

**OKRĘGOWA KOMISJA EGZAMINACYJNA
W GDAŃSKU**

**Sprawozdanie z egzaminu maturalnego
z informatyki
przeprowadzonego w województwie
pomorskim
w 2019 roku**

Opracowanie

Iwona Arcimowicz (Centralna Komisja Egzaminacyjna)

Izabela Szafrńska (Okręgowa Komisja Egzaminacyjna w Poznaniu)

Redakcja

dr Wioletta Kozak (Centralna Komisja Egzaminacyjna)

Opracowanie techniczne

Joanna Dobkowska (Centralna Komisja Egzaminacyjna)

Współpraca

Beata Dobrosielska (Centralna Komisja Egzaminacyjna)

Agata Wiśniewska (Centralna Komisja Egzaminacyjna)

Pracownie ds. Analiz Wyników Egzaminacyjnych okręgowych komisji egzaminacyjnych

Centralna Komisja Egzaminacyjna

ul. Józefa Lewartowskiego 6, 00-190 Warszawa

tel. 022 536 65 00, fax 022 536 65 04

e-mail: sekretariat@cke.gov.pl

www.cke.gov.pl

Okręgowa Komisja Egzaminacyjna w Gdańsku

ul. Na Stoku 49, 00-874 Gdańsk

tel. 58 320 55 61, fax 58 520 55 90

e-mail: komisja@oke.gda.pl

www.oke.gda.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 dla arkusza kalkulacyjnego, które można było również rozwiązać, pisząc program komputerowy.

Arkusz I zestawu egzaminacyjnego zawierał 3 zadania (8 poleceń), za które zdający mógł uzyskać maksymalnie 15 punktów. Arkusz II zawierał 3 zadania (12 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

W maju 2019 r. do pisemnego egzaminu maturalnego z informatyki w nowej formule, przystąpili absolwenci liceów ogólnokształcących oraz absolwenci techników.

Tabela 1. Zdający rozwiązujący zadania w arkuszu standardowym*

Liczba zdających		396
Zdający rozwiązujący zadania w arkuszu standardowym	z liceów ogólnokształcących	195
	z techników	201
	ze szkół na wsi	4
	ze szkół w miastach do 20 tys. mieszkańców	40
	ze szkół w miastach od 20 tys. do 100 tys. mieszkańców	156
	ze szkół w miastach powyżej 100 tys. mieszkańców	196
	ze szkół publicznych	378
	ze szkół niepublicznych	18
	kobiety	28
	mężczyźni	368
	bez dysleksji rozwojowej	318
	z dysleksją rozwojową	78

* 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	6
	słabowidzący	0
	niewidomi	0
	słabosłyszący	1
	niestyszący	0
	ogółem	7

3. Przebieg egzaminu

Tabela 3. Informacje dotyczące przebiegu egzaminu

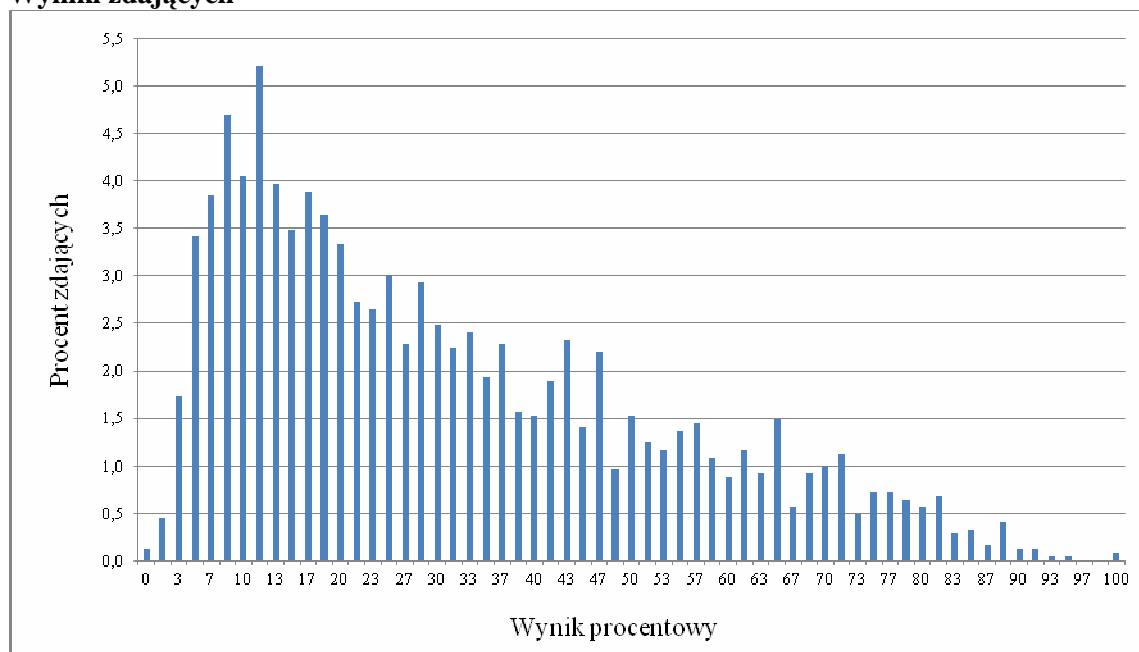
Termin egzaminu		13 maja 2019	
Czas trwania egzaminu		210 minut	
Liczba szkół		71	
Liczba zespołów egzaminatorów		1	
Liczba egzaminatorów		8	
Liczba obserwatorów ¹ (§ 8 ust. 1)		3	
Liczba unieważnień ²	art. 44zzv pkt 1	stwierdzenia niesamodzielnego rozwiązywania zadań przez zdającego	0
	art. 44zzv pkt 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 maturalnego	2
	art. 44zzy ust. 10	niemożności ustalenia wyniku (np. zaginięcie karty odpowiedzi)	0
Liczba wglądów ² (art. 44zzz)		6	
Liczba prac, w których nie podjęto rozwiązania zadań		0	

¹ Na podstawie rozporządzenia Ministra Edukacji Narodowej z dnia 21 grudnia 2016 r. w sprawie szczegółowych warunków i sposobu przeprowadzania egzaminu gimnazjalnego i egzaminu maturalnego (Dz.U. z 2016 r., poz. 2223, ze zm.).

² Na podstawie ustawy z dnia 7 września 1991 r. o systemie oświaty (tekst jedn. Dz.U. z 2019 r., poz. 1481).

4. Podstawowe dane statystyczne

Wyniki zdających



Wykres 1. Rozkład wyników zdających

Tabela 4. Wyniki zdających – parametry statystyczne*

Zdający	Liczba zdających	Minimum (%)	Maksimum (%)	Mediana (%)	Modalna (%)	Średnia (%)	Odchylenie standardowe (%)
ogółem	396	0	100	40	22	43	24
w tym:							
z liceów ogólnokształcących	195	0	100	52	42	53	24
z techników	201	0	98	30	22	34	19

* 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: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 5) posługuje się podstawowymi technikami algorytmicznymi; 11) opisuje podstawowe algorytmy i stosuje: [...] b) algorytmy wyszukiwania i porządkowania (sortowania) [...] 16) opisuje własności algorytmów na podstawie ich analizy; 17) ocenia zgodność algorytmu ze specyfikacją problemu; 18) oblicza liczbę operacji wykonywanych przez algorytm; 20) bada efektywność komputerowych rozwiązań problemów.	55
1.2.	III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji [...], stosowanie podejścia algorytmicznego. Zdający: 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 5) posługuje się podstawowymi technikami algorytmicznymi; 16) opisuje własności algorytmów na podstawie ich analizy; 17) ocenia zgodność algorytmu ze specyfikacją problemu; 18) oblicza liczbę operacji wykonywanych przez algorytm; 20) bada efektywność komputerowych rozwiązań problemów.	69
2.1.	III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego.	5. Rozwiązywanie problemów i podejmowanie decyzji [...], stosowanie podejścia algorytmicznego. Zdający: 5) posługuje się podstawowymi technikami algorytmicznymi; 9) stosuje rekurencję w prostych sytuacjach problemowych; 11) opisuje podstawowe algorytmy i stosuje: [...] d) algorytmy na tekstach [...], 16) opisuje własności algorytmów na podstawie ich analizy; 17) ocenia zgodność algorytmu ze specyfikacją problemu; 18) oblicza liczbę operacji wykonywanych przez algorytm.	71
2.2.	III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem	5. Rozwiązywanie problemów i podejmowanie decyzji [...], stosowanie podejścia algorytmicznego. Zdający: 5) posługuje się podstawowymi technikami	46

	<p>podejścia algorytmicznego.</p>	<p>algorytmicznymi; 9) stosuje rekurencję w prostych sytuacjach problemowych; 11) opisuje podstawowe algorytmy i stosuje: [...] d) algorytmy na tekstach [...], 16) opisuje własności algorytmów na podstawie ich analizy; 17) ocenia zgodność algorytmu ze specyfikacją problemu; 18) oblicza liczbę operacji wykonywanych przez algorytm.</p>	
2.3.	<p>III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego.</p>	<p>5. Rozwiązywanie problemów i podejmowanie decyzji [...], stosowanie podejścia algorytmicznego. Zdający: 5) posługuje się podstawowymi technikami algorytmicznymi; 9) stosuje rekurencję w prostych sytuacjach problemowych; 16) opisuje własności algorytmów na podstawie ich analizy; 17) ocenia zgodność algorytmu ze specyfikacją problemu; 18) oblicza liczbę operacji wykonywanych przez algorytm.</p>	31
3.1.	<p>II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.</p>	<p>2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Zdający: 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL).</p>	46
3.2.	<p>I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.</p>	<p>1. Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej. Zdający: 1) przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków.</p>	50
3.3.	<p>I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.</p>	<p>3. Bezpieczne posługiwanie się komputerem, jego oprogramowaniem i korzystanie z sieci komputerowej. Zdający: 3) [...] 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.</p>	32
4.1.	<p>III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia</p>	<p>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</p>	47

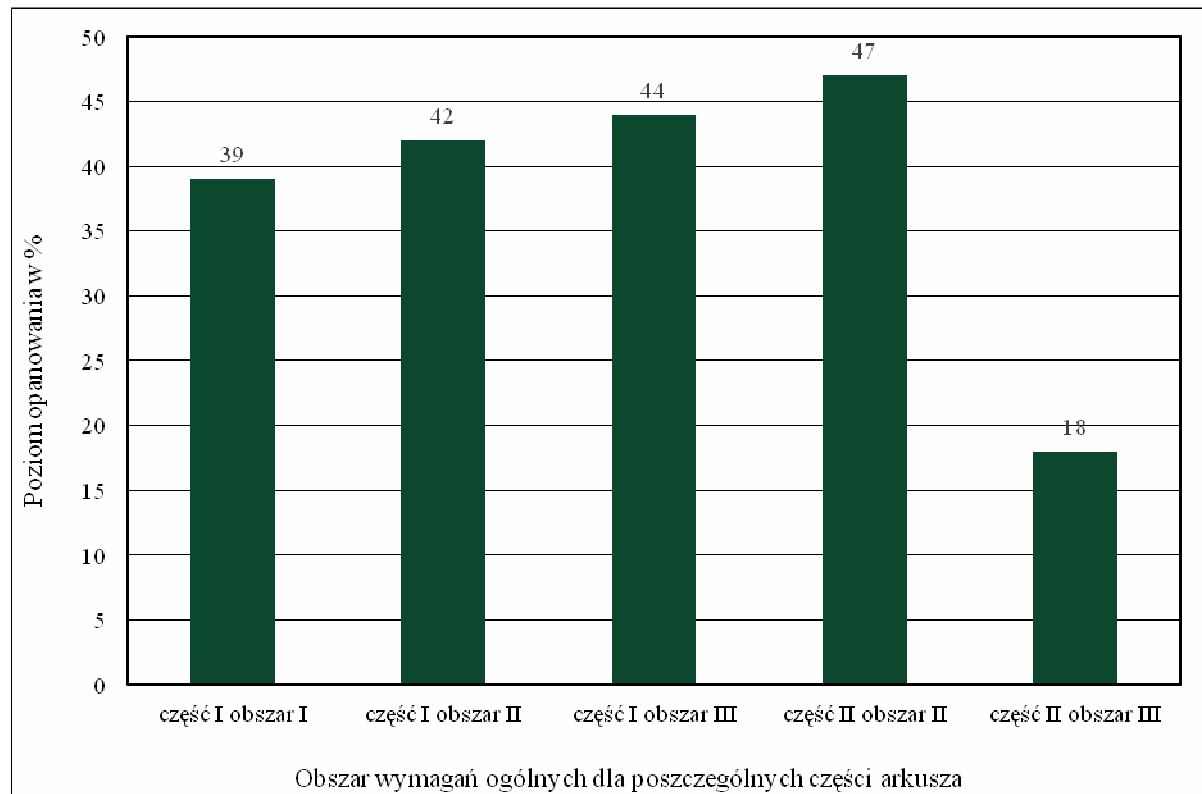
	algorytmicznego.	<p>problemu;</p> <p>3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera;</p> <p>4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji;</p> <p>5) posługuje się podstawowymi technikami algorytmicznymi;</p> <p>6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;</p> <p>7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;</p> <p>11) opisuje podstawowe algorytmy i stosuje:</p> <p style="padding-left: 20px;">a) algorytmy na liczbach całkowitych [...],</p> <p style="padding-left: 20px;">c) algorytmy numeryczne [...],</p> <p>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;</p> <p>26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.</p>	
4.2.	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.	<p>5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.</p> <p>Zdający:</p> <p>1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin;</p> <p>2) stosuje podejście algorytmiczne do rozwiązywania problemu;</p> <p>3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera;</p> <p>4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji;</p> <p>5) posługuje się podstawowymi technikami algorytmicznymi;</p> <p>6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;</p> <p>7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;</p> <p>11) opisuje podstawowe algorytmy i stosuje:</p> <p style="padding-left: 20px;">a) algorytmy na liczbach całkowitych [...],</p> <p style="padding-left: 20px;">c) algorytmy numeryczne [...],</p> <p>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;</p> <p>26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.</p>	35
4.3.	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera,	<p>5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.</p> <p>Zdający:</p> <p>1) analizuje, modeluje i rozwiązuje sytuacje problemowe</p>	12

	z zastosowaniem podejścia algorytmicznego.	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: a) algorytmy na liczbach całkowitych [...], c) algorytmy numeryczne [...], 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.	
5.1.	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; 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.	69
5.2.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji	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:	60

	<p>multimedialnych.</p> <p>III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.</p>	<p>1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin;</p> <p>2) stosuje podejście algorytmiczne do rozwiązywania problemu;</p> <p>3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera;</p> <p>6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;</p> <p>7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania.</p>	
5.3.	<p>II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.</p> <p>III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.</p>	<p>4. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów.</p> <p>Zdający:</p> <p>4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów.</p> <p>5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.</p> <p>Zdający:</p> <p>1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin;</p> <p>2) stosuje podejście algorytmiczne do rozwiązywania problemu;</p> <p>3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera;</p> <p>6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;</p> <p>7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania.</p>	58
5.4.	<p>II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.</p> <p>III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.</p>	<p>4. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów.</p> <p>Zdający:</p> <p>4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów.</p> <p>5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.</p> <p>Zdający:</p> <p>1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin;</p> <p>2) stosuje podejście algorytmiczne do rozwiązywania problemu;</p> <p>3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera;</p> <p>6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;</p> <p>7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania</p>	11

		problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania.	
6.1.	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.	79
6.2.	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.	58
6.3.	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.	32
6.4.	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.	53
6.5.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera:	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	29

	rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.	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.	
--	---	---	--



Wykres 2. Poziom wykonania zadań w obszarze wymagań ogólnych

Komentarz

W maju 2019 roku absolwenci po raz piąty przystąpili do egzaminu z informatyki na poziomie rozszerzonym w nowej formule. Tak jak w poprzednich latach egzamin podzielony był na dwa arkusze – teoretyczny (bez użycia komputera) oraz praktyczny (z komputerem). Maksymalna liczba punktów wynosiła 50, w tym 15 za część I oraz 35 za część II.

Zadania 1. i 2. z arkusza I były typowymi zadaniami sprawdzającymi umiejętność myślenia algorytmicznego oraz zapisu algorytmu w wybranej przez zdającego notacji. Zadanie 3.1. sprawdzało umiejętność posługiwania się relacyjnymi bazami danych, a w szczególności strukturalnym językiem zapytań SQL. Zadanie 3.2. wymagało wiedzy na temat sposobów reprezentacji liczb w komputerze w tym umiejętności zapisywania liczb w systemach o różnej podstawie. Zadanie 3.3. związane było z wiedzą na temat administrowania siecią komputerową w architekturze klient-serwer, korzystania z usług w sieci komputerowej, lokalnej i globalnej, związanych z dostępem do informacji, wymianą informacji i komunikacją. Rozwiązanie problemu postawionego w zadaniu 4. w arkuszu II wymagało zastosowania podejścia algorytmicznego z wykorzystaniem dowolnego języka programowania. Zadanie 5. wymagało opracowania za pomocą komputera danych liczbowych, z wykorzystaniem analizy statystycznej danych oraz podejścia algorytmicznego do podanego problemu. Zadaniem sprawdzającym umiejętność posługiwania się relacyjnymi bazami danych oraz językiem SQL było zadanie 6. Zadanie to można było rozwiązać, korzystając z narzędzi bazodanowych – programu bazodanowego (np. MS Access, Apache OpenOffice Base, LibreOffice Base), strukturalnego języka zapytań SQL lub dowolnego języka programowania.

1. Analiza jakościowa zadań

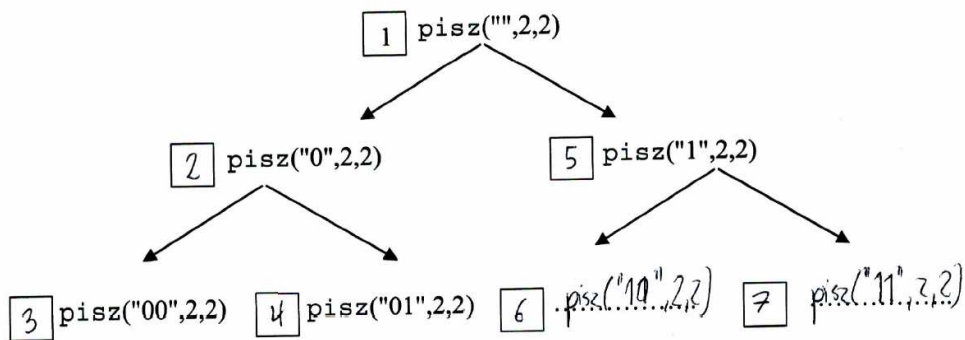
W arkuszu I najtrudniejszymi okazały się zadania: 2.3. (poziom wykonania – 31%) i 3.3. (poziom wykonania – 32%). Do zadań trudnych (poziom wykonania z zakresu od 20% do 49%) należały także zadania: 2.2. (poziom wykonania – 45%), 3.1. (poziom wykonania – 46%). Reprezentują one wszystkie trzy obszary wymagań programowych występujące w arkuszu tj.:

- obszar I – bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych – zadanie 3.3.,
- obszar II – wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł – zadania 3.1.,
- obszar III – rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego – zadania , 2.2., 2.3.

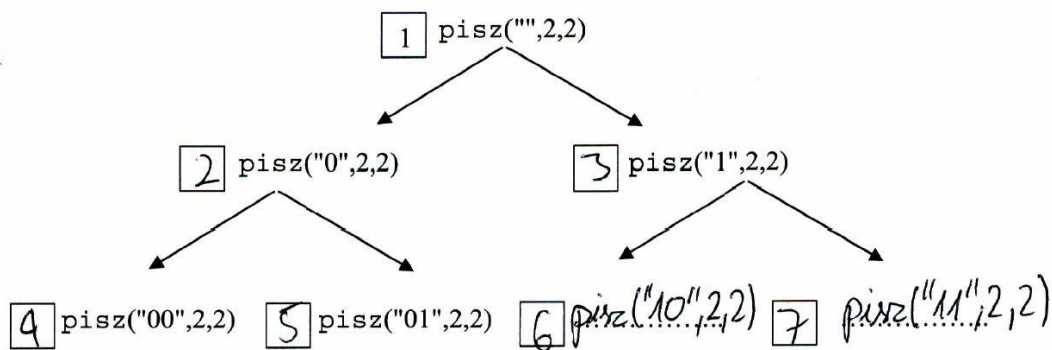
W zadaniu 2.3. należało podać łączną liczbę wywołań funkcji `pisz` w wyniku wywołania `pisz("",n,k)`, co wymagało przedtem dokonania analizy działania funkcji zapisanej w pseudojęzyku programowania i umiejętności obliczania liczby operacji wykonywanych przez algorytm. Największą trudnością był fakt, że zadanie dotyczyło algorytmu rekurencyjnego.

Tego samego algorytmu dotyczyło zadanie 2.1. (poziom wykonania – 71%) oraz zadanie 2.2. (poziom wykonania – 46%). Zadanie 2.1., w którym należało uzupełnić drzewo wywołań funkcji `pisz`, okazało się być dla zdających dużo łatwiejszym niż 2.3. oraz 2.2. Zdający dobrze poradzili sobie z wypełnieniem ostatnich dwóch węzłów drzewa, czyli zapisem wywołania funkcji z odpowiednimi parametrami. Mieli oni możliwość przeanalizowania wpisów w podanych węzłach i wypełnienia kolejnych na zasadzie analogii. Trudniejsze okazało się ustalenie kolejności wywołań funkcji (Przykład 1. Prawidłowa odpowiedź, Przykład 2. – błędna kolejność wywołań funkcji `pisz`). W zadaniu 2.2. należało zapisać teksty wypisywane po wywołaniu funkcji `pisz` oraz podać ilość wywołań tej funkcji przy zadanych parametrach, co wymagało już dokładniejszej analizy działania algorytmu.

Przykład 1.



Przykład 2.



Do zadań trudnych zaliczało się, jak w poprzednich latach, zadanie wymagające zapisania algorytmu, czyli zadanie 1.1. W zadaniu tym należało napisać algorytm znajdujący pierwszą liczbę zapisaną przez „Jasia”, czyli pierwszą liczbę parzystą w ciągu liczb, w którym najpierw zapisano wszystkie liczby nieparzyste (liczby „Małgosi”), a następnie liczby parzyste. W zadaniu maksymalną liczbę punktów można było uzyskać zapisując algorytm o złożoności lepszej niż liniowa. Duża część zdających zapisała prawidłowy algorytm o złożoności liniowej, który był stosunkowo prosty do zapisania – wystarczyło przeglądać liczby po kolei do trafienia na parzystą (Przykład 3.). Zdający, którzy zdecydowali się na algorytm o złożoności lepszej niż liniowa najczęściej zapisywali algorytm wyszukiwania binarnego. Cieszy to, że większość zdających poprawnie oceniła złożoność zapisanego przez siebie algorytmu, którą należało podać w zadaniu 1.2 (poziom wykonania – 61%).

Przykład 3.

```
for (int i = 1; i <= n; i++) {  
    if (A[i] % 2 == 0) {  
        w = A[i];  
        break;  
    }  
}
```


Wśród zadań trudnych znalazły się także wszystkie trzy zadania testowe z części I arkusza (3.1 poziom wykonania – 46%, 3.2 poziom wykonania – 48% oraz 3.3 poziom wykonania – 32%), podobnie jak w latach poprzednich. W tym przypadku dużą rolę odgrywa specyfika tych zadań – w każdym z nich trzeba poprawnie ocenić prawdziwość wszystkich czterech zdań, aby otrzymać jeden punkt. Warto zwrócić uwagę, że zdający lepiej poradzili sobie z zadaniami testowymi niż w roku poprzednim. Da się zauważyć prawidłowość, że łatwiejsze są dla zdających zadania testowe „konkretne” czyli np. z przykładami zapytaniai SQL, czy też zapisami liczb w różnych systemach liczbowych (takie jak tegoroczne zadania 3.1, 3.2) niż pytania wymagające wiedzy teoretycznej (zadanie 3.3.).

W arkuszu II najtrudniejsze okazały się zadania 4.3(poziom wykonania-12%) i 5.4. (poziom wykonania – 11%). Są to zadania z zakresu rozwiązywania problemów i podejmowania decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.

W zadaniu 4.3. należało napisać program znajdujący w pliku z liczbami najdłuższy ciąg liczb występujących kolejno po sobie i takich, że największy wspólny dzielnik ich wszystkich jest większy niż 1. Wymagało to umiejętności napisania realizacji algorytmu (omawianego w szkole) liczącego największy wspólny dzielnik i zastosowania go przy szukaniu ciągu liczb spełniającego zadane warunki. Rozwiązania tego zadania zostaną omówione w dalszej części komentarza.

Rozwiązanie zadania 5.4. wymagało zaplanowania i przeprowadzenia symulacji zjawisk przedstawionych w treści zadania z wykorzystaniem funkcji i formuł arkusza kalkulacyjnego. Można je było także rozwiązać pisząc program, ale bardzo niewielu zdających się na to zdecydowało. Zadania wymagające wykonania symulacji w arkuszu kalkulacyjnym tradycyjnie należą do najtrudniejszych zadań na maturze z informatyki. W omawianym zadaniu należało wykonać symulację rozwijania się chmur danej wielkości i rodzaju (kategorii) na fikcyjnej planecie według podanej teorii, wykorzystując dane na temat temperatury i wielkości opadów. Wykonując obliczenia mające dawać w wyniku wielkości chmur, należało uwzględnić trzy czynniki: czy poprzedni dzień był pochmurny, zachowanie chmur odpowiednich wielkości według teorii i wielkość opadu poprzedniego dnia. Zapisanie warunku dającego w wyniku odpowiednią przewidywaną wielkość chmur nie było trywialne. Łatwiejsze było wykonanie drugiej części symulacji przewidującej kategorię chmur, ponieważ wystarczyło wykorzystać informację o temperaturze i o tym, czy poprzedni dzień był zachmurzony. Po przeprowadzeniu symulacji należało wykonać zestawienie liczby dni z poszczególnymi wielkościami chmur oraz obliczyć liczbę dni, w których przewidywane wielkości chmur i kategorie pokrywały się z obserwacjami.

Najtrudniejszym elementem zadania 5.4. była dokładna analiza zadanych warunków i ich zapisanie. Poniżej przykład poprawnego zapisu warunku dla symulacji wielkości chmur:

Przykład 4.

F7		=JEŻELI(ORAZ(F6=5;C6>=20);0;JEŻELI(F6=5;5;JEŻELI(F6=0;1;JEŻELI(ORAZ(F4=F5;F5=F6);F6+1;F6))))											
	A	B	C	D	E	F	G	H	I	J	K	L	
1	Dzien	Temperatura	Opad	Kategoria_chmur	Wielkosc_chmur	wielkosc	kategoria	b	c				
2	1	19	0		0	0	0	1	1				
3	2	22	1	C		1	1 C	1	1		a	296	
4	3	23,6	4	C		1	1 C	1	1		b	286	
5	4	23,6	4	C		1	1 C	1	1				
6	5	22,3	10	C		2	2 C	1	1				
7	6	20,4	8	C		2	2 C	1	1				
8	7	18,9	10	C		2	2 C	1	1				
9	8	18,5	11	C		3	3 C	1	1				
10	9	19,5	14	C		3	3 C	1	1				

oraz kategorii chmur:

Przykład 5.

G7		=JEŻELI(D7=0;0;JEŻELI(F6=0;JEŻELI(B7<10;"S";"C");G6))						
	A	B	C	D	E	F	G	H
1	Dzien	Temperatura	Opad	Kategoria_chmur	Wielkosc_chmur	wielkosc	kategoria	b
2	1	19	0	0	0	0	0	1
3	2	22	1 C		1	1	1 C	1
4	3	23,6	4 C		1	1	1 C	1
5	4	23,6	4 C		1	1	1 C	1
6	5	22,3	10 C		2	2	2 C	1
7	6	20,4	8 C		2	2	2 C	1
8	7	18,9	10 C		2	2	2 C	1

Przykład nieprawidłowego zapisu warunku dla przewidywanych wielkości chmur:

Przykład 6.

		=JEŻELI(ORAZ(E6=5;C6>=20);0;JEŻELI(E6=0;1;JEŻELI(ORAZ(E4=E5;E5=E6);E6+1;E6)))				
	C	D	E	F		
	Opad	Kategoria_chmur	Wielkosc_chmur	teoria wielkosc	teoria	
19	0		0	0	0 C	
22	1	1 C		1	1 Nielic:	
3,6	4	4 C		1	1 Nielic:	
3,6	4	4 C		1	1 Nielic:	
2,3	10	10 C		2	2 Nielic:	
0,4	8	8 C		2	2 =JEŻELI(ORAZ(E6=5;C6>=20) Nielic:	
8,9	10	10 C		2	2 Nielic:	

W tym przypadku zdający popełnił kilka błędów. Zapisując warunek, odwołał się do danych dotyczących wielkości chmur z obserwacji, a przecież właśnie wykonywał symulację ich powstawania (powinien odwoływać się do kolumny F a nie E). Tak zapisany warunek spowodował po 300 wierszu przywidywanie wyłącznie chmur o wielkości 1 (ponieważ po 300 dniu zgodnie z treścią zadania, występował brak danych o wielkości chmur). Ten błąd spowodował również uzyskanie danych dotyczących utrzymywania się określonej wielkości chmur, sprzecznych z „teorią profesora” podaną w treści zadania - ta sama wielkość chmur, mniejsza niż 5, utrzymuje się dłużej niż trzy dni (przykład 7. fragment pracy tego samego zdającego).

Przykład 7.

✓ fx		=JEŻELI(ORAZ(E62=5;C62>=20);0;JEŻELI(E62=0;1;JEŻELI(ORAZ(E60=E61;E61=E62);E62+1;E62)))			
C	D	E	F	G	
16 C		3	3	Nielicz	
14 C		3	2	Nielicz	
14 C		3	4	Nielicz	
6 C		4	4	Nielicz	
21 C		4	4	Nielicz	
21 C		5	4	Nielicz	
0	0	0	0	C	
4 C		1	1	Nielicz	

Dodatkowo źle skonstruowany zapis spowodował, że w symulacji pojawiły się chmury o wielkości 6 podczas gdy maksymalna wielkość chmur w treści zadania to 5.

Dużym problemem dla zdających (poza samą analizą treści i warunków zadania) był w tym zadaniu zapis zagnieżdżonych instrukcji JEŻELI i instrukcji ORAZ. Część zdających błędnie interpretowała także informację o kategorii powstających chmur według przedstawianej w zadaniu teorii np. pojawiały się w symulacjach zdających kategorie chmur C lub S wpisane w dniu bezchmurnym (gdzie powinno być 0 zgodnie z opisem w treści zadania). Wielu zdających nie podjęło się rozwiązania tego zadania.

Do zadań trudnych (poziom wykonania z zakresu od 20% do 49%) w arkuszu II należały zadania: 6.3. (poziom wykonania – 32%), 6.5. (poziom wykonania – 29%), 4.2. (poziom wykonania – 35%), 4.1. (poziom wykonania – 46%). Reprezentują one II i III obszar wymagań programowych:

- obszar II – wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł – zadania 6.3., 6.5. ,
- obszar III – rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego – zadania 4.1. oraz 4.2.

Aby rozwiązać zadanie 6.3. należało w bazie danych znaleźć i podać w kolejności alfabetycznej nazwy marek perfum, które nie zawierały składnika mającego w nazwie słowo „paczula”. Było to typowe zadanie bazodanowe polegające na znajdowaniu niepasujących danych. Wymagało ono wyszukania za pomocą odpowiedniej kwerendy najpierw marek perfum zawierających składniki ze słowem „paczula” a następnie wyszukania tych marek, które nie pasowały do poprzedniego wyniku. Tego typu zadania powtarzają się w kolejnych latach maturalnych, ale nadal sprawiają trudności zdającym. Część zdających podawała w wyniku listę marek, które zawierały w składnikach nazwę „paczula” zamiast tych, które jej nie zawierały.

Zadanie 6.5. wymagało znalezienia w bazie danych marek, których wszystkie perfumy należą tylko do jednej rodziny zapachów. Jako odpowiedź należało podać nazwy marek oraz nazwy rodzin zapachów tych marek. Częstym błędem było podawanie przez zdających listy tylko trzech marek zamiast pięciu, co wynikało z nieprawidłowego liczenia liczby rodzin zapachów danej marki (np. dwukrotnego policzenia tej samej rodziny z powodu istnienia dwóch rodzajów perfum z tej rodziny w danej marce).

Tradycyjnie do zadań trudnych zaliczały się także zadania wymagające napisania programu w języku programowania. W zadaniu 4.1., o najwyższym poziomie wykonania spośród zadań programistycznych – 45%, należało zliczyć, ile z podanych w pliku z danymi liczb jest potęgami liczby 3. W zadaniu 4.2. (poziom wykonania – 27%) należało w tym samym pliku znaleźć i wypisać w kolejności ich występowania liczby, które są równe sumie silni swoich cyfr. Do obydwu zadań dołączone zostały pliki z przykładowymi danymi i odpowiedziami służące do testowania programów pisanych przez zdających.

Do zadań umiarkowanie trudnych (poziom wykonania z zakresu od 50% do 69%) w arkuszu II należały zadania: 6.2. (poziom wykonania – 58%), 6.4. (poziom wykonania – 53%), 5.3. (poziom wykonania – 58%) oraz 5.1. (poziom wykonania – 69%). Zadania 6.2. i 6.4 to zadania bazodanowe, natomiast zadanie 5.3. oraz 5.1. są zadaniami z zakresu przetwarzania informacji w arkuszu kalkulacyjnym.

Za łatwe można uznać tylko jedno zadanie z poziomem wykonania 77% – jest to zadanie 6.1., czyli pierwsze z zadań bazodanowych. W zadaniu tym należało wyszukać w bazie danych i podać listę nazw perfum zawierających składnik „absolut jaśminu”.

2. Problem „pod lupą”

Zadanie 4.3. – zadanie programistyczne.

W tym roku najwięcej problemów sprawiły zdającym dwa zadania: zadanie 5.4. wymagające wykonania symulacji w arkuszu kalkulacyjnym oraz ostatnie z zadań programistycznych – zadanie 4.3. Zadanie 5.4. zostało omówione wcześniej.

Spośród zadań wymagających napisania programu jak co roku wszystkie były trudne dla zdających, jednak zadanie 4.3. było także najtrudniejszym z zadań tegorocznego arkusza. Było ono także bardzo często pomijane przez zdających.

W zadaniu tym należało znaleźć najdłuższy ciąg kolejno występujących liczb zapisanych w pliku liczby.txt taki, że największy wspólny dzielnik wszystkich liczb w ciągu jest większy niż 1. Jako odpowiedź należało podać pierwszą liczbę ciągu, długość ciągu oraz największy wspólny dzielnik liczb w ciągu. W pliku z danymi był tylko jeden ciąg spełniający warunki zadania o największej długości.

Dodatkowo dołączona została informacja, że największy wspólny dzielnik kilku liczb można obliczyć wykorzystując największy wspólny dzielnik dwóch liczb: $NWD(a,b,c) = NWD(NWD(a,b),c)$. Pojęcie największego wspólnego dzielnika dwóch liczb powinno być znane wszystkim zdającym z lekcji informatyki (wraz z realizacją algorytmu wyliczającego NWD) jak i z lekcji matematyki.

W arkuszu podano także przykłady:

Dla liczb 3, 7, **4**, **6**, **10**, **2**, 5 odpowiedzią jest 4 (pierwsza liczba ciągu), 4 (długość ciągu) i 2 (największy wspólny dzielnik), natomiast dla liczb 5, **70**, **28**, **42**, **98**, 1 odpowiedzią jest 70 (pierwsza liczba ciągu), 4 (długość ciągu) i 14 (największy wspólny dzielnik).

Zdający mieli do dyspozycji, tak jak w poprzednich latach, plik z przykładowymi liczbami i odpowiedzi dla danych z tego pliku, aby mogli sprawdzić poprawność działania swojego programu.

Przeanalizujmy najpierw przykłady dołączone do zadania. Mamy dane liczby 3, 7, 4, 6, 10, 2, 5. Bierzemy dwie pierwsze liczby z tego zestawu i obliczamy NWD (największy wspólny dzielnik). Otrzymujemy $NWD(3,7)=1$, co oznacza, że 3 nie jest początkiem ciągu spełniającego warunki zadania. $NWD(7,4)$ także jest równe 1, $NWD(4,6)=2$, a więc więcej niż 1, liczymy więc $NWD(2,10)=2$, dalej $NWD(2,2)=2$ i $NWD(2,5)=1$. Otrzymaliśmy pierwszy i jedyny ciąg z największym wspólnym dzielnikiem wszystkich liczb w ciągu większym niż 1 to jest: 4,6,10,2. Pierwszą liczbą ciągu jest 4, długość ciągu 4, dzielnik 2 – zgodnie z opisem przykładu.

Podobnie mając liczby 5,70,28,42,98,1 liczymy: $NWD(5,70)=5$, $NWD(5,28)=1$, więc mamy ciąg 5,70, ale 28 do niego nie należy. Szukamy następnego ciągu. Bierzemy $NWD(70,28)=14$, następnie $NWD(14,42)=14$, $NWD(14,98)=14$, $NWD(14,1)=1$. Otrzymaliśmy drugi ciąg 70,28,42,98 w którym pierwszą liczbą jest 70, długość to 4, a dzielnik 14, jest dłuższy niż poprzedni i jest wynikiem dla podanego przykładu.

Analiza przykładów jest istotna i pomocna, ponieważ pozwala właściwie zrozumieć polecenie. W drugim przykładzie widać, że początek ciągu, którego szukamy może być jednocześnie końcem ciągu poprzedniego. Ponadto po wykonaniu kolejnych obliczeń wiemy już co powinien robić nasz

algorytm.

Przykładowe rozwiązania poprawne:

Przykład 8.

```
int mxl=0,first=0,dzielnik;
for (int i=0; licz.size()>i; i++){
    int nw=licz[i],tmpfirst=licz[i],l=1;
    for (int j=i+1; licz.size()>j; j++){
        nw=NWD(nw,licz[j]);
        if (nw!=1){
            l++;
            if (l>mxl){
                first=tmpfirst;
                dzielnik=nw;
                mxl=l;
            }
        }
        else break;
    }
}
```

Przykład 8. jest fragmentem programu zdającego, w którym zapisane zostało wyszukiwanie najdłuższego ciągu spełniającego warunek zadania. Zdający zapisał liczby wczytane z pliku w wektorze nazwanym przez siebie „licz”. Kolejne zmienne użyte przez zdającego oznaczają: mxl – maksymalną długość szukanego ciągu, first – pierwszą liczbę w szukanym ciągu, dzielnik – największy wspólny dzielnik szukanego ciągu, nw – największy wspólny dzielnik aktualnie sprawdzanego ciągu liczb, tmpfirst – pierwsza liczba aktualnie sprawdzanego ciągu liczb oraz l – długość badanego ciągu. Jak widać w zapisie zdający zastosował dwie pętle.

W zewnętrznej pętli pobierane są kolejne wartości liczb z zestawu danych, za każdym razem przypisywane jako tymczasowe pierwsze elementy sprawdzanego ciągu liczb, następnie w drugiej pętli zdający wylicza kolejne wartości największego wspólnego dzielnika ciągu liczb zaczynającego się od wyznaczonej pierwszej liczby w zewnętrznej pętli a kończącego na liczbie, która jako ostatnia daje NWD różne od 1. Za każdym razem po obliczeniu NWD następuje sprawdzenie czy aktualna długość ciągu nie jest większa niż zapamiętana maksymalna (czyli szukany jest najdłuższy ciąg).

Zdający otrzymał poprawny wynik, czyli pierwszą liczbę – 31968, długość ciągu – 150 i dzielnik – 74.

Przykład 9.

```
struct ciag
{
    int pierwsza, dl, nwd;

    ciag(int p)
    {
        pierwsza=p;
        dl=1;
        nwd=1;
    }

    void kopiuj(ciag c)
    {
        pierwsza=c.pierwsza;
        dl=c.dl;
        nwd=c.nwd;
    }
};
```


Przykład 10.

```

wyjscie << "4.3" << endl;
ciag najdl(0);

for (int i=0; i<n; i++)
{
    ciag obecny(i);
    obecny.nwd = liczby[i];
    int nwd2 = obecny.nwd;
    for (int j=i+1; j<n; j++)
    {
        obecny.nwd = nwd(obecny.nwd, liczby[j]);
        if (obecny.nwd==1)
            break;
        else
        {
            obecny.dl++;
            nwd2 = obecny.nwd;
        }
    }
    if (obecny.dl>najdl.dl)
    {
        obecny.nwd = nwd2;
        najdl.kopiuj(obecny);
    }
}

```

Przykłady 9. i 10. są fragmentami rozwiązania kolejnego zadającego. W przykładzie 9. zadeklarowano typ strukturalny, który został wykorzystany do przechowywania danych o ciągach liczb najdłuższym (którego szukamy) i „obecnym” czyli analizowanym podczas wyszukiwania. Sam sposób wyszukiwania ciągu liczb spełniającego warunki zadania jest podobny do tego w poprzednim przykładzie, a więc mamy pętlę zewnętrzną i wewnętrzną. Jediną różnicą jest to, że zadający sprawdza czy znaleziony ciąg jest dłuższy od ostatnio zapamiętanego najdłuższego dopiero po znalezieniu końca aktualnego („obecnego”) ciągu liczb.

Przykład 11.

```

vector<int> zad3()
{
    vector<int> result;

    int m_cnt=0;
    int pierwsza_liczba=0;
    int r_nwd=1;
    for(int i=0;i<dane.size()-1;++i)
    {
        int cnt=2;
        int l_nwd=nwd(dane[i],dane[i+1]);
        if(l_nwd>1)
        {
            for(int j=i+2;j<dane.size();++j)
            {
                int n_nwd=nwd(l_nwd,dane[j]);
                if(n_nwd>1) //jesli nowe nwd jest wieksze od zera zwiekszam dlugosc ciagu
                {
                    l_nwd=n_nwd;
                    cnt++;
                }
                else
                {
                    if(cnt>m_cnt) //jestli nowy ciag jest najdluzszym ciagiem
                    {
                        m_cnt=cnt;
                        pierwsza_liczba=dane[i];
                        r_nwd=l_nwd; //najwieszke NWD dla tego ciagu
                    }

                    break;
                }
            }
        }
    }
    result.push_back(pierwsza_liczba);
    result.push_back(m_cnt);
    result.push_back(r_nwd);
    return result;
}

```

Przykład 11. reprezentuje podobne podejście, jednak w tym przypadku zdający wykorzystał klasę vector do przechowywania informacji o znalezionym ciągu.

Wśród błędnych rozwiązań często powtarzała się sytuacja, która nie powinna się zdarzyć, gdyby wcześniej została przeprowadzona dokładna analiza problemu. Najdłuższy szukany ciąg, który powinien być podany w prawidłowej odpowiedzi zaczynał się od liczby, która jednocześnie była ostatnią w poprzednim ciągu liczb – tak jak w drugim przykładzie podanym w treści zadania. Zdający, którego fragment rozwiązania pokazany został w Przykładzie 12. otrzymał tego typu rozwiązanie (pierwsza liczba ciągu – 56536, długość ciągu – 149, NWD – 74)

Przykład 12.

```

while(!input3.eof()){
    input3 >> temp;
    hold = NWD(temp, dzielnik);
    if(hold != 1){
        len++;
        dzielnik = hold;
    } else {
        first = temp;
        len = 1;
        dzielnik = first;
    }
    if(len > bestLen){
        bestFirst = first;
        bestLen = len;
        bestDzielnik = dzielnik;
    }
}

```

Zdający w każdym przejściu pętli pobiera nową liczbę i wylicza kolejne dzielniki. Nie zauważa, że liczba, którą sprawdził, a która dała podczas liczenia NWD wynik 1 (z poprzednią liczbą lub poprzednią wartością dzielnika) powinna zostać sprawdzona jeszcze raz z następną liczbą, ponieważ może być początkiem następnego ciągu spełniającego warunki zadania.

Dość często zdający wyliczali największe wspólne dzielniki par sąsiadujących liczb zamiast wspólnych dzielników całych ciągów liczb, co dawało nieprawidłowe wyniki i wskazywało na niewłaściwe zrozumienie zagadnienia przedstawionego w zadaniu. Np. dla liczb z drugiego przykładu zamieszczonego w treści zadania (5, 70, 28, 42, 98, 1) otrzymalibyśmy w wyniku ciąg zaczynający się od liczby 5, a kończący na 98 (każda para liczb od 5 do 98 daje największe wspólne dzielniki większe od 1). Zdający, którego fragment programu przedstawiony jest w Przykładzie 13. popełnił między innymi wymieniony błąd. Dodatkowo ma błędny warunek końca pętli wewnętrznej oraz jako największy wspólny dzielnik całego ciągu podaje dzielnik dwóch pierwszych liczb tego ciągu.

Przykład 13.

```
for (int x=0;x<500;x++)
{
    d=0;
    if (nwd (tab[x], tab[x+1])>1)
    {
        p=tab[x];
        d++;
        w=nwd (tab[x], tab[x+1]);
        for (int y=x+1;y<500-x;y++)
            if (nwd (tab[y], tab[y+1])>1)
                d++;
            else
                break;
    }
    if (d>dmax)
    {
        dmax=d;
        wmax=w;
        pmax=p;
    }
}
```

3. Wnioski i rekomendacje

1. Analiza wyników tegorocznego egzaminu maturalnego pozwala zauważyć, że zdający dość dobrze radzą sobie z typowymi zadaniami z zakresu opracowywania i przetwarzania informacji z różnych źródeł oraz przeprowadzania analizy statystycznej danych w arkuszu kalkulacyjnym. Trudności natomiast sprawiają zdającym zadania obejmujące umiejętności z ww. obszarów, w rozwiązaniu których należy wykorzystać złożenie (zagnieżdżenie) funkcji arkusza czy zastosować złożone zapytania w bazach danych.
2. Niezmiernie duże trudności sprawiają zdającym zadania wymagające napisania programu realizującego rozwiązanie problemu postawionego w treści zadania, czego skutkiem jest częste pomijanie tego typu zadań. Aby uczniowie nie bali się podejmować prób rozwiązania zadań dotyczących programowania podczas egzaminu maturalnego warto zwiększyć liczbę tego typu zadań rozwiązywanych podczas nauki. Warto również zachęcać uczniów do pracy własnej i ćwiczenia umiejętności rozwiązywania typowych zadań programistycznych oraz ćwiczenia umiejętności stosowania znanych algorytmów do rozwiązywania różnych problemów.
3. Trudności zdających często wynikają z pobieżnego analizowania treści zadania i załączonych przykładów lub niezrozumienia zawartych w nich informacji. Ważne jest, aby zdający zwrócili większą uwagę na uważne czytanie i analizowanie zadania przed przystąpieniem do jego realizacji, w tym także na wskazówki dotyczące formy odpowiedzi.
4. Istotnym elementem rozwiązań zadań egzaminacyjnych z II części arkusza są pliki z realizacją komputerową obliczeń. Stanowią one integralną część rozwiązania, wymaganą przy sprawdzaniu rozwiązania przez egzaminatora. Brak tych plików jest traktowany jako brak rozwiązania zadania i skutkuje uzyskaniem przez zdającego 0 punktów za dane zadanie. Nadal zdarzają się przypadki braku plików z realizacją komputerową rozwiązania zadania. Podczas zajęć informatyki z uczniami warto zwrócić uwagę na sposób zapisywania wyników zadań zgodnie z poleceniami w poszczególnych zadaniach oraz na zapisywanie ich komputerowych realizacji.