

---

**PRÓBNY EGZAMIN MATURALNY Z INFORMATYKI STYCZEŃ 2018**

---

Arkusz I

---

Czas pracy: **60 minut**

Liczba punktów do uzyskania: **15**

---

**Instrukcja dla zdającego**

1. Sprawdź, czy arkusz egzaminacyjny zawiera 10 stron (zadania 1 – 3). Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
  2. Rozwiązania i odpowiedzi zamieść w miejscu na to przeznaczonym.
  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 zadeklarowane (wybrane) przez Ciebie na egzamin środowisko komputerowe, kompilator języka programowania oraz program użytkowy.
  7. Jeżeli rozwiązaniem zadania lub jego części jest algorytm, to zapisz go w wybranej przez siebie notacji: listy kroków, pseudokodu lub języka programowania, który wybrałaś/eś na egzamin.
- 

**Dane uzupełnia uczeń:**

**WYBRANE:**

.....  
(środowisko)

.....  
(kompilator)

.....  
(program użytkowy)

**PESEL:**

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

**Klasa:**

--	--	--

## Zadanie 1. Test

Oceń, czy poniższe zdania są prawdziwe. 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)

Dana jest funkcja `fun` z jednym parametrem `n` będącym liczbą naturalną, którą można zapisać:

```
fun (n)
```

```
  if n = 0 then return 1
```

```
  else return 2*fun(n-1)
```

1.	<code>fun</code> jest funkcją rekurencyjną	<b>P</b>	<b>F</b>
2.	Dla $n = 2$ funkcja zwraca 4.	<b>P</b>	<b>F</b>
3.	Dla $n = 3$ funkcja zwraca 4.	<b>P</b>	<b>F</b>
4.	Funkcja <code>fun</code> oblicza wartość $2^n$ .	<b>P</b>	<b>F</b>

### Zadanie 1.2. (0–1)

W systemie szesnastkowym, dla dwóch liczb `C8` i `B0`

1.	suma tych liczb jest równa <code>DC</code> .	<b>P</b>	<b>F</b>
2.	iloczyn tych liczb jest równy <code>F00</code> .	<b>P</b>	<b>F</b>
3.	różnica <code>C8 – B0</code> jest równa <code>B4</code> .	<b>P</b>	<b>F</b>
4.	iloraz <code>C8/B0</code> jest równy <code>10</code> .	<b>P</b>	<b>F</b>

### Zadanie 1.3. (0–1)

1.	Liczba <code>150.22.33.1</code> to adres zapisany zgodnie z protokołem IPv4.	<b>P</b>	<b>F</b>
2.	Host o adresie <code>192.168.0.1</code> z maską <code>255.255.255.0</code> należy do podsieci <code>192.168.10.0</code>	<b>P</b>	<b>F</b>
3.	Adres IPv4 jest liczbą 64 bitową.	<b>P</b>	<b>F</b>
4.	Adres IPv4 jest liczbą 128 bitową.	<b>P</b>	<b>F</b>

#### Zadanie 1.4. (0–1)

W pliku `ilustarcja.html` zapisano następujący kod:

```
<a href="rysunki/krajobraz.jpg">widok</a>
```

Po wyświetleniu pliku `ilustracja.html` w przeglądarce internetowej, na otrzymanej stronie www:

1.	słowo <code>widok</code> jest hiperłączem do pliku <code>krajobraz.jpg</code> zapisanego w folderze <code>rysunki</code>	<b>P</b>	<b>F</b>
2.	jest wyświetlony rysunek z pliku <code>krajobraz.jpg</code>	<b>P</b>	<b>F</b>
3.	jako tło strony wyświetli się rysunek z pliku <code>krajobraz.jpg</code>	<b>P</b>	<b>F</b>
4.	po kliknięciu na słowo <code>widok</code> wyświetli się rysunek z pliku <code>krajobraz.jpg</code>	<b>P</b>	<b>F</b>

Wypełnia egzaminator	Nr zadania	1.1	1.2	1.3	1.4	Suma
	Maksymalna liczba punktów	1	1	1	1	4
	Uzyskana liczba punktów					

## Zadanie 2.

Palindromem nazywamy słowo, które składa się kolejno z tych samych znaków przy czytaniu go od lewej i od prawej strony. Przykładem palindromu jest słowo `kajak`.

### Zadanie 2.1 (0–2)

W wybranym języku programowania napisz funkcję realizującą algorytm dla podanej poniżej specyfikacji.

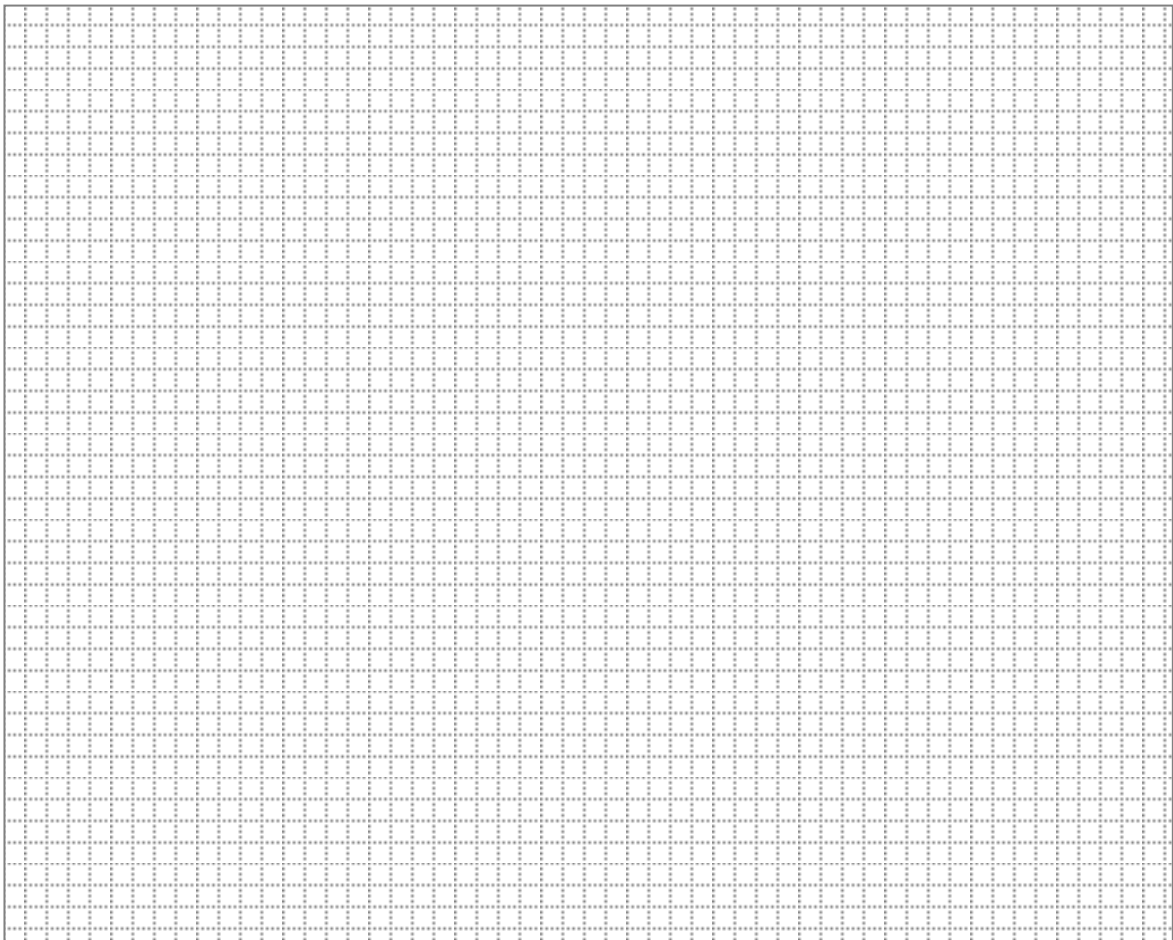
**Dane:**

liczba naturalna  $n > 0$  (długość słowa),  
słowo  $w$  o  $n$  znakach (słowo do sprawdzenia).

**Wynik:**

wartość logiczna `true`, jeśli słowo jest palindromem, wartość logiczna `false`, w przeciwnym wypadku.

*Miejsce na wpisanie funkcji:*



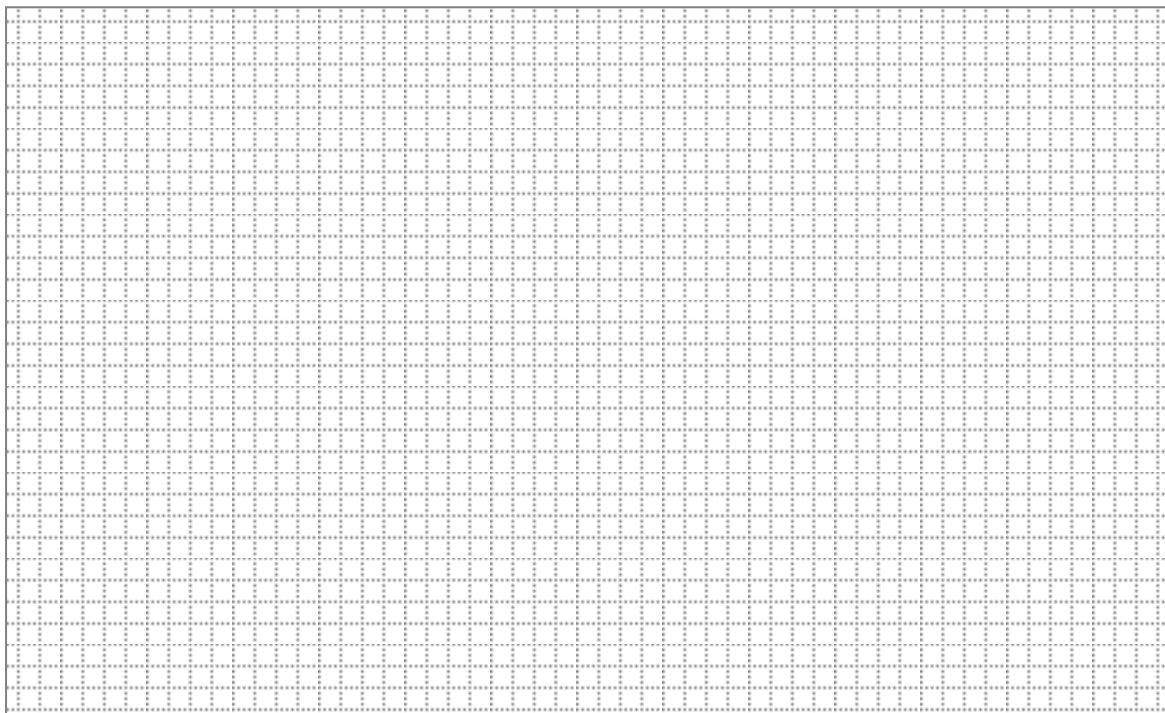
**Zadanie 2.2 (0–1)**

**Przesunięciem cyklicznym** słowa  $w$  w prawo o  $k$  znaków nazywamy przesunięcie, w którym  $k$  ostatnich znaków danego słowa przesuwanych jest kolejno na jego początek.

W tabeli poniżej wpisz najmniejsze  $k$  takie, że po przesunięciu cyklicznym danego słowa  $w$  w prawo o  $k$  będzie ono palindromem. Zapisz także słowo powstałe po tym przesunięciu. Jeżeli takie przesunięcie nie istnieje, wpisz wartość  $-1$ .

Słowo $w$	$K$	$w$ po przesunięciu cyklicznym w prawo o $k$ znaków
akkaj	3	kajak
dabccba		
oowocow		
kajak		
ninaanil		

**Miejsce na obliczenia:**



### Zadanie 2.3 (0–3)

W wybranym języku programowania napisz funkcję realizującą algorytm dla podanej poniżej specyfikacji.

**Dane:**

liczba naturalna  $n > 0$  (długość słowa),

słowo  $w$  o  $n$  znakach (słowo do sprawdzenia),

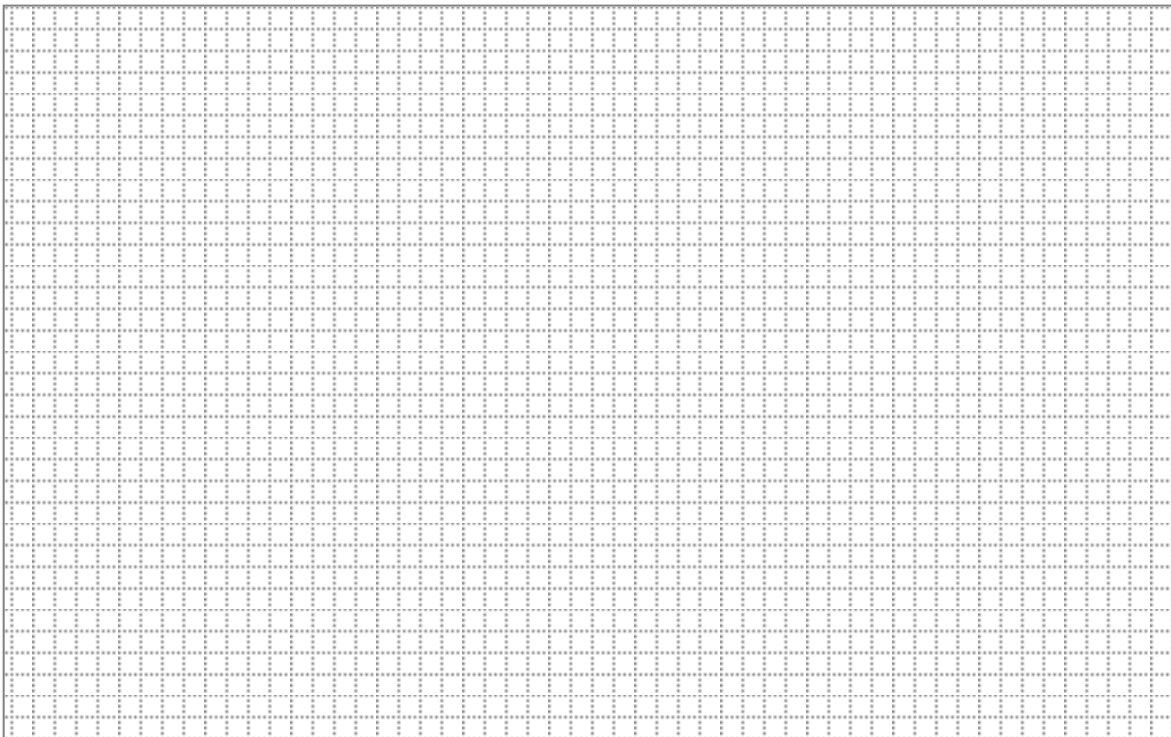
liczba naturalna  $k$  taka, że  $0 \leq k < n$  (liczb znaków przesunięcia cyklicznego w prawo).

**Wyniki:**

wartość logiczna `true`, jeśli słowo  $w$  przesunięte o  $k$  liter w prawo jest

palindromem, wartość logiczna `false` w przeciwnym wypadku.

*Miejsce na wpisanie funkcji:*



	Nr zadania	2.1	2.2	2.3	Suma
Wypełnia egzaminator	Maksymalna liczba punktów	2	1	3	6
	Uzyskana liczba punktów				

### Zadanie 3. Zliczanie (0-5)

#### Zadanie 3.1 (0-1)

Przeanalizuj działanie podanego poniżej algorytmu. Przyjmij, że tablica  $T$  zawiera  $n$  liczb całkowitych z przedziału  $[0, k]$ .

**Krok 1:** Dla  $i = 0, 1, 2, \dots, k$  wykonaj  $Z[i] = 0$ .

**Krok 2:** Dla  $i = 0, 1, 2, \dots, n - 1$  wykonaj  $Z[T[i]] = Z[T[i]] + 1$ .

Uzupełnij specyfikację dla przedstawionego powyżej algorytmu:

**Dane:**

- liczba naturalna  $n > 0$  (liczba elementów tablicy  $T$ ),
- liczba całkowita  $k \geq 0$ ,
- $n$  elementowa tablica  $T$  zawierająca liczby całkowite z przedziału  $[0, k]$ .

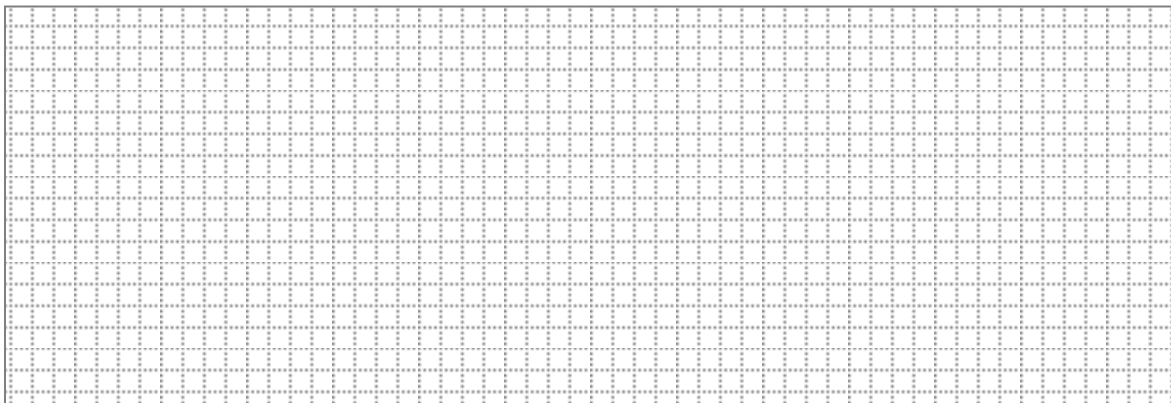
**Wynik:**

..... elementowa tablica  $Z$  zawierająca.....  
.....  
.....

Wypełnij tabelę zgodnie z podanym powyżej algorytmem:

$n$	$k$	$T$	$Z$
5	5	1, 3, 5, 4, 0	1, 1, 0, 1, 1, 1
8	10	0, 0, 9, 9, 3, 4, 1, 1	
7	8	3, 3, 3, 3, 3, 3, 3	

**Miejsce na obliczenia:**



### Zadanie 3.2 (0–1)

W tablicy liczb  $T$  szukamy wartości występujących najczęściej i wartości występujących najrzadziej. Jeżeli takich wartości jest w tablicy więcej niż jedna, to powinny być wypisane wszystkie takie liczby.

#### Przykłady

Dla ciągu (5, 4, 4, 8, 0, 1, 2, 4, 6, 5, 0) liczbą występującą najczęściej jest 4, natomiast liczbami występującymi najrzadziej są 8, 2, 6 i 1.

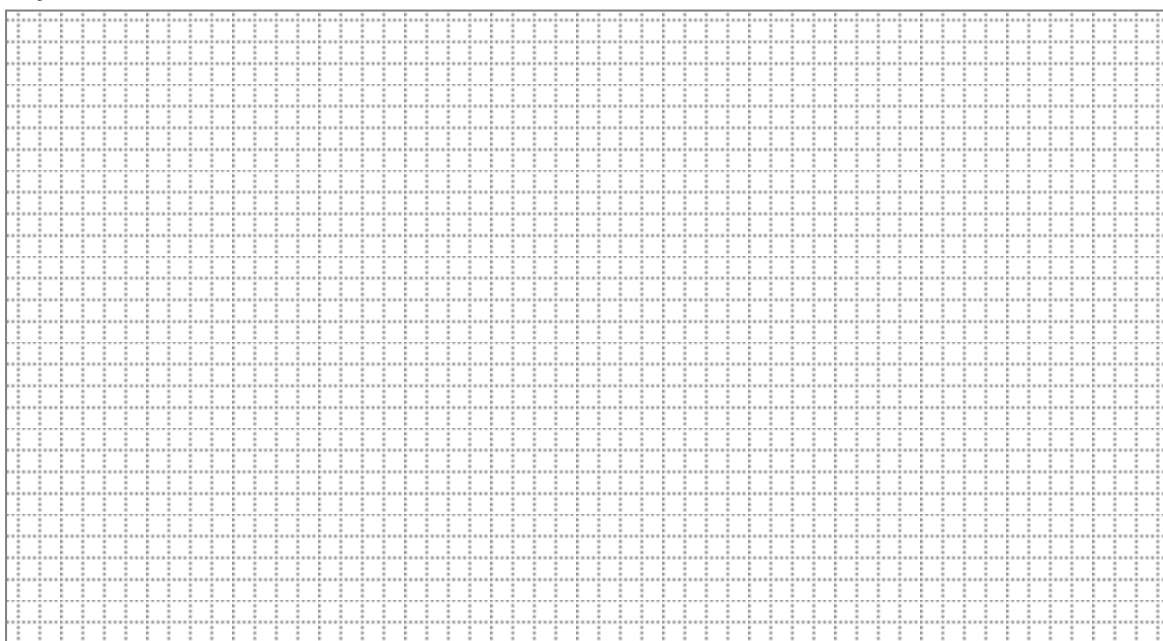
Dla ciągu (4, 8, 4, 7, 8, 8, 5, 5) liczbą występującą najczęściej jest 8, natomiast najrzadziej pojawia się wartość 7.

W ciągu (2, 2, 2, 2, 2, 2, 2, 2, 2) liczba 2 jest wartością występującą najczęściej i najrzadziej.

Uzupełnij poniższą tabelę, podając wyniki, jakie powinny pojawić się dla podanych ciągów.

<i>Ciąg</i>	<i>Liczba występująca najczęściej</i>	<i>Liczba występująca najrzadziej</i>
(4, 5, 6, 6, 3, 4, 4, 9, 2, 7, 9, 1)		
(0, 0, 0, 0, 3, 0, 0, 0)		
(4, 5, 6, 7, 8, 7, 6, 5, 4, 8)		
(1, 1, 1, 3, 5, 3, 5, 3, 5, 3, 1)		

#### Miejsce na obliczenia:





### Zadanie 3.3. (0-3)

W wybranej przez siebie notacji (lista kroków lub wybrany przez Ciebie język programowania) zapisz algorytm dla podanej niżej specyfikacji.

**Dane:**

liczba naturalna:  $n > 0$  (liczba elementów w tablicy  $T$ ).

liczba całkowita:  $k \geq 0$

$n$  elementowa tablica  $T$  zawierająca liczby całkowite z przedziału  $[0, k]$ .

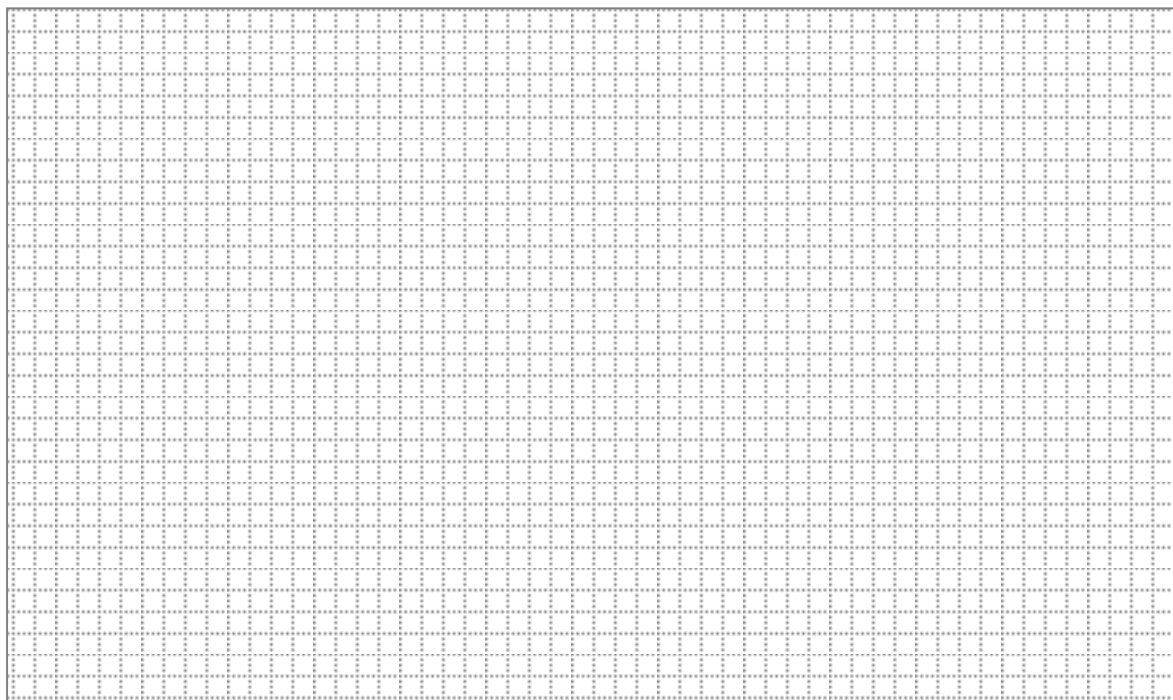
**Wyniki:**

wszystkie liczby całkowite występujące najczęściej w tablicy  $T$ ,

wszystkie liczby całkowite występujące najrzadziej w tablicy  $T$ .

**Wskazówka:** W rozwiązaniu możesz wykorzystać algorytm z zadania 3.1.

*Miejsce na wpisanie algorytmu:*



	Nr zadania	3.1	3.2	3.3	Suma
Wypełnia egzaminator	Maksymalna liczba punktów	1	1	3	5
	Uzyskana liczba punktów				

**BRUDNOPIS** (*nie podlega ocenie*)