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**-2606

DATA: **10 czerwca 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 10 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. Szyfrowanie

Algorytm Szyfr szyfruje jawny tekst o długości *n* znaków, który jest zapisany w tablicy *T*. Kluczem szyfrowania jest dodatnia liczba całkowita *k*. Szyfrogram jest zapisywany w tablicy *S*.

Specyfikacja:

Dane:

n – długość tekstu, dodatnia liczba całkowita
T[1..*n*] – tekst jawny, *T*[*i*] – *i*-ty znak w tekście
k – klucz, dodatnia liczba całkowita

Wynik:

S[1..*n*] – szyfrogram, *S*[*i*] – *i*-ty znak w szyfrogramie

Szyfr:

```
a ← n div k
c ← n – a
dla i = 1, 2, …, n wykonuj
    jeżeli i mod k = 0
        b ← i div k
        S [ n – a + b ] ← T [ i ]
    w przeciwnym razie
        S [ c ] ← T [ i ]
        c ← c – 1
```

Uwaga:

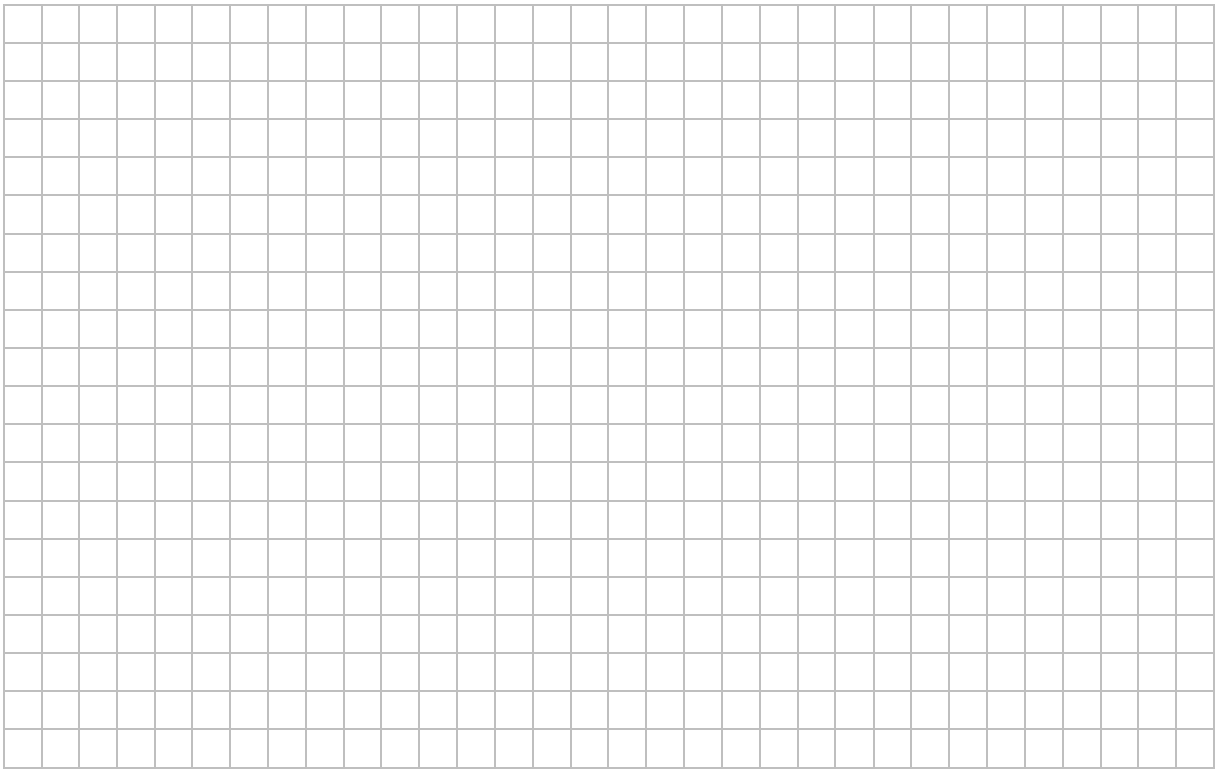
← oznacza przypisanie wartości
zapis *x* div *y* oznacza dzielenie całkowite liczby *x* przez liczbę *y*
zapis *x* mod *y* oznacza resztę z dzielenia liczby *x* przez liczbę *y*.

Zadanie 1.1. (0–3)

Uzupełnij tabelę – zapisz wynik działania algorytmu (szyfrogram) dla podanych wartości tekstu jawnego i klucza.

Liczba znaków n	Tekst jawny	Klucz *k*	Szyfrogram
10	korporacja	6	
10	tropikalny	5	
14	defragmentacja	3	

Miejsce na obliczenia (brudnopis)



Zadanie 1.2. (0–1)

Podaj wartość klucza *k*, dla której zawsze szyfrogram i tekst jawny są identyczne.

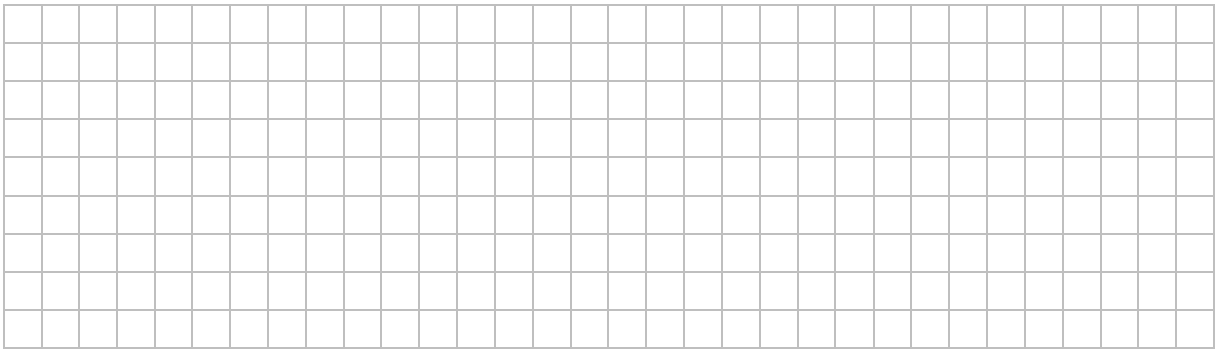
Odp. *k* = ……

Zadanie 1.3. (0–1)

Podaj wartość klucza *k*, dla której, dla każdego *n* zawsze szyfrogram jest tekstem jawnym zapisanym od końca.
Np. tak jest dla tekstu *serwer* i szyfrogramu *rewres*.
Uwaga: podana wartość może zależeć od *n*.

Odp. ……

Miejsce na obliczenia (brudnopis)



Zadanie 2. Algorytm

Dane są: liczba $n > 1$ oraz tablica $T[1..n]$ zawierająca liczby całkowite. Dla liczby całkowitej $1 \le k < n$ powiemy, że tablica T jest ***k-podobna*** wtedy i tylko wtedy, gdy $T[1..k] = T[n - k + 1..n]$. Ponadto przyjmujemy, że każda tablica jest 0-*podobna* ($k = 0$).

Przykład:
Tablica T = [1,2,1,1,2,1,1] jest 0-*podobna*,1-*podobna* i 4-*podobna*.
Tablica T = [1,2,2] jest 0-*podobna*.
Tablica T = [2,2,2] jest 0-*podobna*,1-*podobna* i 2-*podobna.*

Uwaga:
T = [2,2,2] nie jest 3-podobna, bo k musi być mniejsze niż liczba elementów tablicy (analogicznie T = [1,2,2] nie jest 3-podobna, a T = [1,2,1,1,2,1,1] nie jest 7-podobna).

Zadanie 2.1. (0–3)

Uzupełnij tabelę – dla podanych wartości tablicy T wypisz największe takie k, dla którego T jest k-podobna.

Długość tablicy - n	**Tablica *T***	**Największe takie *k*, dla którego *T* jest *k-podobna***
4	[1,2,3,4]	
8	[1,3,4,1,1,3,4,1]	
10	[3,5,3,5,3,5,3,5,3,5]	

Zadanie 2.2. (0–4)

W pseudokodzie lub w zadeklarowanym języku programowania zapisz algorytm, który oblicza największe takie k, dla którego tablica T jest *k-podobna*.

Uwaga: W zapisie algorytmu możesz korzystać tylko z operatorów arytmetycznych: dodawania, odejmowania, mnożenia, dzielenia, dzielenia całkowitego i reszty z dzielenia; operatorów logicznych, porównań liczb, odwoływania się **za pomocą indeksów** do **pojedynczych elementów** tablicy, 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, w szczególności nie wolno używać operacji pobierania fragmentu tablicy (listy).

Specyfikacja:

Dane:

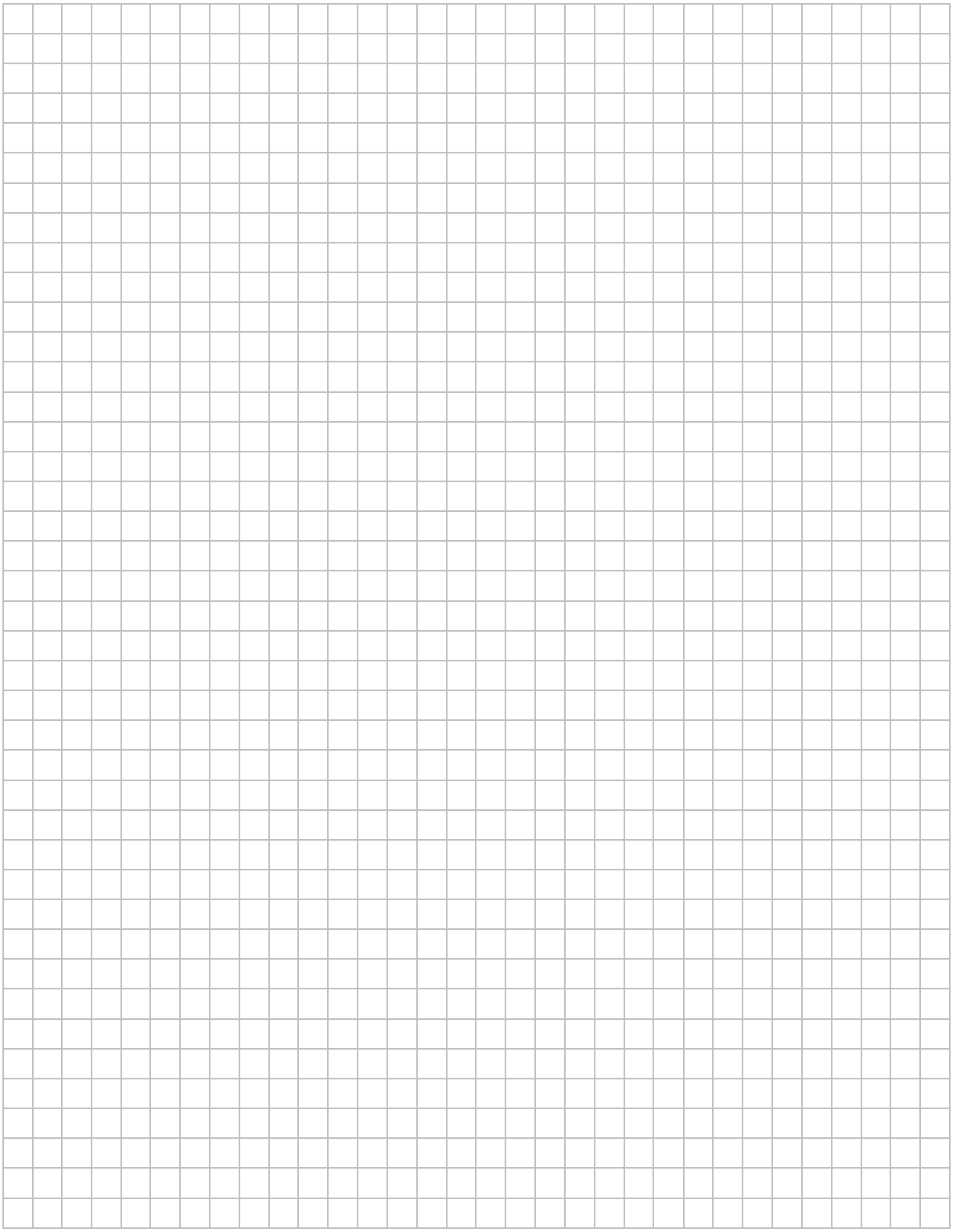
n – liczba całkowita większa od 1

$T[1..n]$ – tablica n liczb

Wynik:

k – największe takie k, dla którego tablica T jest *k-podobna*

Miejsce na zapis algorytmu



Zadanie 3. Test

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

Zadanie 3.1. (0–1)

Liczba 11101010_2 (zapisana w systemie binarnym) jest

1.	$= EA_{16}$	**P**	**F**
2.	$= 3222_4$	**P**	**F**
3.	$= 253_8$	**P**	**F**
4.	$> 255_{10}$	**P**	**F**

Informacja do zadań 3.2. i 3.3.

Dana jest tabela *wyniki*, która przedstawia punkty uzyskane w 12 rundach pewnego meczu drużyny *gospodarze* z 3 przeciwnikami.

runda	gospodarze	goscie	nr_przeciwnika
1	9	4	1
2	4	7	2
3	1	4	3
4	7	3	1
5	10	5	2
6	9	6	3
7	1	10	1
8	3	10	2
9	9	10	3
10	4	3	1
11	5	7	2
12	2	4	3

Zadanie 3.2. (0–1)

W wyniku zapytania:

```
SELECT nr_przeciwnika, SUM(gospodarze)
FROM wyniki
GROUP BY nr_przeciwnika;
```

otrzymamy odpowiedź:

1.	1 21 2 22 3 21	**P**	**F**
2.	1 20 2 29 3 24	**P**	**F**
3.	21 22 21	**P**	**F**
4.	20 29 24	**P**	**F**

Zadanie 3.3. (0–1)

W wyniku zapytania:

```
SELECT runda
FROM wyniki
WHERE
(gospodarze > 8 OR goscie > 8) AND (gospodarze – 7  >  goscie);
```

otrzymamy odpowiedź:

1.	1 2 3	**P**	**F**
2.	1 5 6 7 8 9	**P**	**F**
3.	pusty wynik	**P**	**F**
4.	1 5 6	**P**	**F**

BRUDNOPIS *(nie podlega ocenie)*

INFORMATYKA

Poziom rozszerzony

Formuła 2015

INFORMATYKA

Poziom rozszerzony

Formuła 2015

INFORMATYKA

Poziom rozszerzony

Formuła 2015