

IV Liceum Ogólnokształcące w Białymstoku

PRÓBNY EGZAMIN MATURALNY Z INFORMATYKI 2022

Arkusz I

Czas pracy: **60 minut**

Liczba punktów do uzyskania: **15**

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 8 stron (zadania 1 – 3). Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
2. Odpowiedzi zapisz w miejscu na to przeznaczonym przy każdym zadaniu.
3. Pisz czytelnie. Używaj długopisu/pióra tylko z czarnym tuszem/atramentem.
4. Nie używaj korektora, a błędne zapisy wyraźnie przekreśl.
5. Pamiętaj, że zapisy w brudnopisie nie podlegają ocenie.
6. Wpisz poniżej zadeklarowany przez Ciebie na egzamin system operacyjny, program użytkowy oraz środowisko programistyczne.
7. Jeżeli rozwiązaniem zadania lub jego części jest algorytm, to zapisz go w wybranej przez siebie notacji: listy kroków, pseudokodu lub języka programowania, który wybrałeś/eś na egzamin.

WYBRANE:

.....

(system operacyjny)

.....

(program użytkowy)

.....

(środowisko programistyczne)

PESEL:

--	--	--	--	--	--	--	--	--	--	--

Klasa:

--	--	--

Zadanie 1. Test (0-3)

Oceń prawdziwość podanych zdań. Zaznacz **P**, jeśli zdanie jest prawdziwe, albo **F**, jeśli zdanie jest fałszywe. W każdym zadaniu uzyskasz punkt, jeśli poprawnie odpowiesz na wszystkie jego części.

Zadanie 1.1. (0–1)

Przeanalizuj przedstawiony poniżej algorytm.

Dane:

n – liczba całkowita dodatnia

Algorytm:

Wczytaj liczbę n

Dopóki $(n \bmod 2) = 0$ powtarzaj:

$n \leftarrow n \text{ DIV } 2$ (*)

Jeśli $n > 1$ to wypisz NIE, w przeciwnym razie wypisz TAK

Uwaga. $n \bmod m$ oznacza resztę z dzielenia całkowitego liczby n przez m

$n \text{ DIV } 2$ oznacza dzielenie całkowite n przez 2

\leftarrow oznacza instrukcję przypisania

1.	Instrukcja oznaczona (*) dla liczby 32 wykona się 5 razy	P	F
2.	Dla każdego $n \leq 200$ instrukcja (*) wykona się co najwyżej 7 razy	P	F
3.	Algorytm wypisuje TAK dla każdej parzystej dodatniej liczby n	P	F
4.	Dla n będącą naturalną potęgą liczby 2 algorytm wypisze TAK	P	F

Zadanie 1.2. (0–1)

Dana jest następująca funkcja:

funkcja $f(n)$:

jeżeli $n > 0$

wypisz n

$F(n - 2)$

wypisz n

1.	W wyniku wywołania $f(5)$ otrzymamy ciąg 5 5 5 5 5 5	P	F
2.	W wyniku wywołania $f(6)$ otrzymamy ciąg 6 4 2 2 4 6	P	F
3.	W wyniku wywołania $f(7)$ otrzymamy ciąg 7 5 3 1 1 3 5 7	P	F
4.	W wyniku wywołania $f(8)$ otrzymamy ciąg 8 6 4 2 0 0 2 4 6 8	P	F

Zadanie 1.3. (0–1)

Po wykonaniu podanego zapytania SQL do pewnej bazy danych wyniki będą zawsze uporządkowane niemalejąco według pola *nazwa*.

1.	SELECT nazwa, wartość FROM dane ORDER BY nazwa DESC	P	F
2.	SELECT nazwa, wartość FROM dane ORDER BY nazwa	P	F
3.	SELECT nazwa, sum(wartość) FROM dane GROUP BY nazwa	P	F
4.	SELECT nazwa, sum(wartość) FROM nazwa ORDER BY dane	P	F

Zadanie 2. Symetria liczby (0-7)

Liczba symetryczna – liczba naturalna, która nie zmienia się po zapisaniu jej cyfr w odwrotnej kolejności.

Przykłady liczb symetrycznych:

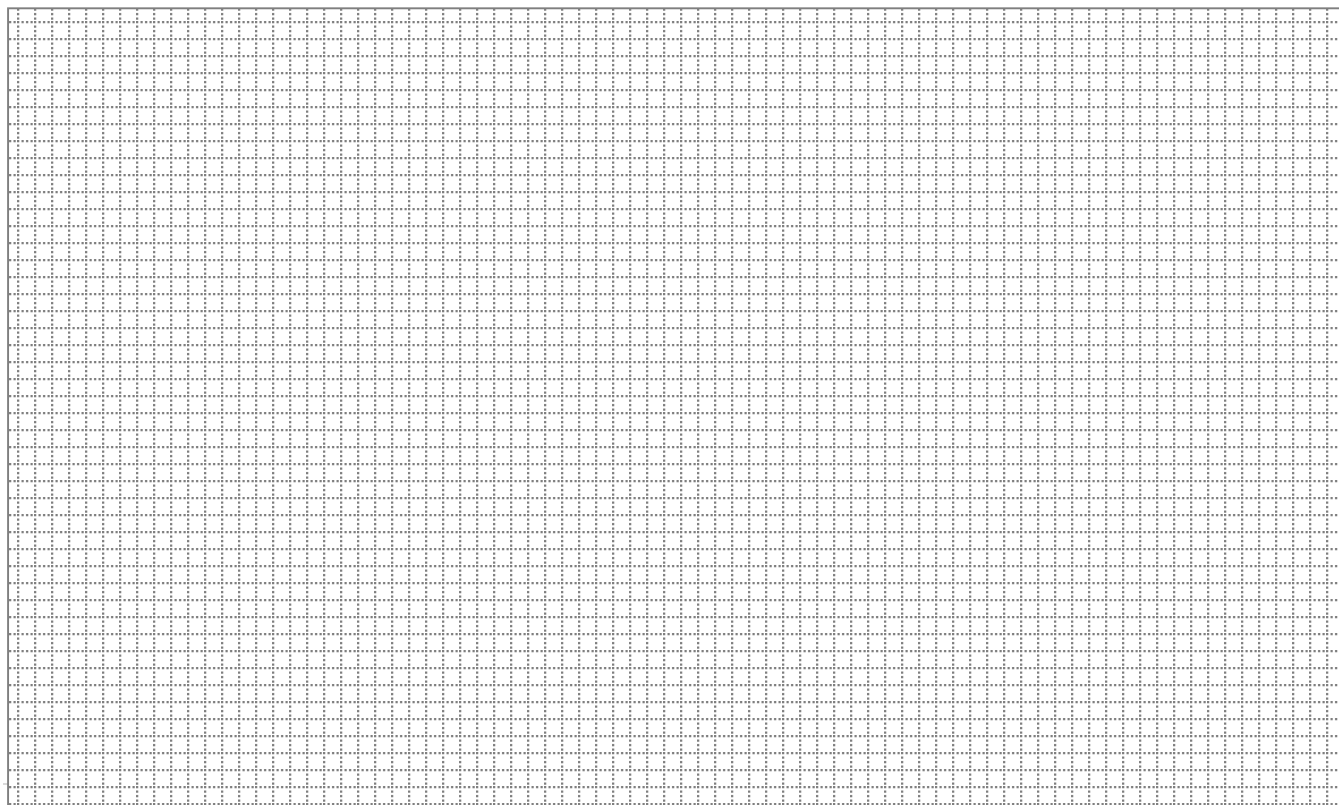
0, 9, 11, 121, 4884, 10001

Uwaga. Przyjmujemy, że liczby podajemy w najkrótszej możliwej postaci, tzn. bez ewentualnych zer wiodących.

Zadanie 2.1 (0–1)

Sprawdź, które liczby sześciocyfrowe symetryczne w systemie dwójkowym są liczbami symetrycznymi w systemie dziesiętnym.

Miejsce na obliczenia:



Zadanie 2.2 (0–6)

Napisz algorytm (w postaci listy kroków, w pseudokodzie lub wybranym języku programowania) który znajduje sumę wszystkich liczb mniejszych niż n , które są symetrycznie jednocześnie w systemie dziesiętnym i dwójkowym. Uwaga: Przy ocenie będzie brana pod uwagę złożoność obliczeniowa algorytmu.

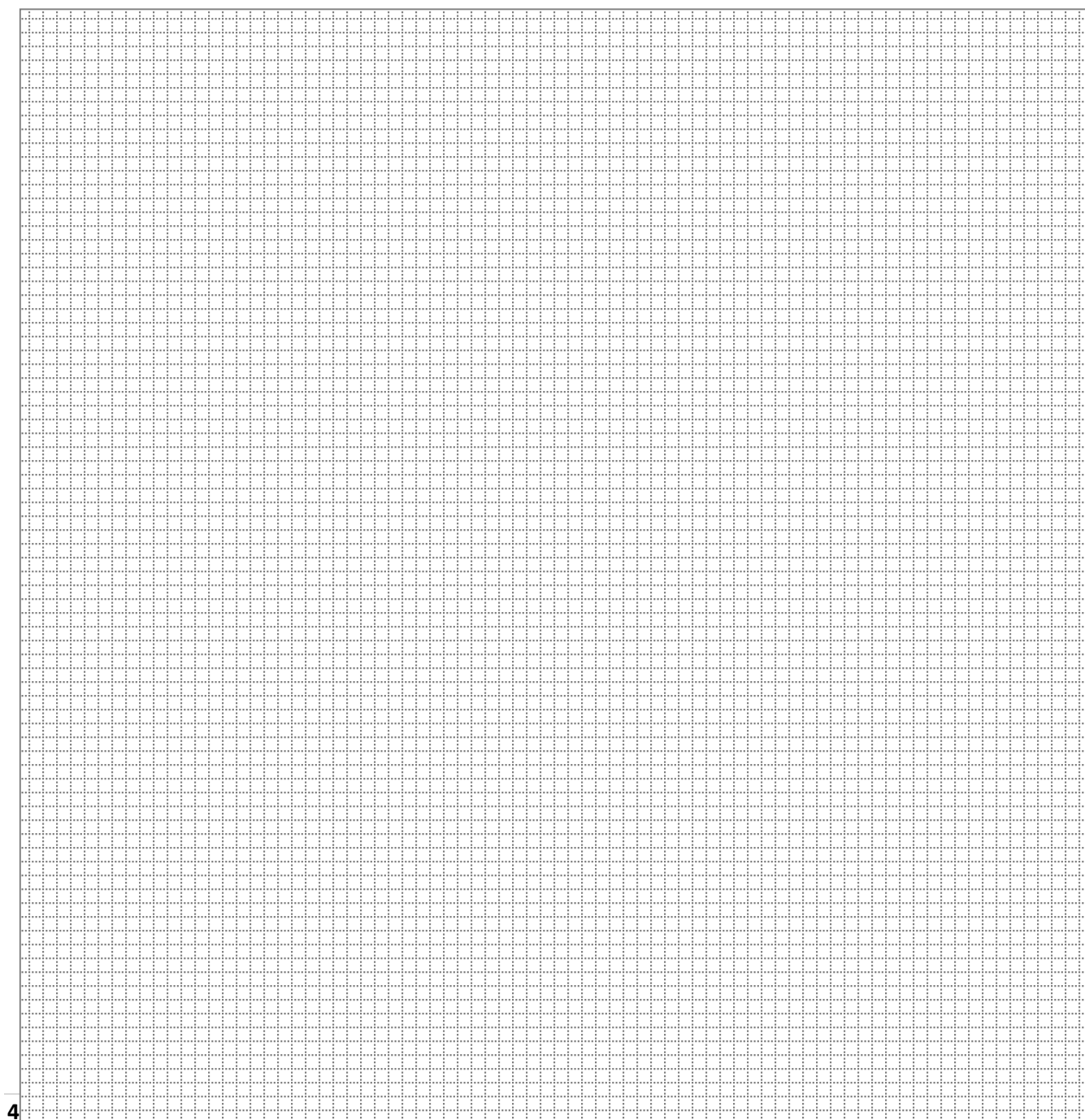
Specyfikacja:

Dane:

n – liczba całkowita dodatnia mniejsza od miliona

Wynik:

suma – liczba całkowita będąca sumą liczb symetrycznych jednocześnie w systemach dziesiętnym i dwójkowym. Wynik podaj w systemie dziesiętnym.



Zadanie 3. Szyfr podstawieniowy (0-5)

Szyfr podstawieniowy – szyfr, w którym każdy znak tekstu jawnego jest zastępowany przez inny znak lub znaki szyfrogramu.

Liczbą Fibonacciego nazywamy każdy wyraz ciągu (a_n) określonego rekurencyjnie jako:

$$\begin{cases} a_1 = 1 \\ a_2 = 1 \\ a_n = a_{n-2} + a_{n-1} \end{cases} \quad \text{dla } n \in N^+.$$

Początkowe elementy ciągu liczb Fibonacciego wynoszą: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Bartek stworzył funkcję szyfrującą, która wykorzystuje liczby Fibonacciego w ten sposób, że zaszyfrowany tekst powstaje poprzez przesunięcie k-tej litery tekstu jawnego o wartość: **$w \bmod 26$** , gdzie w jest wartością k-tej liczby z ciągu Fibonacciego.

Specyfikacja:

Dane:

fib(k) – funkcja rekurencyjna obliczająca wartość k-tej liczby z ciągu Fibonacciego w arytmetyce modularnej *mod 26*

s[1..d] – tekst jawny składający się z dużych liter alfabetu o długości d znaków

znak(k) - funkcja zamieniająca liczbę całkowitą k na znak o kodzie k, np. znak(65) → A

kod(zn) – funkcja zamieniająca znak na jego kod dziesiętny ASCII np. kod('C') → 67

Wynik:

szyfr[1..d] - tekst po zaszyfrowaniu składający się z dużych liter alfabetu o długości d znaków.

Fib(k)

Jeśli $k < 3$

Podaj wynik 1

W przeciwnym przypadku

Podaj wynik $(\text{fib}(k-1) + \text{fib}(k-2)) \bmod 26$

Od $i=1$ do d

Jeżeli $s[i] \geq 'A'$ i $s[i] \leq 'Z'$

$\text{szyfr}[i] = \text{znak}((65 + (\text{kod}(s[i]) - 65 + \text{fib}(i)) \bmod 26))$

Wypisz szyfr

A	B	C	D	E	F	G	H	I	J	K	L	M
65	66	67	68	69	70	71	72	73	74	75	76	77

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---

Napisz algorytm (w postaci listy kroków, pseudokodu lub w wybranym języku programowania), który będzie szyfrował wiadomość zgodnie z wymaganiami Bartka

Specyfikacja:

Dane:

F1, F2 – dwie liczby naturalne określające początkowe wartości pseudociągu Fibinacciego

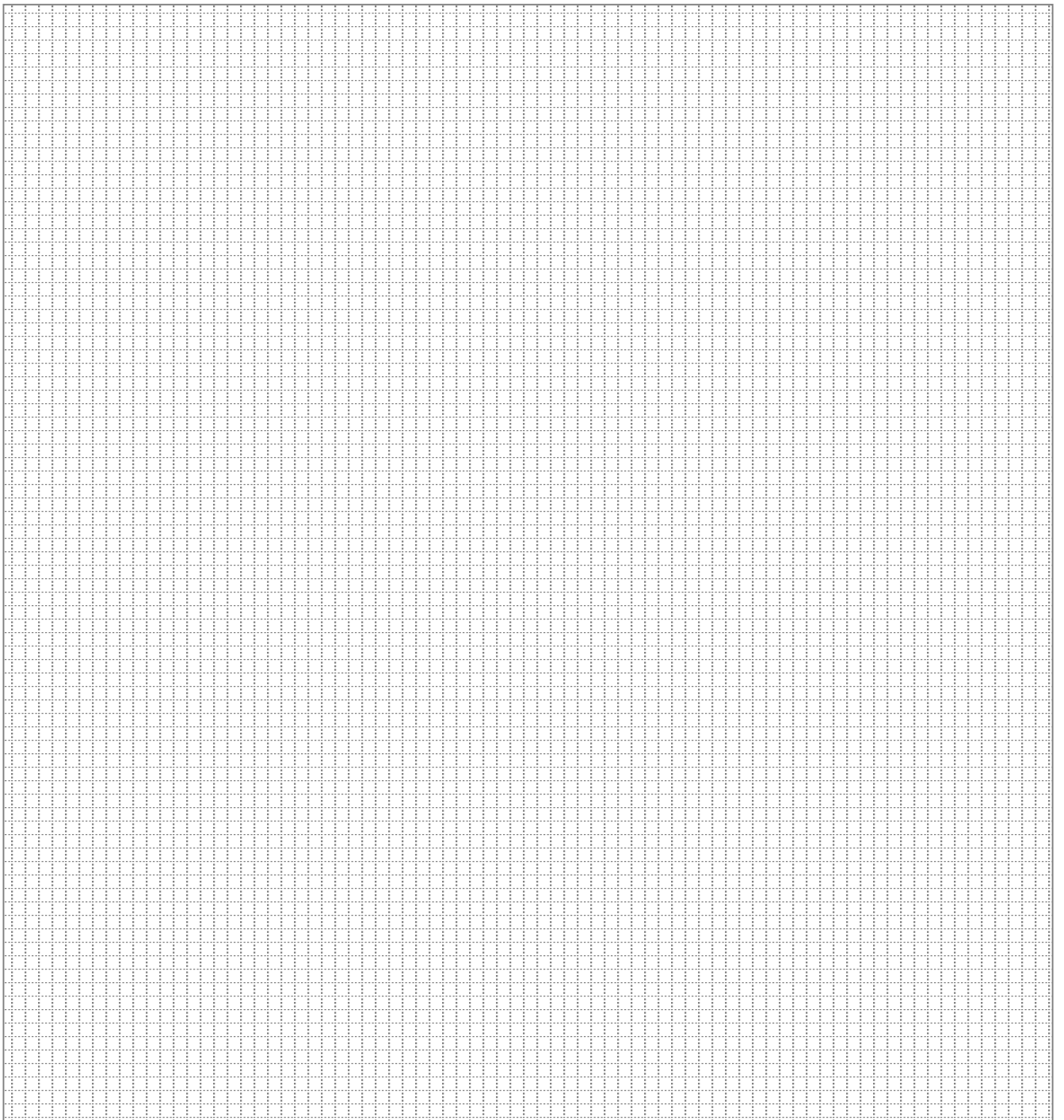
F – kolejny wyraz pseudociągu Fibonacciego liczony w arytmetyce modularnej *mod* 26

s[1..d] – tekst jawny składający się z dużych liter alfabetu o długości d znaków

Wynik:

szyfr[1..d] - tekst po zaszyfrowaniu składający się z dużych liter alfabetu o długości d znaków.

Miejsce na algorytm:



Brudnopis (nie podlega ocenie)