Województwo zachodniopomorskie

Informatyka

Sprawozdanie z egzaminu maturalnego w roku 2016

Opracowanie
Agata Kordas-Łata (Okręgowa Komisja Egzaminacyjna w Jaworznie)
Elżbieta Wierzbicka (Okręgowa Komisja Egzaminacyjna w Gdańsku)

Redakcja
dr Wioletta Kozak (Centralna Komisja Egzaminacyjna)

Opracowanie techniczne
Bartosz Kowalewski (Centralna Komisja Egzaminacyjna)

Współpraca
Beata Dobrosielska (Centralna Komisja Egzaminacyjna)
Agata Wiśniewska (Centralna Komisja Egzaminacyjna)
Pracownie ds. Analiz Wyników Egzaminacyjnych okręgowych komisji egzaminacyjnych

Opracowanie dla województwa zachodniopomorskiego

Okręgowa Komisja Egzaminacyjna w Poznaniu
Izabela Szafrańska
Jacek Pietrzak

Centralna Komisja Egzaminacyjna
ul. Józefa Lewartowskiego 6, 00-190 Warszawa
tel. 022 536 65 00, fax 022 536 65 04
e-mail: sekretariat@cke.edu.pl
www.cke.edu.pl

Informatyka

Poziom rozszerzony

1. Opis arkusza

Egzamin maturalny z informatyki składał się z dwóch części: pisemnej (arkusz I) oraz praktycznej (arkusz II). Zadania sprawdzały opanowanie wymagań zapisanych w podstawie programowej i odnosiły się do głównych treści kształcenia realizowanych w szkołach. Tegoroczny zestaw egzaminacyjny zachował podstawową strukturę dotychczasowych arkuszy i zawierał: zadania dotyczące tworzenia algorytmów, zadania polegające na analizie algorytmów, zadania zamknięte sprawdzające podstawową wiedzę z różnych obszarów informatyki, zadania programistyczne, zadania bazodanowe oraz zadania dedykowane pod arkusz kalkulacyjny, które można było również rozwiązać, pisząc program komputerowy.

Zadania 1.1. i 1.2. to typowe zadania sprawdzające umiejętność myślenia algorytmicznego oraz zapisu algorytmu w wybranej przez zdającego notacji. Zadania 2.1., 2.2., 2.3. i 3.2. sprawdzały umiejętność analizy algorytmów. Zadanie 3.1. dotyczyło wiedzy na temat korzystania z usług w globalnej sieci komputerowej. Zadanie 3.3. weryfikowało znajomość działań wykonywanych na liczbach zapisanych w systemie binarnym. Zadanie 3.4. odnosiło się do roli i funkcji systemu operacyjnego w systemie komputerowym. Zadanie 4. sprawdzało umiejętności programowania numerycznego (prosta arytmetyka na liczbach rzeczywistych, zaokrąglenia, itp.) oraz wykorzystania narzędzia technologii informacyjnej – arkusza kalkulacyjnego, zadanie 5. sprawdzało umiejętności posługiwania się relacyjnymi bazami danych. Zadania można było rozwiązać przy użyciu różnorodnych narzędzi: programem bazodanowym (np. Access), arkuszem kalkulacyjnym, programem w języku bazodanowym (np. SQL) lub w dowolnym języku programowania. Zadanie 6. polegało na kodowaniu, dekodowaniu i sprawdzaniu poprawności kodu przy szyfrowaniu metodą Cezara. W istocie zadanie to sprawdzało zrozumienie i umiejętność posługiwania się kodami ASCII i prostą arytmetyką modularną.

Arkusz I zestawu egzaminacyjnego zawierał 3 zadania (9 poleceń), za które zdający mógł uzyskać maksymalnie 15 punktów. Arkusz II zawierał 3 zadania (11 poleceń), za które zdający mógł uzyskać 35 punktów. Egzamin trwał 60 minut w części I i 150 minut w części II.

2. Dane dotyczące populacji zdających

Tabela 1. Zdający rozwiązujący zadania w arkuszu standardowym*

Liczba zdających		**255**
Zdający rozwiązujący zadania w arkuszu standardowym	z liceów ogólnokształcących	88
	z techników	167
	ze szkół na wsi	1
	ze szkół w miastach do 20 tys. mieszkańców	36
	ze szkół w miastach od 20 tys. do 100 tys. mieszkańców	63
	ze szkół w miastach powyżej 100 tys. mieszkańców	155
	ze szkół publicznych	223
	ze szkół niepublicznych	32
	kobiety	21
	mężczyźni	234

* Dane w tabeli dotyczą tegorocznych absolwentów.

Tabela 2. Zdający rozwiązujący zadania w arkuszach dostosowanych

Zdający rozwiązujący zadania w arkuszach dostosowanych	z autyzmem, w tym z zespołem Aspergera	1
	słabowidzący	0
	niewidomi	0
	słabosłyszący	0
	niesłyszący	0
	ogółem	**1**

3. Przebieg egzaminu

Tabela 3. Informacje dotyczące przebiegu egzaminu

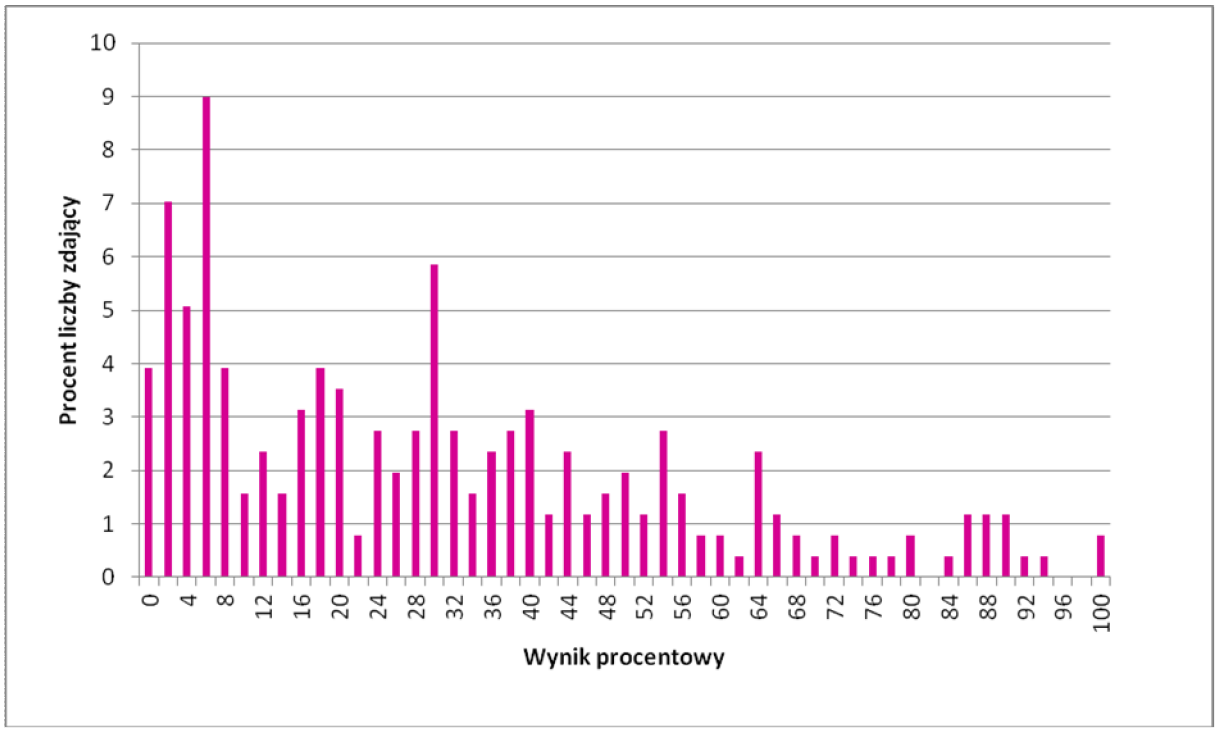
Termin egzaminu			17 maja 2016 r.
Czas trwania egzaminu			180 minut
Liczba szkół			57
Liczba zespołów egzaminatorów			1
Liczba egzaminatorów			14
Liczba obserwatorów[1] (§ 8 ust. 1)			0
Liczba unieważnień[2]	w przypadku:		
	art. 44zzv pkt 1	stwierdzenia niesamodzielnego rozwiązywania zadań przez zdającego	0
	art. 44zzv pkt 2	wniesienia lub korzystania przez zdającego w sali egzaminacyjnej z urządzenia telekomunikacyjnego	0
	art. 44zzv pkt 3	zakłócenia przez zdającego prawidłowego przebiegu egzaminu	0
	art. 44zzw ust. 1.	stwierdzenia podczas sprawdzania pracy niesamodzielnego rozwiązywania zadań przez zdającego	0
	art. 44zzy ust. 7	stwierdzenia naruszenia przepisów dotyczących przeprowadzenia egzaminu	0
	art. 44zzy ust. 10	niemożności ustalenia wyniku (np. zaginięcie karty odpowiedzi)	0
Liczba wglądów[2] (art. 44zzz)			0
Liczba prac, w których nie podjęto rozwiązania zadań			0

[1] Na podstawie rozporządzenia Ministra Edukacji Narodowej z dnia 25 czerwca 2015 r. w sprawie szczegółowych warunków i sposobu przeprowadzania sprawdzianu, egzaminu gimnazjalnego i egzaminu maturalnego (Dz.U. z 2015, poz. 959).

[2] Na podstawie ustawy z dnia 7 września 1991 r. o systemie oświaty (tekst jedn. Dz.U. z 2015, poz. 2156, ze zm.).

4. Podstawowe dane statystyczne

Wyniki zdających



Wykres 1. Rozkład wyników zdających

Tabela 4. Wyniki zdających – parametry statystyczne*

Zdający	Liczba zdających	Minimum (%)	Maksimum (%)	Mediana (%)	Średnia (%)	Odchylenie standardowe (%)
ogółem	**256**	**0**	**100**	**26**	**30**	**25**
w tym:						
z liceów ogólnokształcących	88	0	100	43	45	26
z techników	168	0	88	17	22	20

* Dane dotyczą tegorocznych absolwentów.

5. Poziom wykonania zadań

Tabela 5. Poziom wykonania zadań

Nr zad.	Wymaganie ogólne	Wymaganie szczegółowe	Poziom wykonania zadania (%)
1.1.	III. Rozwiązywanie problemów i podejmowanie decyzji […], z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji […], stosowanie podejścia algorytmicznego. Zdający: 2) stosuje podejście algorytmiczne do rozwiązywania problemu.	67
1.2.	III. Rozwiązywanie problemów i podejmowanie decyzji […], z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji […], stosowanie podejścia algorytmicznego. Zdający: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wy branej notacji; 11) opisuje podstawowe algorytmy i stosuje: a) algorytmy na liczbach całkowitych; 18) oblicz liczbę operacji wykonywanych przez algorytm.	22
2.1.	III. Rozwiązywanie problemów i podejmowanie decyzji […], z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji […], stosowanie podejścia algorytmicznego. Zdający: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] b) algorytmy wyszukiwania i porządkowania (sortowania); 17) Zdający ocenia zgodność algorytmu ze specyfikacją problemu.	31
2.2.	III. Rozwiązywanie problemów i podejmowanie decyzji […], z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji […], stosowanie podejścia algorytmicznego. Zdający: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] b) algorytmy wyszukiwania i porządkowania (sortowania); 17) Zdający ocenia zgodność algorytmu ze specyfikacją problemu.	53
2.3.	III. Rozwiązywanie problemów i podejmowanie decyzji […], z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji […], stosowanie podejścia algorytmicznego. Zdający: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od	39

		sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] b) algorytmy wyszukiwania i porządkowania (sortowania); 17) Zdający ocenia zgodność algorytmu ze specyfikacją problemu.	
3.1.	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej. Komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.	1. Posługiwanie się komputerem i jego oprogramowaniem, korzysta z sieci komputerowej. Zdający: 3) przedstawia warstwowy model sieci komputerowych, określa ustawienia sieciowe dane go komputera i jego lokalizacji w sieci, opisuje zasady administrowania siecią komputerową w architekturze klient-serwer, prawidłowo posługuje się terminologią sieciową, korzysta z usług w sieci komputerowej, lokalnej i globalnej, związanych z dostępem do informacji, wymianą informacji i komunikacją.	51
3.2.	III. Rozwiązywanie problemów i podejmowanie decyzji […], z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji […], stosowanie podejścia algorytmicznego. Zdający: 9) stosuje rekurencję w prostych sytuacjach problemowych.	45
3.3.	III. Rozwiązywanie problemów i podejmowanie decyzji […], z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji […], stosowanie podejścia algorytmicznego. Zdający: 11) opisuje podstawowe algorytmy i stosuje a) algorytmy na liczbach całkowitych.	64
3.4.	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.	1. Posługiwanie się komputerem i jego oprogramowaniem, korzysta z sieci komputerowej. Zdający: 2) wyjaśnia funkcje systemu operacyjnego i korzysta z nich; opisuje różne systemy operacyjne.	45

W związku z brakiem kilku danych w pliku *dane_6_2.txt* do zadania 6.2., które mogły wpłynąć lub utrudnić poprawne rozwiązanie zadania, Centralna Komisja Egzaminacyjna umożliwiła zdającym ponowne przystąpienie do egzaminu maturalnego z informatyki w terminie dodatkowym, z którego skorzystało około 370 zdających w całym Okręgu. Zdający ci przystąpili wyłącznie do arkusza II, tj. wykonania zadań z wykorzystaniem komputera i ten wynik był brany pod uwagę przy ustalaniu ostatecznego wyniku maturalnego z informatyki. W przypadku części pierwszej brany był pod uwagę wynik uzyskany w egzaminie majowym. Poziom wykonania wszystkich zadań z arkusza II wyniósł 26%.

Nr zadania	**Wymaganie ogólne**	**Wymaganie szczegółowe**
4.1.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji	4. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Zdający:

	z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych. III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.	4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. 5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania.
4.2.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych. III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.	4. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Zdający: 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. 5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania.
4.3.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych. III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.	4. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. Zdający: 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. 5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania.
5.1.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Zdający: 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL); 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji.
5.2.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł;	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Zdający:

	opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.	1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL); 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji.
5.3.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Zdający: 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL); 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji.
5.4.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Zdający: 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL); 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji.
5.5.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Zdający: 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL); 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji.
6.1.	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Zdający 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 5) posługuje się podstawowymi technikami algorytmicznymi; 6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: f) algorytmy kompresji i szyfrowania; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe,

		rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.
6.2.	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Zdający 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 5) posługuje się podstawowymi technikami algorytmicznymi; 6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: f) algorytmy kompresji i szyfrowania; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.
6.3.	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Zdający 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 5) posługuje się podstawowymi technikami algorytmicznymi; 6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: f) algorytmy kompresji i szyfrowania; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.

Komentarz

1. Analiza jakościowa

Analiza odpowiedzi do zadań wskazuje na duże zróżnicowanie poziomu umiejętności zdających, dlatego w komentarzu zwrócono szczególną uwagę na mocne i słabe strony, dotyczące umiejętności zdających.

Maturzyści dobrze radzili sobie z rozwiązywaniem zadań dotyczących wyszukiwania i przetwarzania danych. Zdecydowana większość zdających do ich rozwiązywania wybierała aplikację bazodanową Access. W pojedynczych przypadkach korzystali z aplikacji Base z pakietu Open Office lub formułowali zapytania w języku SQL. Z zestawu zadań: 5.1., 5.2., 5.3., 5.4. i 5.5. najwyższy poziom wykonania osiągnęły zadania 5.1. i 5.4. W pierwszym przypadku wystarczyło utworzyć relacje pomiędzy dwoma tabelami `Studenci` i `Wypożyczenia` oraz wykonać grupowanie wg nazwisk i policzyć dla każdego nazwiska liczbę wypożyczeń. Po znalezieniu nazwiska osoby, która wypożyczyła największą liczbę książek, należało odfiltrować tytuły wypożyczonych książek.

Przykładowe rozwiązanie w języku SQL:

```
SELECT IMIE, NAZWISKO, TYTUL
FROM
(
      SELECT MAX(LICZBA_KSIAZEK), NAZWISKO, IMIE, PESEL
      FROM
      (
            SELECT PESEL, IMIE, NAZWISKO, COUNT(TYTUL) AS LICZBA_KSIAZEK
            FROM STUDENCI NATURAL JOIN WYPOZYCZENIA
            GROUP BY PESEL
      ) AS LISTA
) AS DANE NATURAL JOIN WYPOZYCZENIA;
```

W zadaniu 5.4. wystarczyła jedna kwerenda, pod warunkiem że pomiędzy tabelami zostało utworzone sprzężenie, uwzględniające wszystkie rekordy z tabeli `Studenci` i tylko te rekordy z tabeli `Meldunek`, dla których sprzężone pola są równe (`LEFT JOIN`), kolejno wystarczyło wybrać wszystkie imiona i nazwiska, dla których pole PESEL z tabeli `Meldunek` jest puste.

Przykładowe rozwiązanie w języku SQL:

```
SELECT STUDENCI.NAZWISKO, STUDENCI.IMIE
FROM STUDENCI
    LEFT JOIN MELDUNEK ON (STUDENCI.PESEL = MELDUNEK.PESEL)
WHERE MELDUNEK.PESEL IS NULL
ORDER BY STUDENCI.NAZWISKO;
```

Prawidłowo rozwiązywano również zadanie 4.1., w którym należało podać współrzędne punktów, z pliku `punkty.txt`, należących do brzegu koła o współrzędnych środka (200; 200) i promieniu 200 oraz podać liczbę punktów należących do wnętrza koła. Zdecydowana większość zdających zapisywała odpowiednią formułę w arkuszu kalkulacyjnym, ale była również wśród zdających grupa osób rozwiązująca opisany problem programistycznie.

Przykład rozwiązania za pomocą programu:

```
#include <stdio.h>
#include <stdlib.h>

FILE *dane;
FILE *wynik;

main()
{
      int pktwew = 0;
```

```
	int x,y;

	dane = fopen("punkty.txt", "rt");
	wynik = fopen("wyniki_5_1.txt", "w");

	fprintf(wynik,"Punkty nalezace do brzegu kola to:\n");
	while(1)
	{
		fscanf(dane,"%d %d\n", &x,&y);
		if( ((x-200)*(x-200))+((y-200)*(y-200)) == 200*200)
			fprintf(wynik,"\t%d %d\n",x,y);
		if( ((x-200)*(x-200))+((y-200)*(y-200)) < 200*200) pktwew++;
		if (feof(dane)) break;
	}

	fprintf(wynik,"\nLiczba punktow we wnetrzu kola: %d\n",pktwew);

	fclose(dane);
	fclose(wynik);

	return 0;
}
```

Trudno jednoznacznie wskazać obszary, które zostały opanowane przez maturzystów na niskim poziomie. Zadania z każdego typu miały różny poziom trudności i zdający, w zależności od swojej wiedzy i posiadanych umiejętności, dochodzili do pewnego etapu i zatrzymywali się na zagadnieniach, które przekraczały ich kompetencje. Tradycyjnie, najczęściej opuszczane były zadania programistyczne, choć wykorzystywany algorytm szyfrowania należy do tych najprostszych i jest omawiany jako jeden z pierwszych na lekcjach informatyki.

Najczęstsze błędy w zadaniach 6.1., 6.2. i 6.3. to:

- błędne wykonanie zawinięcia cyklicznego lub brak zawijania
- przyjęcie błędnej długości alfabetu, błąd w pętli (pominięcie ostatniego wiersza, pominięcie ostatniego znaku, wyjście poza długość słowa)
- problemy z wczytaniem danych oraz zapisaniem wyników do pliku tekstowego.

Poniżej przedstawione zostały przykładowe prawidłowe kody programów:

```
#include<iostream>
#include<fstream>
#include<string>

using namespace std;

string szyfruj(string s, int k)
{
    for(int i=0;i<s.size();i++)
    {
        s[i]+=k;
        if(s[i]>90)
            s[i]-=26;
    }
    return s;
}

string deszyfruj(string s, int k)
{
    for(int i=0;i<s.size();i++)
    {
        s[i]-=k;
        if(s[i]<65)
            s[i]+=26;
    }
    return s;
```

```
}

int main()
{
ifstream pl;
ofstream zapis;

//  6.1
    pl.open("dane_6_1.txt");
    zapis.open("wyniki_6_1.txt");

    for(int i=0;i<100;i++)
    {
        string s;
        pl>>s;
        int k = 107%26;
        zapis<<szyfruj(s,k)<<endl;
    }

    pl.close();
    zapis.close();


//6.2
    pl.open("dane_6_2.txt");
    zapis.open("wyniki_6_2.txt");

    for(int i=0;i<3000;i++)
    {
        string s;
        int k;
        pl>>s>>k;
        k %= 26;
        zapis<<deszyfruj(s,k)<<endl;
    }

    pl.close();
    zapis.close();


//6.3
    pl.open("dane_6_3.txt");
    zapis.open("wyniki_6_3.txt");

    for (int i=0;i<3000;i++)
    {
        string s1,s2;
        pl>>s1>>s2;

        int k=s2[0]-s1[0];
        if(k<0)
            k+=26;

        string s=szyfruj(s1,k);

        if(s!=s2)
        {
            zapis<<s1<<endl;
        }

    }

    pl.close();
    zapis.close();

    return 0;
}
```

2. Problem „pod lupą”. Stosowanie prawidłowego zapisu algorytmu w postaci listy kroków, pseudokodu lub języka programowania wybranego przez zdającego

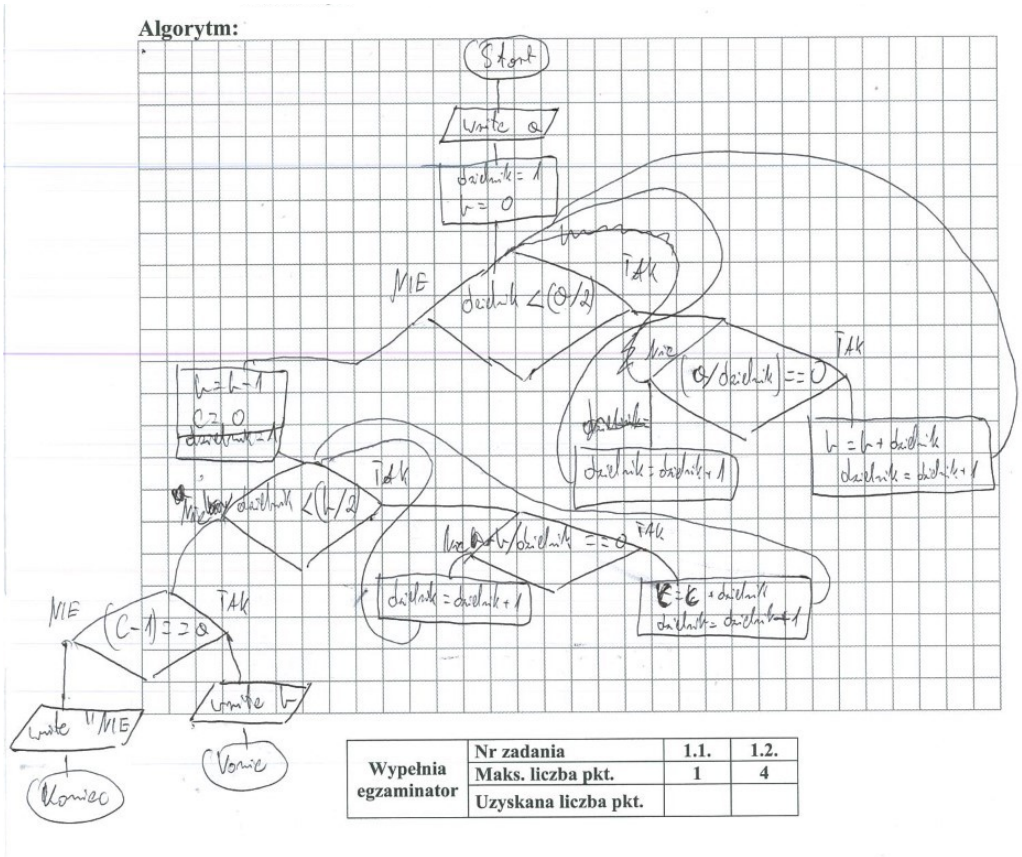
Zadania 1.1. i 1.2. to zadania dotyczące tworzenia algorytmów. Sprawdzają one podstawowe umiejętności zapisane w podstawie programowej. Główną trudnością zadania było napisanie algorytmu, który obliczy sumę dzielników podanej liczby *a*, ustali liczbę *b*, obliczy sumę dzielników liczby *b*, sprawdzi, czy liczby *a* i *b* są skojarzone. Dodatkowo można było uzyskać 1 punkt za wydajność algorytmu (sprawdzanie potencjalnych dzielników do $\sqrt{n}$).

W instrukcji dla zdającego na pierwszej stronie w punkcie 7. zapisana jest informacja o sposobie zapisu algorytmu:

> „Jeżeli rozwiązaniem zadania lub jego części jest algorytm, to zapisz go w notacji wybranej przez siebie: listy kroków, pseudokodu lub języka programowania, który wybierasz na egzamin.”

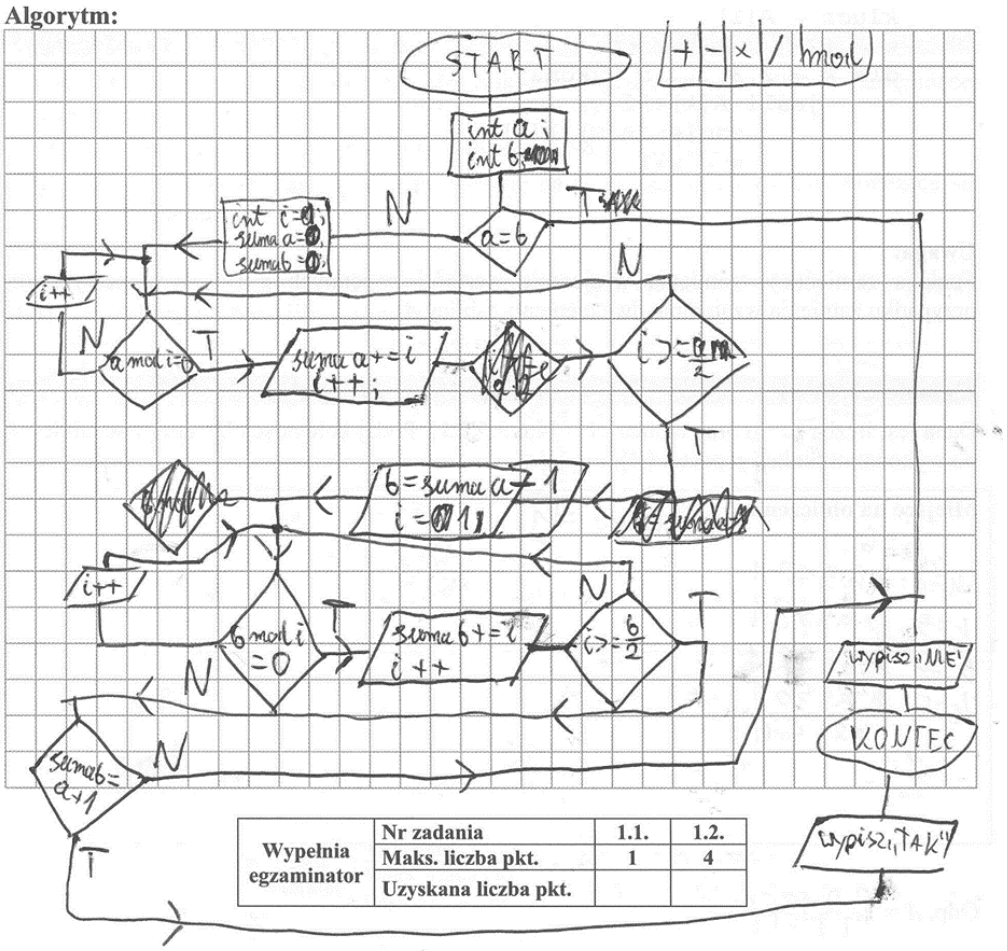
W instrukcji dla zdającego nie wymieniono schematu blokowego jako możliwego sposobu zapisu algorytmu, a pomimo to zdarzało się, że zdający wybierali właśnie taką opcję (niestety, nie była to dobra decyzja). Autorzy zadań celowo pomijają schemat blokowy, gdyż jego wykonanie jest czasochłonne i łatwo popełnić w nim błędy, na co wskazują poniższe przykłady rozwiązań zdających:

Przykład 1.



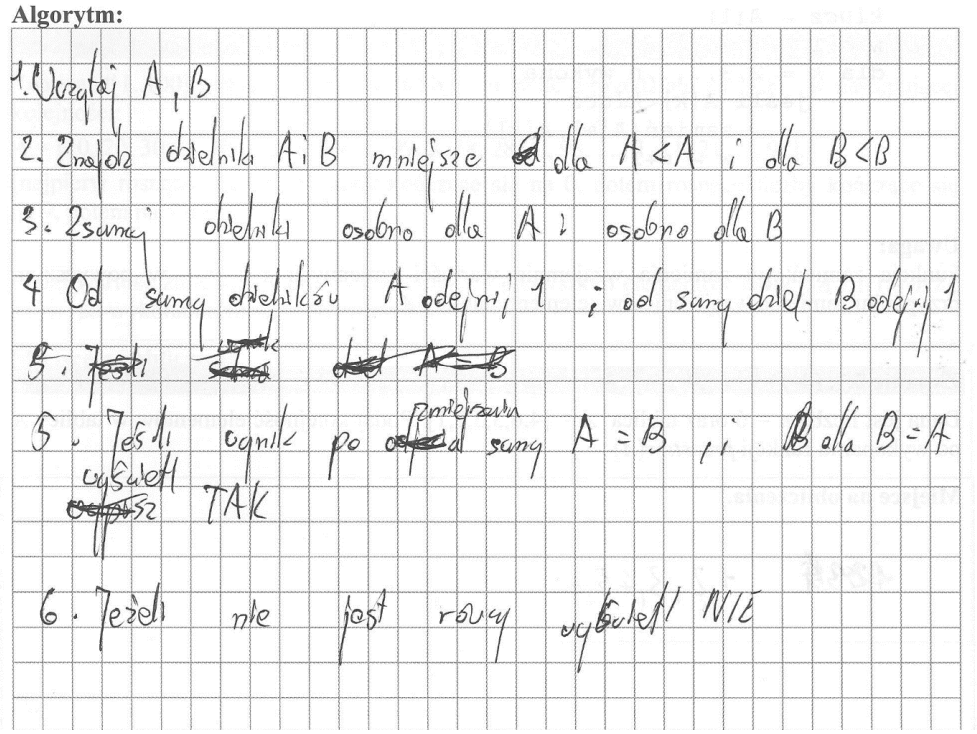
Wypełnia egzaminator	Nr zadania	1.1.	1.2.
	Maks. liczba pkt.	1	4
	Uzyskana liczba pkt.		

Przykład 2.



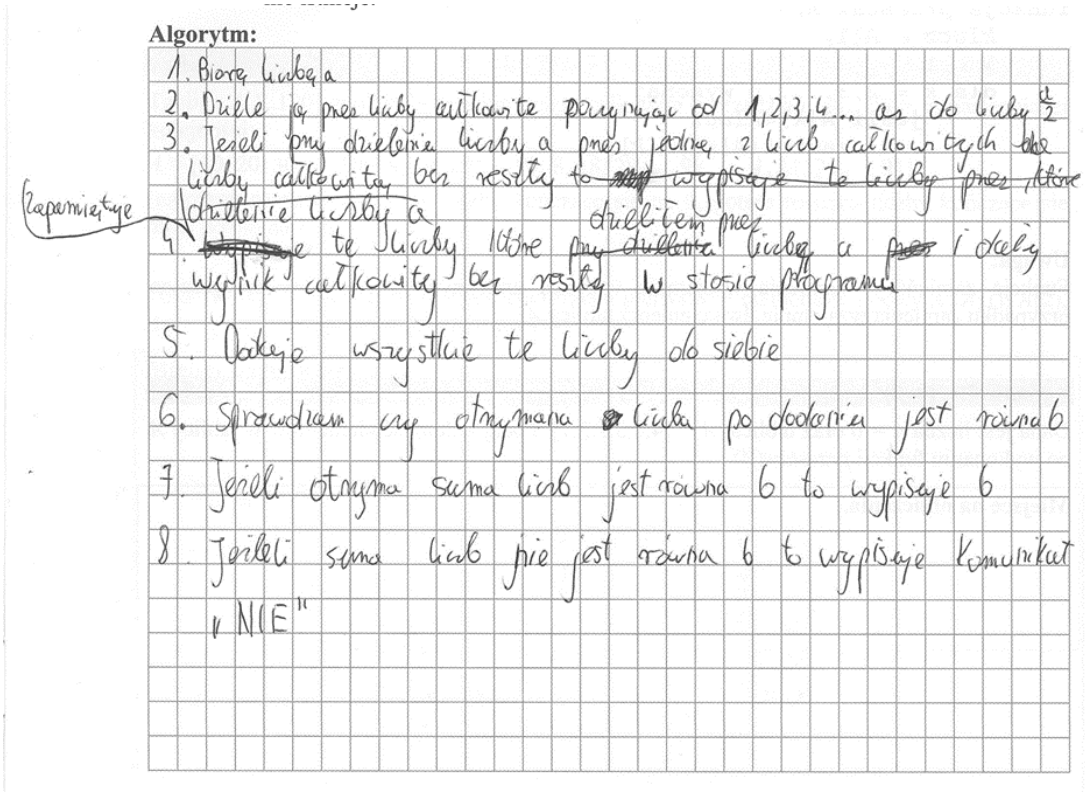
Kolejną możliwością zapisu algorytmu jest lista kroków. Zdający często mylą listę kroków z ponumerowaniem zdań opisu problemu podanego w tekście zadania. Za poniższe przykładowe rozwiązanie (sugerujące zastosowanie listy kroków) maturzysta otrzymał 0 punktów. Zdający wczytał liczbę *b*, która nie jest dana, ale należy ją wyznaczyć; nie pokazał, w jaki sposób wyznacza dzielniki liczby *a*, ani do jakiej zmiennej zapisuje sumę dzielników; dodatkowo jako wynik podał „TAK” lub „NIE”, zamiast liczbę *b* lub „NIE”. Poniższy zapis to nieudolna próba rozwiązania innego problemu niż ten podany w treści zadania i w specyfikacji.

Przykład 3.

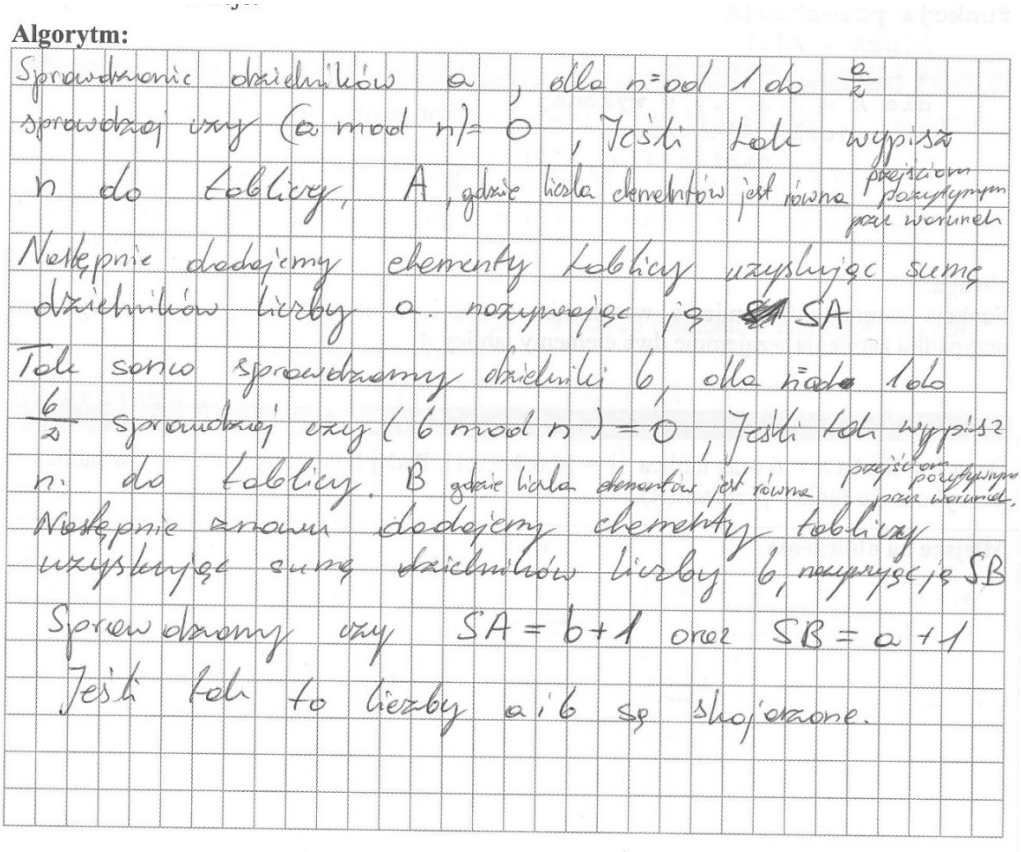


W dwóch poniższych fragmentach prac zdający uzyskał jedynie 1 punkt za poprawną konstrukcję pętli przy wyznaczaniu dzielników liczby *a*. Trudno przyznać punkt za sumowanie kolejnych dzielników, bo nie można rozstrzygnąć, czy dzielniki zostały prawidłowo zapisane do tablicy i prawidłowo zsumowane. Dodatkowo zdający zakłada, że liczba *b* jest daną, a przecież powinien ją dopiero wyznaczyć.

Przykład 4.



Przykład 5.



W poniższej notacji za pomocą listy kroków została zapisana pętla (można sprawdzić jej zakres) oraz obliczenia sumy dzielników (wartość początkowa zmiennej i sumowania kolejnych dzielników).

Przykład 6.

Błąd! Nie można odnaleźć źródła odwołania.

Najszybszy do napisania, zwięzły i czytelny jest zapis w pseudokodzie. Nie istnieje jednoznacznie zdefiniowana składnia pseudojęzyka. Oto kilka propozycji do stosowania w zapisie algorytmu.

Bloki kodu zaznaczamy, używając samych wcięć (bez *begin* ... *end* czy klamer)

```
k ← 1
s ← 1
dopóki k ≤ n wykonuj
      s ← s * k
      k ← k + 1
```

Dla łatwości odróżnienia zmienne skalarne piszemy z małej litery, tablicowe - z wielkiej.
Operatorem przypisania jest strzałka (←).
Instrukcję zamiany (ang. *swap*) zapisujemy za pomocą operatora ↔

```
x ↔ y   // zamień x z y
```

Operatory arytmetyczne: +, -, *,
div, **mod** *(tylko dla liczb całkowitych)*
/ (tylko dla liczb rzeczywistych)
Operatory porównania: <, >, ≤, ≥, =, ≠
logiczne jako słowa: ***i, lub, nie jest***
prawda, fałsz jako wartości logiczne

Instrukcja powtarzania (while)

```
dopóki x > 1 wykonuj
      p ← p + 1
      x ← x div 2
```

Instrukcja powtarzania (do .. while)

```
wykonuj
      p ← p + 1
      x ← x div 2
dopóki x > 1
```

Instrukcja iteracji (while true)

```
wykonuj
      p ← p + 1
      jeżeli x mod 2 = 0
            x ← x div 2
      w przeciwnym razie
            zakończ
```

Instrukcja iteracji (for)

```
dla s = 1, 2, … k wykonuj
      p ← p * s
```

Instrukcja warunkowa (if)

```
jeżeli x mod 2 = 0
      p ← p * p
w przeciwnym razie
      p ← p * s
```

Funkcje

```
funkcja silnia(n)
```

```
      w ← 1
      dla k = 1, 2, …, n wykonuj
            w ← w * k
      zwróć w i zakończ
```

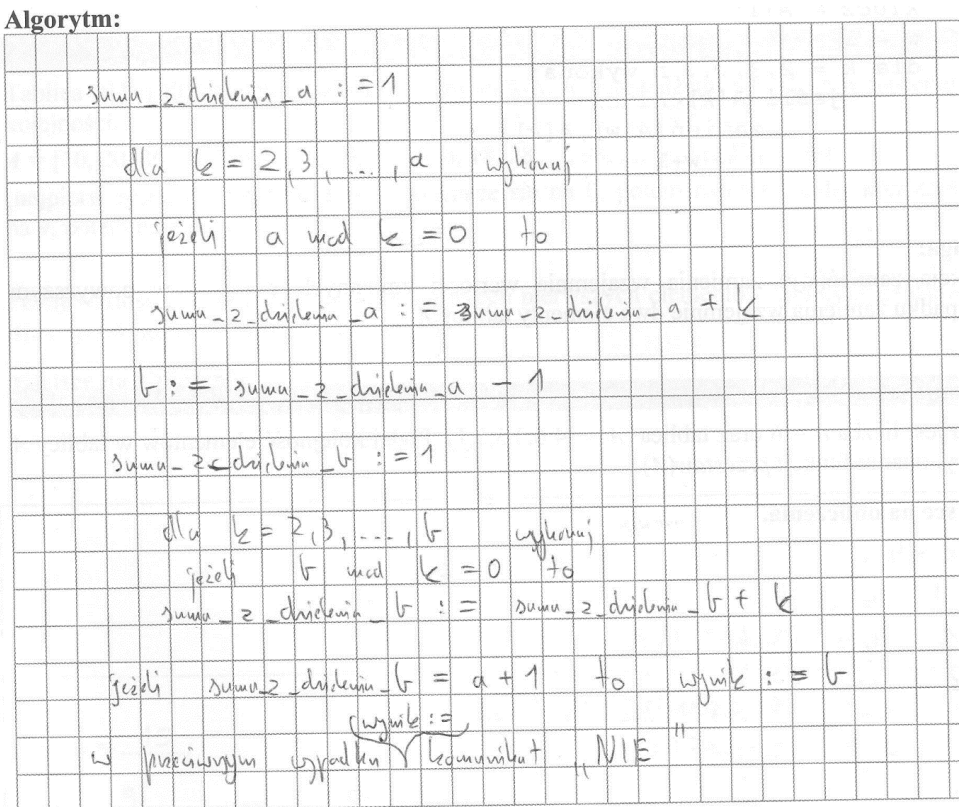
Przy zastosowaniu powyżej opisanego pseudokodu prawidłowe rozwiązanie wyglądałoby następująco:

```
funkcja sumadz(n)
      suma ← 1
      i ← 2
      dopóki i*i ≤ n wykonuj
            jeżeli n mod i = 0
                  suma ← suma + i
                  jeżeli n div i ≠ i
                        suma ← suma + n/i
            i ← i + 1
      zwróć suma

x ← sumadz(a)
y ← sumadz(x-1)
jeżeli y-1 = a
      wypisz x-1
      w przeciwnym wypadku wypisz „NIE”
```

Kolejne rozwiązanie maturzysty, zapisane w pseudokodzie (innym niż przedstawiona propozycja), zostało oceniony na 3 punkty (brak 1 punktu za algorytm o złożoności nie gorszej niż $\sqrt{n}$).

Przykład 7.



Inną bezpieczną notacją jest zapis w języku programowania (na kartce można pominąć dołączane biblioteki), wybranym przez zdającego. Należy jednak nadal pamiętać o uwagach zamieszczonych w treści zadania, nakładających ograniczenia:

> „W zapisie algorytmu możesz korzystać tylko z następujących operacji arytmetycznych: dodawania, odejmowania, mnożenia, dzielenia całkowitego i obliczania reszty z dzielenia."

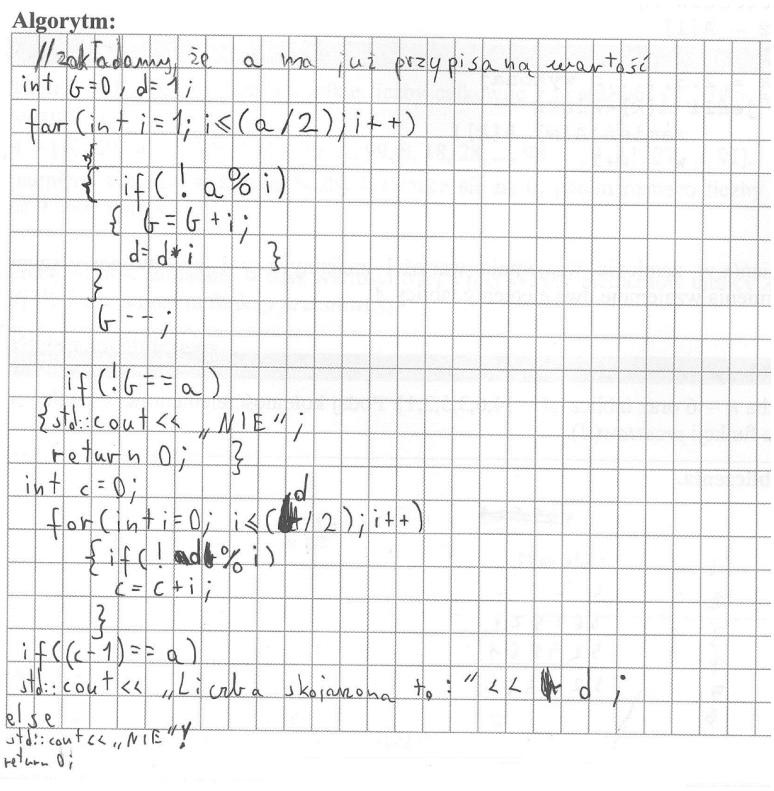
Warto zwrócić uwagę na stosowane operatory: zdarza się, że zdający wymiennie (błędnie) stosują *div* i *mod* (/ lub %), czy używają w zapisie symbolu matematycznego dzielenia „:" (nie wiadomo wtedy, jakie dzielenie zdający miał na myśli). Jeżeli niedopuszczalny jest pierwiastek, to można wykonać zapis równoważny, np. $i \cdot i < a$. Przykładowe rozwiązania w postaci kodu programu komputerowego.

Przykład 8.

Algorytm:

```
int s=0;
int b=0;
int pom=2;
while ((pom*pom)<a)
{
  if (a%pom==0) b=b+pom+(a/pom);
  pom++;
}
if ((pom*pom)==a) b=b+pom;
pom=2;
while ((pom*pom)<b)
{
  if (b%pom==0) s=s+pom+(b/pom);
  pom++
}
if ((pom*pom)==b) s=s+pom;
if (s==a) return b; else cout << "NIE";
```

Przykład 9.



3. Wnioski i rekomendacje

Poniżej przedstawiono kilka wskazówek, jak poprawić wyniki osiągane przez maturzystów oraz w jaki sposób ukierunkować przygotowania uczniów do przyszłej matury.

1. W dużej mierze rezultaty osiągane przez uczniów na egzaminie zależą od stosowanych przez nauczyciela informatyki na lekcjach strategii działań. Warto, aby nauczyciel w czasie zajęć z informatyki zwracał uwagę i eliminował błędne nawyki uczniów oraz niestaranność w rozwiązywaniu zadań (nieuważne czytanie poleceń, pomijanie pewnych elementów polecenia, podawanie wyników z inną dokładnością niż żądana w zadaniu, brak opisów na wykresach itp.).
2. Ważne jest, aby uczeń już na sprawdzianach przyzwyczajał się do formy prezentowania rozwiązania (plik tekstowy z odpowiedziami oraz dołączone pliki komputerowej realizacji rozwiązania). Zdarzają się bowiem prace bez dołączonej komputerowej realizacji obliczeń, co powoduje, że uczeń otrzymuje za zadanie 0 punktów. Wszystkie wyniki otrzymane podczas rozwiązywania zadań muszą być odzwierciedleniem komputerowej realizacji i oprócz wyników zawartych w plikach tekstowych należy obowiązkowo dołączyć pliki źródłowe, zawierające rozwiązania poszczególnych zadań. Dodatkowo administratorzy nagrywający na koniec egzaminu maturalnego pracę zdającego na płytę powinni zwrócić uwagę na konieczność zapisu wszystkich niezbędnych plików.
3. Biorąc pod uwagę zakres materiału z informatyki, realizowany w szkole ponadgimnazjalnej, można stwierdzić, iż zdający nadal mają najwięcej problemów z zadaniami dotyczącymi algorytmiki i programowania, a przecież są to dla informatyka umiejętności kluczowe i dlatego tym zagadnieniom należy poświęcić najwięcej czasu podczas przygotowań. W procesie nauczania należy zwrócić szczególną uwagę na rozwijanie umiejętności myślenia algorytmicznego oraz stosowania algorytmów w sytuacjach praktycznych. Należy dobrze wyćwiczyć różnorodne struktury danych, tak, aby korzystanie z nich nie stanowiło już dla maturzystów problemu.
4. Warto zwracać uczniom uwagę na dokładne czytanie poleceń oraz uwzględnienie wszystkich warunków zadania przy tworzeniu planu rozwiązania, ponieważ pominięcie choćby jednego

warunku w zadaniu daje błędny wynik lub generuje błędne rozwiązanie całego zadania. Zdający powinni wykonywać zadania starannie, zgodnie z poleceniem. Rolą nauczyciela jest również zapoznanie uczniów z różnymi prawidłowymi sposobami zapisu algorytmów. W części teoretycznej zdający często tracą dużo czasu na pisanie całego dużego programu, zamiast wymaganego w zadaniu algorytmu.

5. Wskazane jest, by przez cały okres edukacji uwrażliwiać uczniów na stosowanie języka informatycznego podczas rozwiązywania i opisu problemu.
6. Zauważalnie poprawiły się umiejętności zdających w zakresie wykorzystania dostępnych narzędzi informatycznych, tj. arkusza kalkulacyjnego czy aplikacji do obsługi baz danych. Dobrze byłoby, gdyby uczeń od początku pracował z programami, których będzie używał podczas egzaminu (warto w tym miejscu przypomnieć, że od roku szkolnego 2017/2018 na maturze z informatyki nie będzie już możliwości wyboru języka programowania Pascal).