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**-2506

DATA: **10 czerwca 2025 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 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. Rozkład

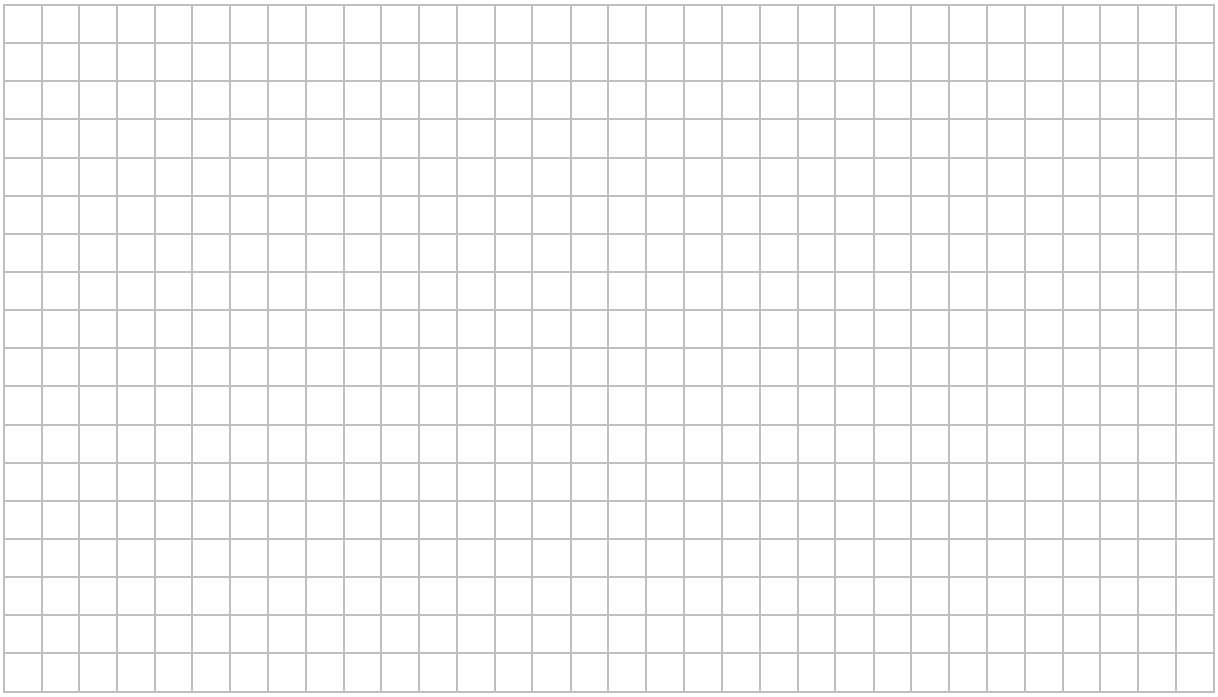
Każdą liczbę całkowitą większą od 1 można rozłożyć na czynniki pierwsze.

Zadanie 1.1. (0–2)

Uzupełnij tabelę – zapisz rozkład na czynniki pierwsze podanych liczb *n* oraz zapisz TAK, jeśli *n* jest iloczynem dokładnie dwóch różnych liczb pierwszych, albo NIE – w innym przypadku.

Liczba *n*	**Rozkład na czynniki pierwsze**	**Czy liczba *n* jest iloczynem dokładnie dwóch różnych liczb pierwszych?**
6	$2 \cdot 3$	TAK
9	$3 \cdot 3$	NIE
12	$2 \cdot 2 \cdot 3$	NIE
13	13	NIE
14		
27		
33		

Miejsce na obliczenia (brudnopis)



Zadanie 1.2. (0–4)

Niech n będzie liczbą całkowitą nie mniejszą od 2. W tablicy $P[2..n]$ dla każdego $i = 2, 3, \ldots, n$ zapisano najmniejszy czynnik pierwszy w rozkładzie i na czynniki pierwsze.

Przykład:

Dla $n = 19$ zawartość tablicy P to:

i	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$P[\, i \,]$	2	3	2	5	2	7	2	3	2	11	2	13	2	3	2	17	2	19

Twoim zadaniem jest uzupełnienie luk w algorytmie Rozkład zapisanym poniżej.
Dla każdej liczby całkowitej $i = 2, 3, \ldots, n$ algorytm powinien obliczać, czy jest ona liczbą pierwszą, czy też iloczynem dokładnie dwóch różnych liczb pierwszych, czy inną liczbą (potęgą liczby pierwszej o wykładniku co najmniej 2 lub iloczynem co najmniej trzech – niekoniecznie różnych – liczb pierwszych).
Algorytm Rozkład musi być zgodny z następującą specyfikacją:

Specyfikacja:

Dane:

$P[2..n]$ – tablica liczb całkowitych taka, że dla każdego $i = 2, 3, \ldots, n$, $P[\, i \,]$ jest równe najmniejszemu czynnikowi pierwszemu w rozkładzie i na czynniki pierwsze

Wynik:

$P[2..n]$ – tablica liczb całkowitych taka, że

$$P[\, i \,] = \begin{cases} 1 & \text{gdy } i \text{ jest liczbą pierwszą} \\ 2 & \text{gdy } i \text{ jest iloczynem dokładnie dwóch różnych liczb pierwszych} \\ 3 & \text{w pozostałych przypadkach} \end{cases}$$

Algorytm Rozkład:

dla $i = 2, 3, \ldots, n$

jeżeli $P[\, i \,] = i$

$P[\, i \,] =$ ……

inaczej

$j1 \leftarrow P[\, i \,]$

$j2 \leftarrow i \,/\, j1$

jeżeli $j1 \neq j2$ **oraz** ……

$P[\, i \,] =$ ……

inaczej

$P[\, i \,] =$ ……

Zadanie 2. Funkcja rekurencyjna

Zastosowane w poniższym algorytmie funkcje: ***skróć***, *dopisz* i *ostatnia*, przyjmują jako argument nieujemną liczbę całkowitą *x*.
Wynikiem funkcji ***skróć***(*x*) jest liczba powstała z *x* przez usunięcie najmniej znaczącej cyfry w jej zapisie dziesiętnym. Jeśli *x* jest liczbą jednocyfrową, wtedy ***skróć***(*x*) = 0. Przykładowo: ***skróć***(249) = 24, ***skróć***(87) = 8, ***skróć***(5) = 0.
Wynikiem funkcji *dopisz*(*x*) jest liczba, której zapis dziesiętny powstaje z zapisu dziesiętnego liczby *x* przez dopisanie jako najmniej znaczącej cyfry 0. Dla liczby 0 przyjmujemy *dopisz*(0) = 0. Przykładowo: *dopisz*(29) = 290.
Wynikiem funkcji *ostatnia*(*x*) jest liczba – wartość najmniej znaczącej cyfry zapisu dziesiętnego liczby *x*. Przykładowo: *ostatnia*(307) = 7.

Dana jest funkcja *f*(*a, b*), która przyjmuje jako argumenty dwie nieujemne liczby całkowite, a jej wynikiem jest nieujemna liczba całkowita:

```
f(a, b):
    jeżeli b = 0
        wynik 0
        zakończ
    k ← ostatnia(b)
    w ← f(a, skróć(b))
    w ← dopisz(w)
    dopóki k > 0 powtarzaj
        w ← w + a
        k ← k – 1
    wynik w
```

Zadanie 2.1. (0–3)

Uzupełnij tabelę – wpisz w ostatniej kolumnie wynik funkcji *f*(*a*, *b*) dla podanych wartości argumentów *a* i *b*.

a	***b***	***f*(*a, b*)**
42	2	
4	125	
103	104	

Miejsce na obliczenia (brudnopis)

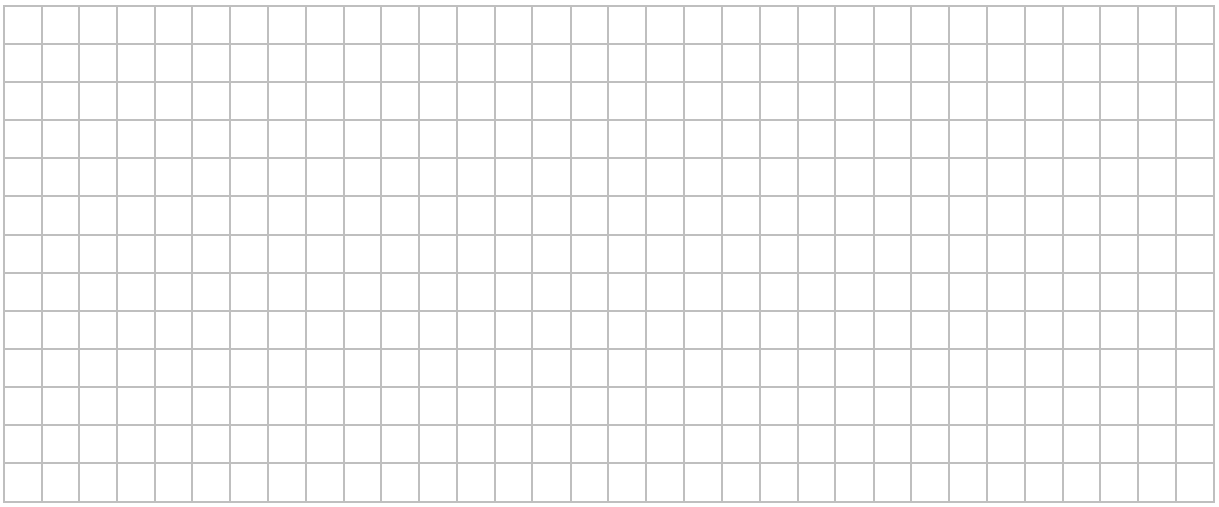


Zadanie 2.2. (0–1)

Ile razy łącznie zostanie wywołana funkcja ***f***, jeśli pierwszym wywołaniem będzie *f*(987654321, 123456789)?

Odpowiedź: .....

Miejsce na obliczenia (brudnopis)



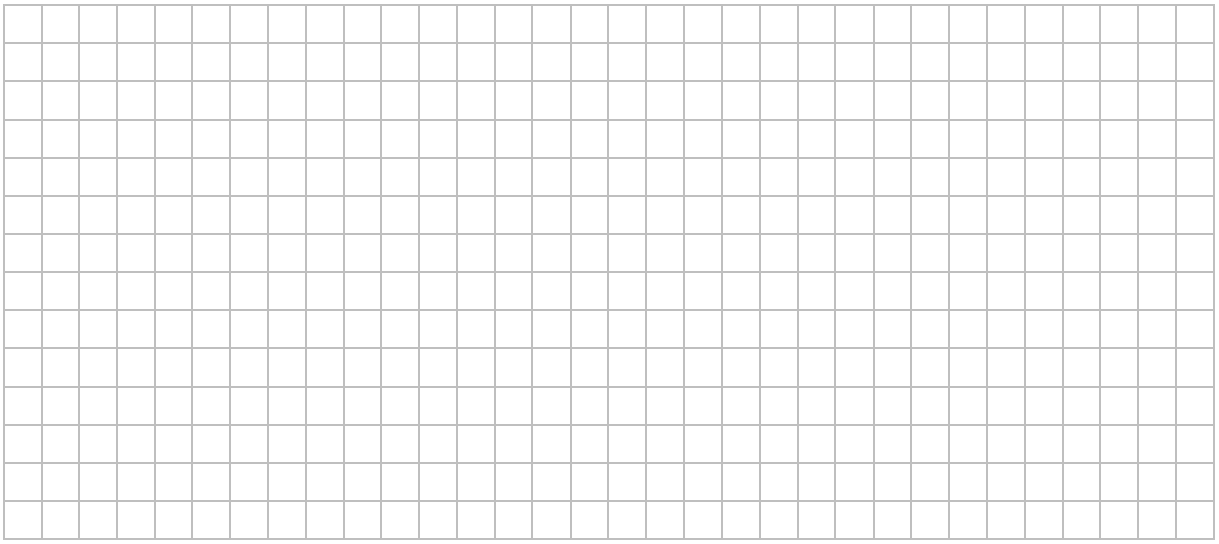
Zadanie 2.3. (0–2)

Ile razy łącznie zostanie wykonana instrukcja ***w*** ← ***w + a***, jeśli pierwszym wywołaniem będzie:

a) *f*(2024, 1000) odpowiedź: ……..

b) *f*(2024, 1234) odpowiedź: ……..

Miejsce na obliczenia (brudnopis)



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)

Dla liczb 11010101_2, 1222_4, 333_8, $D6_{16}$ zapisanych w systemach pozycyjnych o podstawach 2, 4, 8 i 16:

1.	$11010101_2 > 1222_4$	**P**	**F**
2.	$1222_4 > 333_8$	**P**	**F**
3.	$333_8 > D6_{16}$	**P**	**F**
4.	$D6_{16} > 11010101_2$	**P**	**F**

Zadanie 3.2. (0–1)

1.	W pliku w formacie GIF można zapisać obraz z 16 milionami kolorów bez utraty informacji o nich.	**P**	**F**
2.	W formacie JPG są wykorzystywane metody kompresji danych.	**P**	**F**
3.	Format JPG obsługuje przezroczyste tła obrazów.	**P**	**F**
4.	Pliki w formacie JPG mogą mieć rozszerzenie .jpg lub .jpeg.	**P**	**F**

Zadanie 3.3. (0–1)

W bazie danych istnieją dwie tabele: *kontrahenci* (id_kontrahenta, nazwa) oraz *faktury* (nr_faktury, id_kontrahenta, kwota, data). Między tabelami zachodzi relacja „jeden do wielu”. Baza zawiera następujące dane:

FAKTURY

nr_faktury	id_kontrahenta	kwota	data
1/07/2022/F	004	426	01.07.2022
2/07/2022/F	002	142	03.07.2022
3/07/2022/F	001	689	10.07.2022
4/07/2022/F	001	603	12.07.2022
5/07/2022/F	001	599	13.07.2022
6/07/2022/F	004	161	19.07.2022
7/07/2022/F	003	769	25.07.2022
8/07/2022/F	003	748	28.07.2022

KONTRAHENCI

id_kontrahenta	Nazwa
001	Zima
002	Wiosna
003	Lato
004	Jesien

1.	Wynikiem zapytania SQL
	
	`SELECT Sum(Faktury.kwota)`
	`FROM Faktury;`
	
	jest suma kwot.
		P	**F**

2.	Wynikiem zapytania SQL
	
	`SELECT Kontrahenci.Nazwa, Count(Faktury.nr_faktury)`
	`FROM Kontrahenci JOIN Faktury ON`
	`Kontrahenci.id_kontrahenta = Faktury.id_kontrahenta`
	`GROUP BY Kontrahenci.Nazwa`
	`HAVING ((Count(Faktury.nr_faktury))>=2);`
	
	jest
	`Jesien 2`
	`Lato 2`
		P	**F**

3.	Wynikiem zapytania SQL
	
	`SELECT Faktury.nr_faktury`
	`FROM Faktury`
	`WHERE Faktury.kwota Between 100 And 300;`
	
	jest lista **dwóch** numerów faktur.
		P	**F**

4.	Wynikiem zapytania SQL
	
	`SELECT Kontrahenci.Nazwa, Faktury.kwota`
	`FROM Kontrahenci JOIN Faktury ON`
	`Kontrahenci.id_kontrahenta = Faktury.id_kontrahenta`
	`ORDER BY Faktury.kwota DESC;`
	
	jest lista kontrahentów i kwot posortowana rosnąco według kwoty.
		P	**F**

BRUDNOPIS *(nie podlega ocenie)*

INFORMATYKA

Poziom rozszerzony

Formuła 2015

INFORMATYKA

Poziom rozszerzony

Formuła 2015

INFORMATYKA

Poziom rozszerzony

Formuła 2015