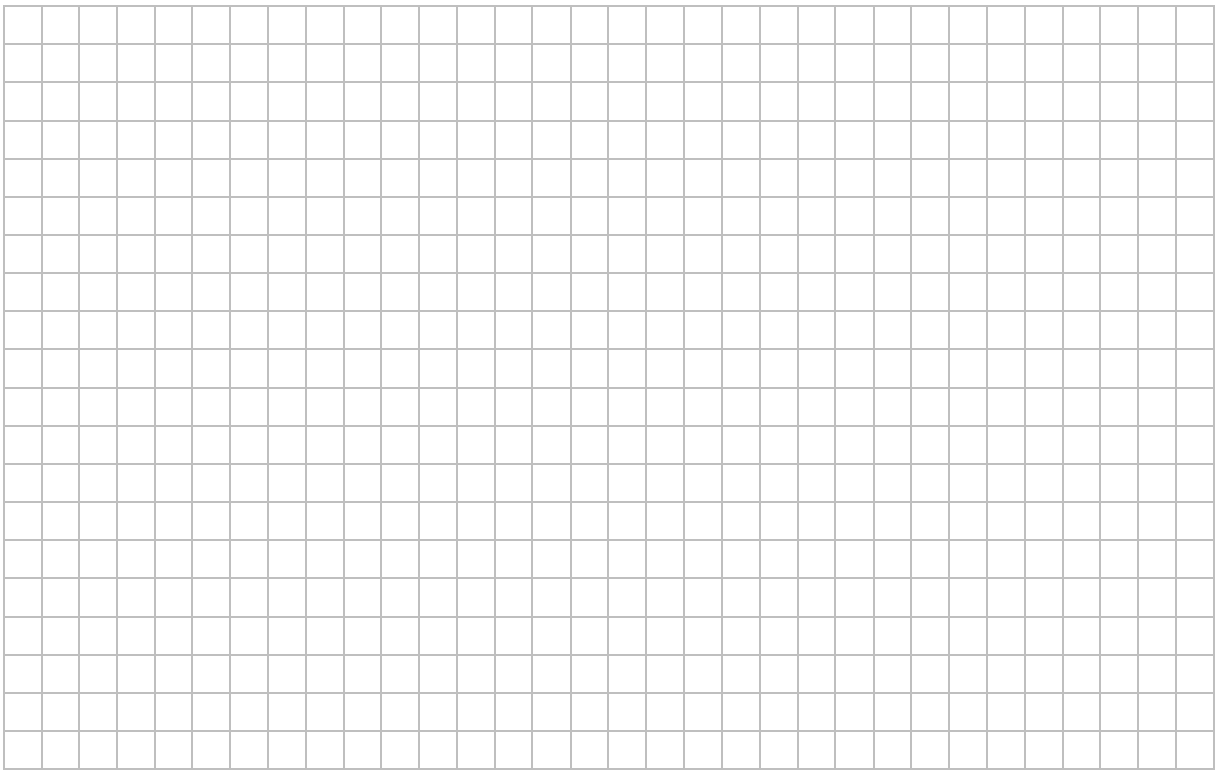
Dane:

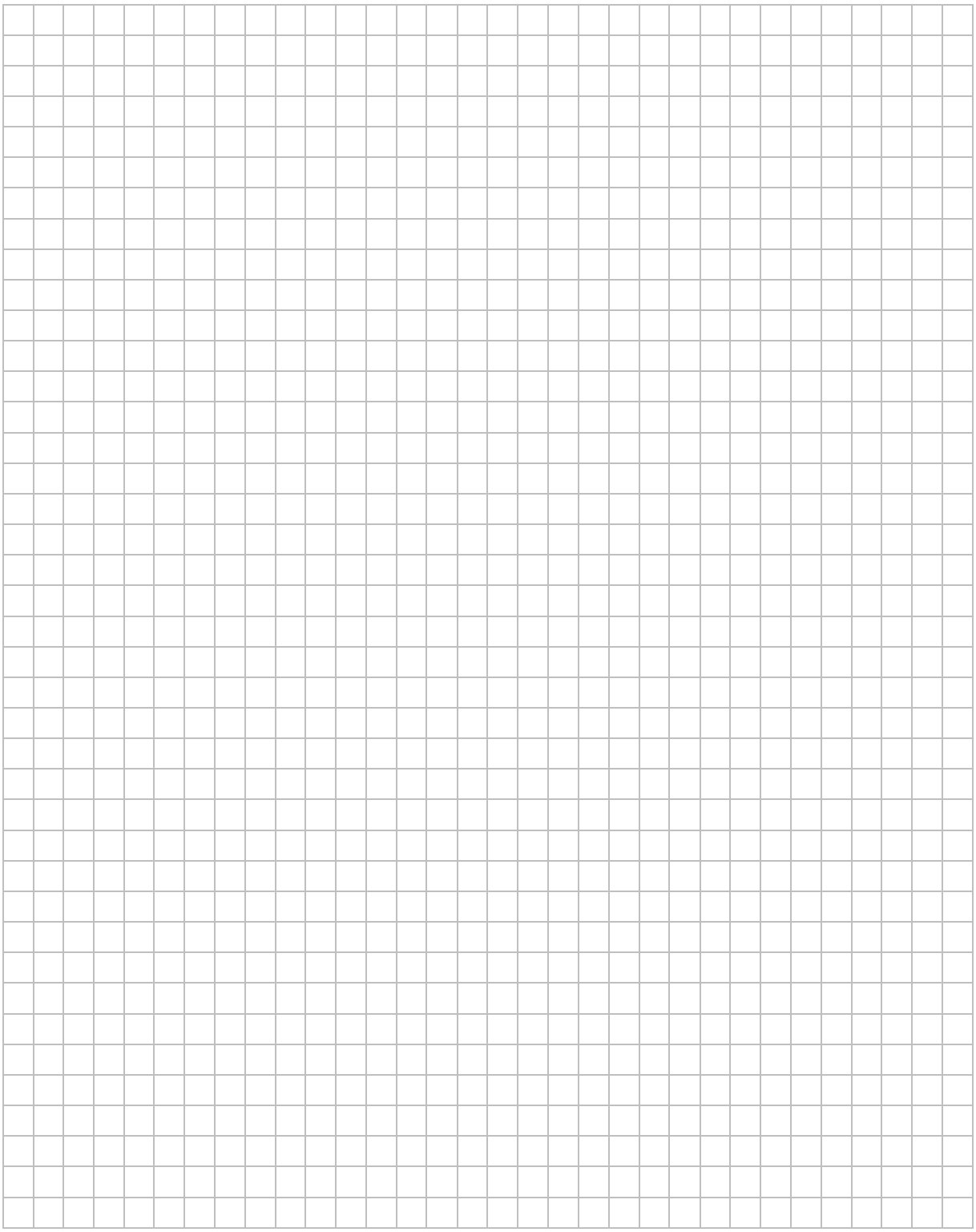
a, *b* – dodatnie liczby całkowite o tej samej liczbie cyfr

Wynik:

p – liczba przeniesień otrzymanych w trakcie dodawania pisemnego liczb *a* i *b*

Miejsce na zapis algorytmu





Zadanie 3. (0–1)

Oceń prawdziwość podanych zdań. Zaznacz **P**, jeśli zdanie jest prawdziwe, albo **F** – jeśli jest fałszywe. Punkt uzyskasz tylko za komplet poprawnych odpowiedzi.

Dane są tabele *Osoby* i *Samochody*, zawierające dane dotyczące samochodów i ich właścicieli.

Tabela *Osoby*

Id_Osoby	Imie	Nazwisko
1	Janusz	Warecki
2	Wojciech	Namyslo
3	Beata	Kosińska
4	Ewelina	Nowicka

Tabela *Samochody*

Nr_rejestracyjny	Marka	Model	Rok_produkcji	Id_Osoby
WN 34598	Fiat	Punto	2015	2
KR 923Y	Toyota	Rav4	2020	2
KRA 39U7	Ford	Focus	2023	4
SK 3987	Ford	Galaxy	2024	4
DW 9821	Opel	Corsa	2020	1
PO 8935	Opel	Corsa	2021	1
SBE 7389	Nissan	QASHQAI	2022	1

Przeanalizuj zapytanie:

SELECT Osoby.Id_Osoby, Osoby.Imie, Osoby.Nazwisko, Count(Samochody.Marka) **AS** Liczba
FROM Osoby
LEFT JOIN Samochody **ON** Samochody.Id_Osoby = Osoby.Id_Osoby
GROUP BY Osoby.Id_Osoby;

W wyniku powyższego zapytania

1.	dla każdego właściciela otrzymamy informację, ile ma różnych marek pojazdów.	**P**	**F**
2.	dla każdego właściciela otrzymamy informację, ile ma pojazdów.	**P**	**F**
3.	dla Eweliny Nowickiej w polu Liczba otrzymamy wynik 3.	**P**	**F**
4.	dla Beaty Kosińskiej w polu Liczba otrzymamy wynik 0.	**P**	**F**

BRUDNOPIS *(nie podlega ocenie)*

INFORMATYKA

Poziom rozszerzony

Formuła 2015

INFORMATYKA

Poziom rozszerzony

Formuła 2015

INFORMATYKA

Poziom rozszerzony

Formuła 2015