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**-2406

DATA: **14 czerwca 2024 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. Liczby *nieliczne*

Dodatnią liczbę całkowitą *L* nazywamy liczbą *nieliczną*, jeżeli w jej zapisie binarnym cyfry 1 nie sąsiadują ze sobą.

Zadanie 1.1. (0–2)

Uzupełnij tabelę – w drugiej kolumnie podaj zapis binarny liczby podanej w zapisie o podstawie 8 oraz liczby podanej w zapisie szesnastkowym, w trzeciej kolumnie wpisz PRAWDA gdy podana liczba jest *nieliczna*, albo FAŁSZ – w przeciwnym przypadku.

Liczba *L*	**Zapis binarny liczby *L***	**Czy liczba *L* jest *nieliczna*?**
259_{10}	100000011	FAŁSZ
20012_{8}		
3574_{16}		

Zadanie 1.2. (0–3)

W wybranej przez siebie notacji (w pseudokodzie lub języku programowania) **napisz funkcję** *czy_nieliczna(k)*, która sprawdzi, czy dana dodatnia liczba *k* jest liczbą *nieliczną*.

Uwaga: Twój algorytm może używać **wyłącznie zmiennych przechowujących liczby całkowite** oraz może **operować wyłącznie na liczbach całkowitych**. W zapisie możesz wykorzystać tylko operacje arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie, dzielenie całkowite, reszta z dzielenia), porównywanie liczb, instrukcje sterujące, przypisania do zmiennych lub samodzielnie napisane funkcje, wykorzystujące powyższe operacje. **Zabronione** jest używanie funkcji wbudowanych oraz operatorów innych niż wymienione, dostępnych w językach programowania. W szczególności nie wolno używać żadnych funkcji zamiany zapisu liczb między systemami pozycyjnymi.

Specyfikacja:

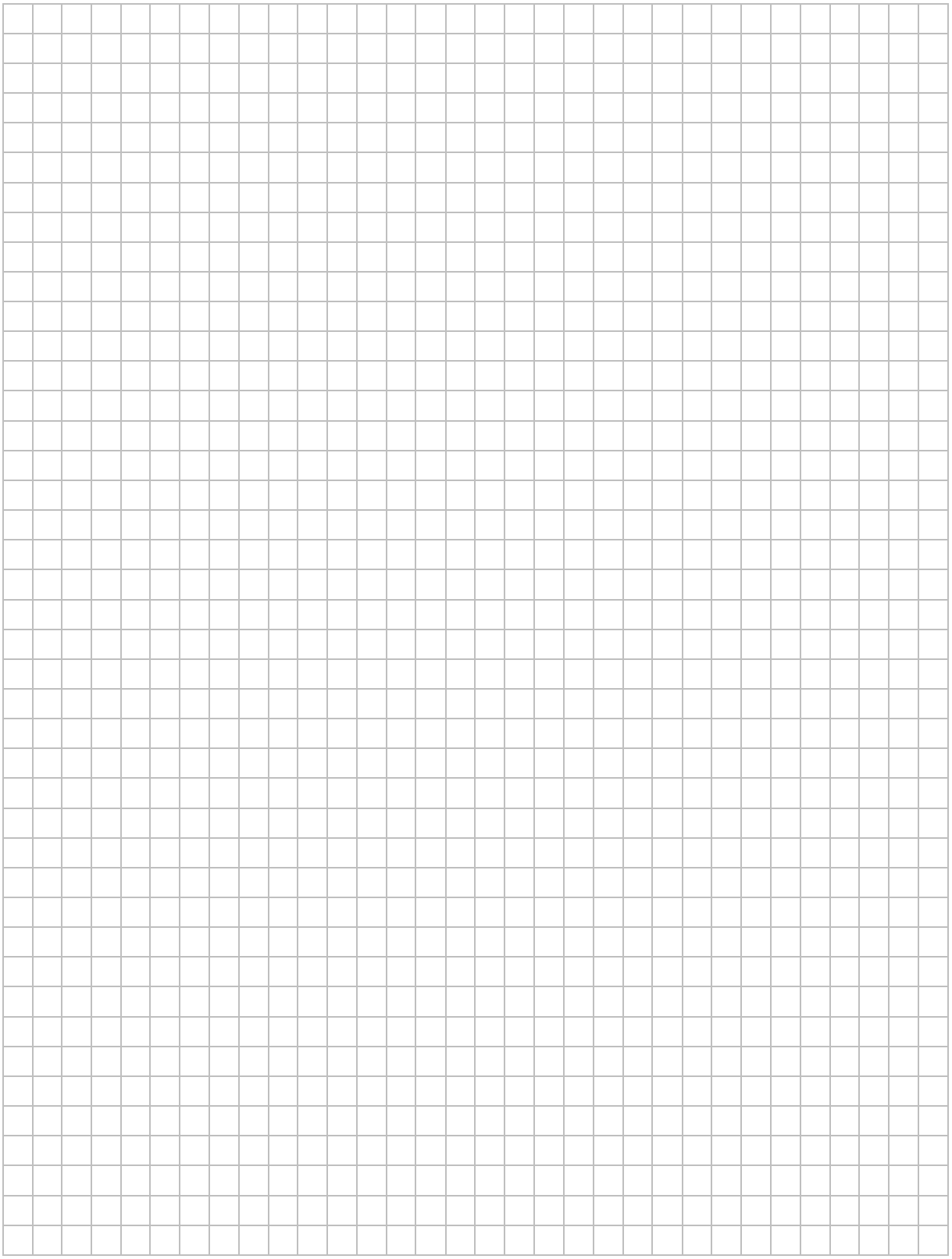
Dane:

k – dodatnia liczba całkowita

Wynik:

PRAWDA – jeżeli liczba *k,* jest liczbą *nieliczną*, albo FAŁSZ – w przeciwnym przypadku

Miejsce na zapis algorytmu



Zadanie 2. Funkcja

Dane są dodatnia liczba całkowita n oraz n-elementowy ciąg liczb, zapisany w tablicy $A[1..n]$. Przeanalizuj poniższy algorytm.

Algorytm 1:

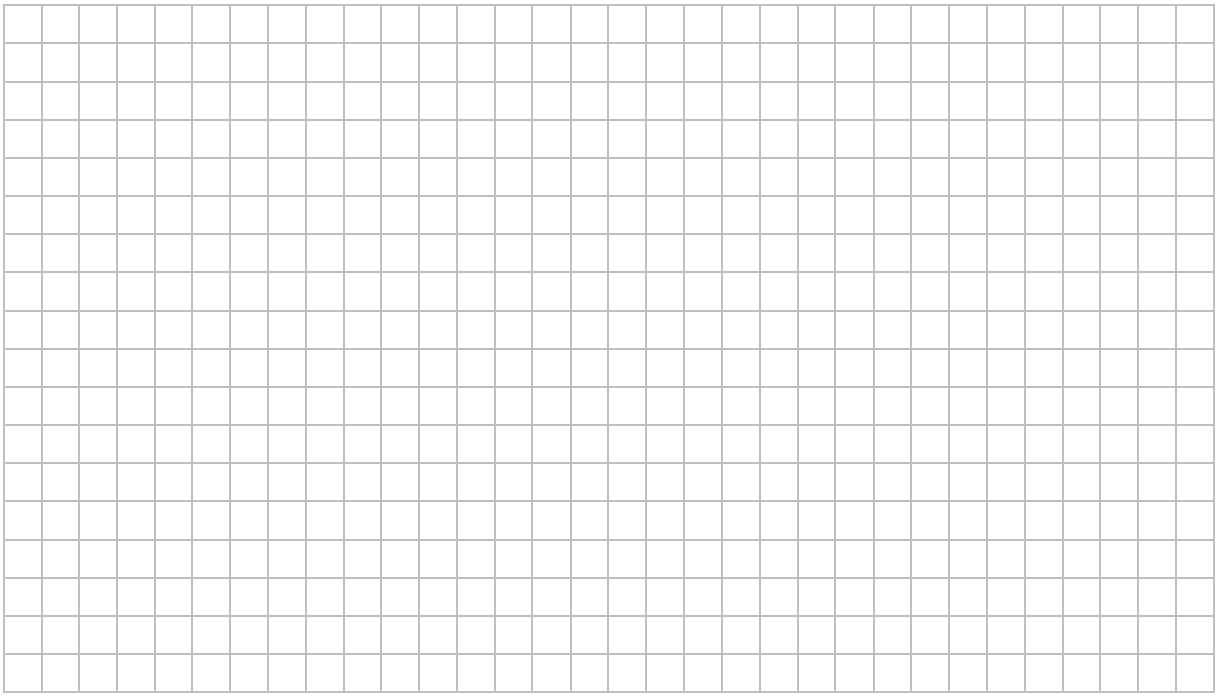
```
p ← 1
m ← 1
dla i = 1, 2, …, n wykonuj
        licz ← 0
        dla j = 1, 2, …, n wykonuj
                jeżeli A[ i ] = A[ j ] wykonuj
                        licz ← licz +1
        jeżeli licz > m wykonuj
                p ← i
                m ← licz
```

Zadanie 2.1. (0–3)

Jakie wartości przyjmą: p, $A[p]$ i m po wykonaniu powyższego algorytmu dla ciągu z tablicy A = [7, 9, 15, 0, 6, 7, 6, 10, 15, 6, 0] ?

p = _____ $A[p]$ = _____ m = _____

Miejsce na obliczenia (brudnopis)



Zadanie 2.2. (0–2)

Podaj, ile razy w algorytmie 1. dla podanych wartości *n* wykonywana jest operacja sprawdzenia, czy $A[i] = A[j]$.

n	Ile razy sprawdzany jest warunek $A[i] = A[j]$
3	
n	

Miejsce na obliczenia (brudnopis)

Zadanie 2.3. (0–3)

Uzupełnij luki w algorytmie 2. tak, by stanowił on odpowiednik algorytmu 1., w którym operacja porównania $A[i] = A[j]$ jest wykonywana nie więcej niż $\frac{n(n-1)}{2}$ razy.

Algorytm 2:

$p \leftarrow 1$

$m \leftarrow 1$

dla *i* = …… **wykonuj**

  licz ← ……

  dla *j* = …… **wykonuj**

    jeżeli $A[i] = A[j]$ **wykonuj**

      licz ← *licz* +1

  jeżeli *licz* > *m* **wykonuj**

    $p \leftarrow i$

    $m \leftarrow licz$

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)

W tabeli *miasta* zamieszczono informacje o liczbie ludności miast Polski. Zawartość tabeli *miasta*:

identyfikator	**miasto**	**wojewodztwo**	**ludnosc**
767	Szamotuły	wielkopolskie	18853
768	Szczawnica	małopolskie	5711
769	Szczawno-Zdrój	dolnośląskie	5569
770	Szczebrzeszyn	lubelskie	4964
771	Szczecin	zachodniopomorskie	401907

Dla podanych danych w wyniku zapytania

SELECT ludnosc
FROM miasta
WHERE identyfikator > 770 OR miasto = "Szamotuły"

otrzymamy

1.	dwie liczby	**P**	**F**
2.	pusty wynik	**P**	**F**
3.	401907	**P**	**F**
4.	420760	**P**	**F**

Zadanie 3.2. (0–1)

Wynik dodawania liczb binarnych 1011101_2 oraz 111_2 jest:

1.	równy 1100110_2	**P**	**F**
2.	mniejszy niż 1111111_2	**P**	**F**
3.	większy niż 1110000_2	**P**	**F**
4.	równy 1100100_2	**P**	**F**

BRUDNOPIS *(nie podlega ocenie)*

INFORMATYKA

Poziom rozszerzony

Formuła 2015

INFORMATYKA

Poziom rozszerzony

Formuła 2015

INFORMATYKA

Poziom rozszerzony

Formuła 2015