

Województwo zachodniopomorskie

Informatyka

**Sprawozdanie z egzaminu maturalnego
w roku 2018**

Opracowanie

Iwona Arcimowicz (Centralna Komisja Egzaminacyjna)

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

Opracowanie dla województwa zachodniopomorskiego

Okręgowa Komisja Egzaminacyjna w Poznaniu

Izabela Szafrąńska

Jacek Pietrzak

Michał Pawlak

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.

Zadania 1. i 2. to typowe zadania sprawdzające umiejętność myślenia algorytmicznego oraz zapisu algorytmu w wybranej przez zdającego notacji. Zadanie 3.1. dotyczyło wiedzy 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. Zadanie 3.2. dotyczyło wiedzy z zakresu przetwarzania obrazów za pomocą komputera i stosowania modeli barw. Zadanie 3.3. było zadaniem sprawdzającym umiejętność posługiwania się relacyjnymi bazami danych, a w szczególności umiejętność posługiwania się strukturalnym językiem zapytań SQL. Zadanie 4. było typowym zadaniem polegającym na rozwiązywaniu problemu z zastosowaniem podejścia algorytmicznego z wykorzystaniem dowolnego języka programowania lub innego narzędzia informatycznego. Zadanie 5. polegało na opracowaniu za pomocą komputera danych liczbowych, wykorzystując analizę statystyczną danych oraz podejścia algorytmicznego do podanego problemu. Zadanie 6. było zadaniem sprawdzającym umiejętność posługiwania się relacyjnymi bazami danych, językiem SQL. Zadanie to można było rozwiązać, korzystając z narzędzi bazodanowych – program bazodanowy (np. MS Access, Apache OpenOffice Base, LibreOffice Base), strukturalnego języka zapytań SQL lub dowolnego języka programowania.

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 2018 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		269
Zdający rozwiązujący zadania w arkuszu standardowym	z liceów ogólnokształcących	87
	z techników	182
	ze szkół na wsi	0
	ze szkół w miastach do 20 tys. Mieszkańców	28
	ze szkół w miastach od 20 tys. do 100 tys. Mieszkańców	77
	ze szkół w miastach powyżej 100 tys. Mieszkańców	164
	ze szkół publicznych	213
	ze szkół niepublicznych	56
	kobiety	20
	mężczyźni	249

* Dane w tabeli dotyczą tegorocznych absolwentów.

Z egzaminu zwolniono 1 absolwenta – laureata/finalistę Olimpiady Informatycznej.

Tabela 2. Zdający rozwiązujący zadania w arkuszach dostosowanych

Zdający rozwiązujący zadania w arkuszach dostosowanych	z autyzmem, w tym z zespołem Aspergera	2
	słabowidzący	0
	niewidomi	0
	słabosłyszący	0
	niesłyszący	0
	ogółem	2

3. Przebieg egzaminu

Tabela 3. Informacje dotyczące przebiegu egzaminu

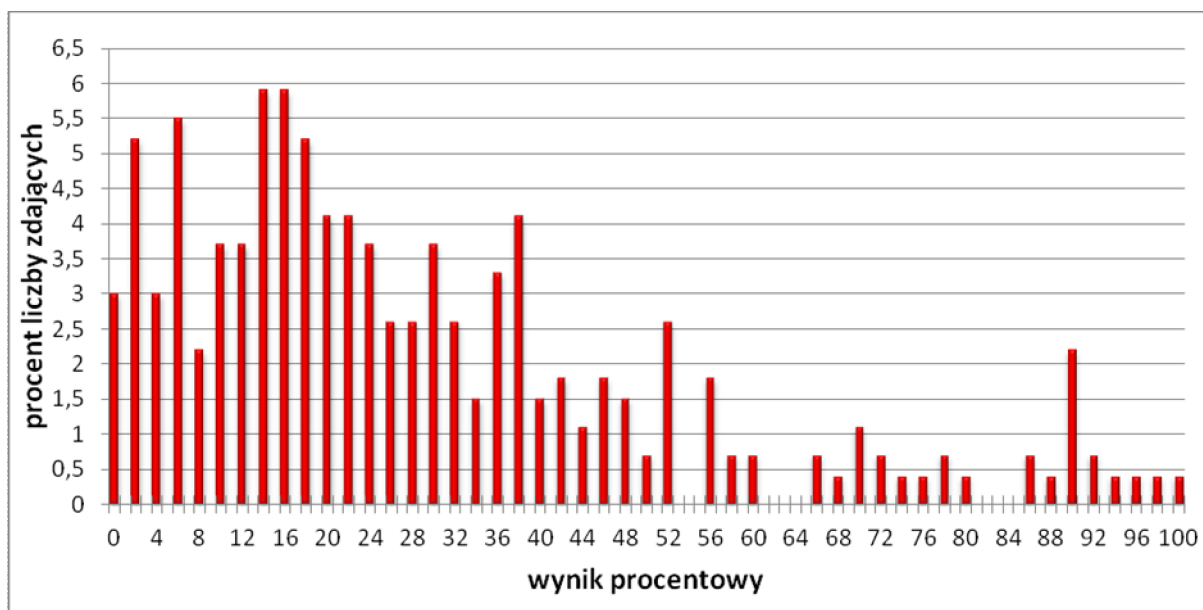
Termin egzaminu		11 maja 2018	
Czas trwania egzaminu		210 minut	
Liczba szkół		49	
Liczba zespołów egzaminatorów		0	
Liczba egzaminatorów		0	
Liczba obserwatorów ¹ (§ 8 ust. 1)		0	
Liczba unieważnień ²	w przypadku:		
	art. 44zzv pkt 1	stwierdzenia niesamodzielnego rozwiązywania zadań przez zdającego	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. 44zzv ust. 1.	stwierdzenia podczas sprawdzania pracy niesamodzielnego rozwiązywania zadań przez zdającego	0
	art. 44zzy ust. 7	stwierdzenia naruszenia przepisów dotyczących przeprowadzenia egzaminu maturalnego	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 21 grudnia 2016 r. w sprawie szczegółowych warunków i sposobu przeprowadzania egzaminu gimnazjalnego i egzaminu maturalnego (Dz.U. z 2016 r., poz. 2223).

² Na podstawie ustawy z dnia 7 września 1991 r. o systemie oświaty (tekst jedn. Dz.U. z 2017, poz. 2198, 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 (%)	Modalna (%)	Średnia (%)	Odchylenie standardowe (%)
ogółem	271	0	100	22	14-16(2)	28,96	23,60
w tym:							
z liceów ogólnokształcących	87	0	98	31	14	36,18	26,50
z techników	184	0	100	20	16	25,49	21,29

* 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: 11) opisuje podstawowe algorytmy i stosuje: a) algorytmy na liczbach całkowitych, 16) opisuje własności algorytmów na podstawie ich analizy; 17) ocenia zgodność algorytmu ze specyfikacją problemu; 18) oblicza liczbę operacji wykonywanych przez algorytm.	38
1.2.	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, 16) opisuje własności algorytmów na podstawie ich analizy; 17) ocenia zgodność algorytmu ze specyfikacją problemu; 18) oblicza liczbę operacji wykonywanych przez algorytm.	18
1.3.	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, 16) opisuje własności algorytmów na podstawie ich analizy; 17) ocenia zgodność algorytmu ze specyfikacją problemu; 18) oblicza liczbę operacji wykonywanych przez algorytm.	51
2.1.	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), np.: – jednoczesne znajdowanie największego i najmniejszego elementu w zbiorze: algorytm naiwny i optymalny, – algorytmy sortowania ciągu liczb: bąbelkowy, przez wybór, przez wstawianie liniowe lub binarne, przez scalanie, szybki, kubełkowy.	26
2.2.	III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia	5. Rozwiązywanie problemów i podejmowanie decyzji [...], stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin;	20

	algorytmicznego.	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), np.: – algorytmy sortowania ciągu liczb: bąbelkowy, przez wybór, przez wstawianie liniowe lub binarne, przez scalanie, szybki, kubełkowy.	
3.1.	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.	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.	46
3.2.	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: 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL).	45
3.3.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.	1. Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej. Zdający: 3) określa ustawienia sieciowe danego komputera i jego lokalizacji w sieci, prawidłowo posługuje się terminologią sieciową. 7. Zdający wykorzystuje komputer i technologie informacyjno-komunikacyjne do rozwijania swoich zainteresowań, opisuje zastosowania informatyki, ocenia zagrożenia i ograniczenia, docenia aspekty społeczne rozwoju i zastosowań informatyki: 3) stosuje normy etyczne i prawne związane z rozpowszechnianiem programów komputerowych, bezpieczeństwem i ochroną danych oraz informacji w komputerze i w sieciach komputerowych; 4) omawia zagadnienia przestępczości komputerowej, w tym piractwo komputerowe, nielegalne transakcje w sieci.	18
4.1.	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia	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	31

	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: d) algorytmy na tekstach.</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: d) algorytmy na tekstach.</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>	14
4.3.	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia	<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</p>	13

	<p>algorytmicznego.</p>	<p>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: d) algorytmy na tekstach. 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.</p>	
5.1.	<p>II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.</p> <p>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. 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; 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; 8) posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów.</p>	74
5.2.	<p>II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów,</p>	<p>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</p>	38

	<p>danych liczbowych, motywów, animacji, prezentacji multimedialnych.</p> <p>III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.</p>	<p>z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.</p> <p>Zdający:</p> <ol style="list-style-type: none"> 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; 8) posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów. 	
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> <ol style="list-style-type: none"> 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. <p>5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.</p> <p>Zdający:</p> <ol style="list-style-type: none"> 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. 	63
5.4.	<p>II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych,</p>	<p>4. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów.</p> <p>Zdający:</p> <ol style="list-style-type: none"> 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. 5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia 	13

	<p>motywów, animacji, prezentacji multimedialnych.</p> <p>III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.</p>	<p>algorytmicznego.</p> <p>Zdający:</p> <ol style="list-style-type: none"> 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. 	
6.1.	<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.</p> <p>Zdający:</p> <ol style="list-style-type: none"> 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. 	48
6.2.	<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.</p> <p>Zdający:</p> <ol style="list-style-type: none"> 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. 	39
6.3.	<p>II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.</p>	<p>2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.</p> <p>Zdający:</p> <ol style="list-style-type: none"> 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. 	20
6.4.	<p>II. Wyszukiwanie,</p>	<p>2. Wyszukiwanie, gromadzenie, selekcjonowanie,</p>	23

	gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.	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.	
6.5.	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.	11

Komentarz

W maju 2018 roku absolwenci po raz czwarty 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 1 oraz 35 za część 2.

1. Analiza jakościowa zadań

Tegoroczna matura z informatyki okazała się być trudna, tak jak i w latach poprzednich. Poziom wykonania zadań mieścił się w przedziale od 11% do 745%.

Najtrudniejszym zadaniem dla zdających okazało się zadanie 6.5. (poziom wykonania – 11%) z zakresu wyszukiwania i przetwarzania danych (bazodanowe) – zostanie ono omówione dokładnie w dalszej części komentarza.

Do najtrudniejszych zadań (poziom wykonania 0%–19%), poza wymienionym wcześniej, należały także zadania: 4.3. (13%), 4.2. (14%), 3.3. (18%) – są to zadania programistyczne oraz bazodanowe, a także zadania: 1.2. (18%) - zadanie algorytmiczne oraz 5.4. (13%) – zadanie wymagające zaprojektowania symulacji w arkuszu kalkulacyjnym.

Umiejętności programistyczne, w kategorii zadań najtrudniejszych, sprawdzały dwa zadania: 4.2. oraz 4.3. Każde z nich wymagało umiejętności ułożenia algorytmu oraz jego zaimplementowania. Zdający otrzymali plik z danymi, na podstawie którego rozwiązywali zadania, a ponadto mieli do dyspozycji przykładowy plik i wyniki, jakie powinni otrzymać przy jego zastosowaniu.

W zadaniu 4.3. należało napisać program wypisujący z podanego pliku słowa, w których każda z liter była oddalona od każdej innej litery tego samego słowa najwyżej o 10 pozycji. Było więc to typowe zadanie programistyczne wymagające umiejętności ułożenia i zaimplementowania odpowiedniego algorytmu. Duża część zdających w tym zadaniu sprawdzała odległości między sąsiednimi literami słowa zamiast, zgodnie z treścią zadania, sprawdzić odległość między każdą parą liter w słowie. Zadanie to (tak jak i 4.2.) było także często pomijane.

W zadaniu 4.2. należało napisać program znajdujący w pliku z danymi słowo składające się z największej liczby różnych liter. Najtrudniejszym z elementów w tym zadaniu było zliczenie liczby różnych liter w danym słowie. Jednym z częstszych sposobów zliczania liter, stosowanych przez zdających, było tworzenie zmiennej pomocniczej złożonej z różnych liter danego słowa (bez powtórzeń). Drugim częstym sposobem było także tworzenie jako zmiennej pomocniczej tablicy reprezentującej wszystkie litery alfabetu i zawierającej ich ilość w danym słowie. Częstymi błędami powodującymi otrzymywanie błędnych wyników były typowe błędy logiczne, np. pomijanie zmiany wartości zmiennych pomocniczych przy kolejnym przejściu pętli. Na przykład we fragmencie rozwiązania poniżej zabrakło przypisania wartości zmiennej `czy=1` po każdym przejściu zewnętrznej pętli `for` (Przykład 1.):

Przykład 1.

```

45 //-----
46 czy=1;
47 iler=0;
48 buf="";
49 for (int i = 0; i < dl; i++)
50 {
51     for ( int h = 0; h<buf.length(); h++)
52     {
53         if(s[i]==buf[h])
54         {
55             czy=0;
56             break;
57         }
58     }
59     if(czy==1)
60     {
61         iler++;
62         buf=buf+s[i];
63     }
64 }
65 if(iler>ilemax)
66 {
67     ilemax=iler;
68     sbuf=s;
69 }
70 }

```

Pozostawione błędy wskazywałyby na to, że część osób nie sprawdziła wyników na przykładach, mimo że miała taką możliwość. Z drugiej strony zdarzały się pojedyncze przypadki, w których zdający spisywali odpowiedzi dla pliku przykładowego (czyli takie, jakie mieli przecież w arkuszu) zamiast dla pliku, na którym powinni pracować.

Zdarzały się także pojedyncze przypadki prób rozwiązania zadania 4. z pomocą arkusza kalkulacyjnego, mimo że w treści zadania wyraźnie napisano, że należy w tym przypadku napisać program.

Zadania bazodanowe w kategorii zadań najtrudniejszych to zadanie 6.5., 6.3. oraz 3.3. W zadaniu 6.5. należało podać liczbę komputerów, które nie uległy awarii o zadanym priorytecie, natomiast w 6.3. należało podać dzień i nazwę sekcji komputerów, w której tego dnia uległy awarii wszystkie komputery – obydwie zadania zostaną dokładniej omówione w dalszej części. Natomiast zadanie 3.3. było jednym z zadań testowych w części pierwszej egzaminu dotyczącym języka SQL. Tak jak we wszystkich zadaniach testowych w arkuszu zdający dostawał punkt tylko wtedy, gdy poprawnie ocenił prawdziwość każdego z czterech zdań. W pytaniu tym najczęściej błędna odpowiedź zaznaczana była w pierwszym i ostatnim zdaniu dotyczącym składni i działania zapytań SQL.

W zadaniu algorytmicznym 1.2. należało podać najmniejszą i największą liczbę n , dla której wynikiem działania algorytmu jest $p=10$. Poprawnym rozwiązaniem zadania 1.2. są liczby 730 i 1000.

Zadanie 5.4. wymagało wykonania symulacji w arkuszu kalkulacyjnym. Jest to tradycyjnie jeden z trudniejszych typów zadań – w ubiegłym roku poziom wykonania tego typu zadania wynosił poniżej 10%, w obecnym wyniki są znacznie lepsze. W zadaniu opisano cykl pracy zbiornika retencyjnego, do którego wpływała woda z rzeki Wirki. Należało wykonać symulację i odpowiedzieć na kilka pytań: podać dzień, w którym po raz pierwszy wypuszczono wodę po przepelnieniu zbiornika, ilość dni, w których, w momencie pomiaru, w zbiorniku znajdowało się co najmniej 800 tys. m³ wody oraz

maksymalny stan, jaki osiągnąłby zbiornik, gdyby zrezygnowano z wypuszczania wody, a pojemność nie byłaby ograniczona.

Najczęstszym błędem zdających była zła interpretacja przepełnienia zbiornika – część osób, przy prawidłowych obliczeniach, wskazywała datę, w której zbiornik miał dokładnie 1000000 m³ wody, a nie powyżej tej sumy. Część zdających w momencie wystąpienia przepełnienia zbiornika myliła kolejność symulowanych zdarzeń, otrzymując nieprawidłowe wartości. Pojawiały się także błędne obliczenia, np. zgodnie z treścią zadania „codziennie rano (o godzinie 8) ze zbiornika wypuszcza się 2% objętości wody wykazanej przez pomiar zaraz po północy. Ilość wypuszczanej wody zaokrągla się w górę” – zdarzały się osoby zaokrąglające ilość m³ wody po odjęciu 2%, zamiast zaokrąglić 2% odejmowanej sumy, co dawało nieco inne wartości od oczekiwanych. Większość błędów wynikała wprost z niedokładnego czytania poleceń.

Do zadań trudnych (poziom wykonania od 20% do 49%) należały zadania: 6.4. (23%), 2.2. (20%), 4.1. (31%), 2.1. (26%), 3.2. (45%), 5.2. (38%), 3.1. (46%), 6.2. (39%), 1.1. (38%), 6.1. (48%), 6.3. (20%). W grupie tych zadań najliczniej były reprezentowane zadania wymagające przeprowadzenia analizy algorytmu, zapisu algorytmu i zadania programistyczne (zadania 1.1., 2.1., 2.2., 4.1.).

Zadaniami trudnymi, jak zwykle, okazały się zadania algorytmiczne z pierwszej części arkusza. Relatywnie najlepiej w tej grupie (poza wymienionym w dalszej części 1.3.) wypadło zadanie 1.1., w którym należało przeprowadzić analizę przedstawionego algorytmu i podać wynik jego działania dla trzech liczb. W zadaniu 1.1. można było rozpisać kolejne kroki dla podanych liczb.

Zadania 2.1. i 2.2. dotyczyły tworzenia algorytmów. Sprawdzały one podstawowe umiejętności z podstawy programowej. Najtrudniejszy okazał się sam zapis algorytmu. W zadaniu 2.1. należało napisać algorytm wyszukujący współrzędne skrajnie lewego szczytu (wcześniej w treści zadania opisana jest definicja wyjaśniająca, czym jest położenie szczytów). W zadaniu 2.2. natomiast należało zapisać algorytm porządkujący współrzędne szczytów, tak by szczyty były ustawione w kolejności od lewej do prawej z punktu widzenia obserwatora. W obu zadaniach podana była specyfikacja danych na wejściu i wyniku.

Warto zauważyć, że z dokładnością do samej operacji porównania, zdefiniowanej we wstępie do zadania, algorytm dla zadania 2.1. to szukanie najmniejszego elementu, a dla zadania 2.2. to sortowanie elementów (przy tym wchodziły w grę najprostsze algorytmy sortowania, ponieważ dopuszczalna była złożoność kwadratowa lub mniejsza). Są to podstawowe algorytmy i teoretycznie nie powinny sprawić trudności zdającym. Podstawową trudnością był fakt, że trzeba było przy zapisywaniu rozwiązań pamiętać o tym, jak ma być zapisany wynik i jak mają wyglądać operacje porównania. Podczas rozwiązywania zadania 2.2. nie można było używać funkcji bibliotecznych dostępnych w językach programowania, wypisane także były dozwolone operacje, co zapobiegało rozwiązaniom w postaci jednej linijki wywołującej funkcję sortującą. Mimo tych zastrzeżeń niektórzy zdający używali w zapisie np. funkcji *swap* zamiast zapisywać zamianę elementów, w tym przypadku w ocenie otrzymywali jeden punkt mniej.

Poniżej przykładowe poprawne rozwiązanie zadania 2.1. (Przykład 2.)

Przykład 2.

```

x ← X[1]
y ← Y[1]
i ← 2
dopóki i ≤ n wykonuj:
    jeżeli X[i]/Y[i] < x/y wykonaj:
        x ← X[i]
        y ← Y[i]
    i ← i + 1

```

Przykładowe poprawne rozwiązanie zadania 2.2. (Przykład 3.)

Przykład 3.

```

for (int i=1; i<=n; i++)
    for (int l=i+1; l<=n; l++)
        if ((X[i]/Y[i]) > (X[l]/Y[l]))
            { buff = X[i];
              X[i] = X[l];
              X[l] = buff;
              buff = Y[i];
              Y[i] = Y[l];
              Y[l] = buff; }

```

Wśród zadań trudnych znalazło się także pytanie testowe wymagające wiedzy z zakresu podstaw elementów grafiki komputerowej (3.2.). Poza samą specyfiką zadania testowego, zawierającego cztery zdania, których prawdziwość musiał bezbłędnie ocenić zdający, aby otrzymać 1 punkt, problemy sprawiły pytania dotyczące modelu barw CMYK, a zwłaszcza jego zastosowanie i rozszyfrowanie, co oznaczają poszczególne litery skrótu – czyli pierwsze i ostatnie zdanie.

Do zadań umiarkowanie trudnych (poziom wykonania z zakresu od 50% do 69%) należało zadanie 1.3. (51%), w którym należało tylko wybrać prawidłową odpowiedź oznaczającą liczbę operacji

wykonywanych przez podany algorytm, oraz zadanie 5.3 (63%), którego rozwiązanie wymagało wykonania symulacji w arkuszu kalkulacyjnym.

Zdający dość dobrze poradzili sobie z zadaniem 5.1 (poziom wykonania – 74%). Było to zadanie z zakresu analizy statystycznej danych przeznaczone do rozwiązania w arkuszu kalkulacyjnym. Zadanie to można było wykonać, korzystając z podstawowych funkcji arkusza.

2. Problem „pod lupą”: zadanie bazodanowe

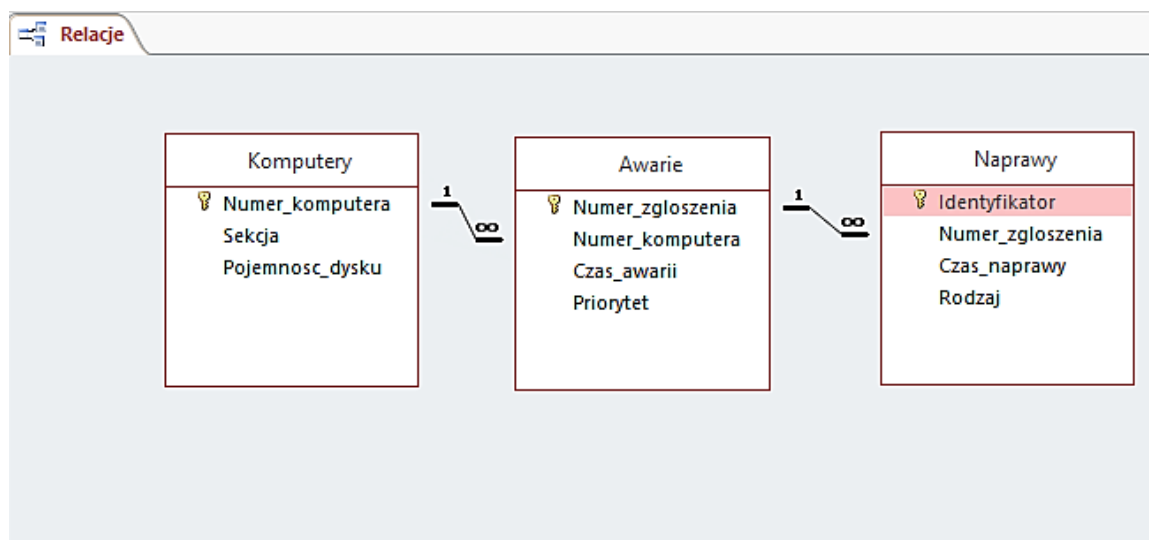
W tym roku zadanie z części praktycznej z zakresu wyszukiwania i przetwarzania danych, czyli z zakresu baz danych okazało się bardzo trudne. Zadanie 6.5. (poziom wykonania 11%) było najtrudniejszym zadaniem z całego arkusza. Bardzo trudne okazało się także zadanie 6.4. (poziom wykonania – 23%) i 6.3. (poziom wykonania – 20%).

Zdecydowana większość zdających rozwiązywała to zadanie w MSAccess. Zdarzały się pojedyncze przypadki rozwiązań, w których używano także pomocniczo arkusza kalkulacyjnego.

W zadaniu zdający otrzymali trzy pliki z danymi: `komputery.txt`, `awarie.txt` oraz `naprawy.txt`. Wykonując zadanie w programie bazodanowym, należało, przed przystąpieniem do realizacji kolejnych zadań, najpierw zaimportować dane z plików tekstowych (pamiętając przy tym o poprawnym formacie dla daty – RMD) do tabel *Awarie*, *Komputery* i *Naprawy* oraz odpowiednio połączyć tabele relacjami. Następnie w zadaniach 6.3. – 6.5. odpowiedzieć na pytania, wykonując odpowiednie kwerendy.

Oczywiście przed wykonaniem podpunktów zadania należało najpierw zaimportować dane z plików tekstowych do tabel *Awarie*, *Komputery* i *Naprawy* oraz odpowiednio połączyć tabele relacjami. Poniżej przykład widoku relacji ukazujący powiązania między tabelami (Przykład 4.).

Przykład 4.



Zadanie 6.3.

W tym zadaniu należało podać datę, w której wszystkie komputery z jednej sekcji uległy awarii. Należało podać także nazwę sekcji.

Przykładowe poprawne rozwiązanie zdającego:

Kwerenda pomocnicza 1 („6.3 robo”) zliczająca liczbę awarii komputerów z podziałem na sekcje i daty. Widok projektu i wynik działania kwerendy przedstawiono poniżej (Przykład 5.)

Przykład 5.

The screenshot shows the Microsoft Access interface. At the top, there are three tabs: '6_3_robo', '6_3', and '6_3_robo_2'. Below the tabs, there are two table objects: 'Awarii' and 'Komputery'. The 'Awarii' table has fields: Numer_zgloszenia, Numer_komputera, Czas_awarii, and Priorytet. The 'Komputery' table has fields: Numer_komputera, Sekcja, and Pojemnosc_dysku. A relationship line connects the two tables, with a cardinality of 1 to many (∞).

Below the design view, there is a query design grid. The fields are: Miesiac: Month([Awarie].[Czas_awarii]), Dzień: Day([Awarie].[Czas_awarii]), Sekcja: Komputery.Sekcja, and Policz: Count([Komputery].[Numer_komputera]). The 'Suma' row is set to 'Grupuj według' for all fields. The 'Sortuj' row is empty. The 'Pokaż' row has checkboxes checked for all fields. The 'Kryteria' row is empty. The 'Iub' row is empty.

Below the design grid, there is a results view showing a table with the following data:

Miesiac	Dzień	Sekcja	PoliczOfNumer_kompu
12	23	Q	14
12	21	N	3
7	15	N	3
7	19	O	3
11	6	E	3
2	28	S	3
7	29	S	3
3	5	F	3
5	15	T	3
3	8	Q	3
1	21	C	2

Zdający w tym przypadku oddzielnie wyznaczył dzień i miesiąc z daty awarii (można było tak postąpić, ponieważ wszystkie dane pochodziły z jednego roku i nie było niebezpieczeństwa, że dane dotyczące daty w wyniku byłyby niejednoznaczne).

Zapis SQL:

```
SELECT Month(Awarie.Czas_awarii) AS Miesiac, Day(Awarie.Czas_awarii)
AS Dzień, Komputery.Sekcja, Count(Komputery.Numer_komputera) AS
PoliczOfNumer_komputera
FROM Komputery INNER JOIN Awarie ON Komputery.Numer_komputera =
Awarie.Numer_komputera
GROUP BY Month(Awarie.Czas_awarii), Day(Awarie.Czas_awarii),
Komputery.Sekcja;
```

Kwerenda pomocnicza 2 zliczająca liczbę komputerów w każdej z sekcji (Przykład 6.)

Przykład 6.

Pole:	Sekcja	Numer_komputera
Tabela:	Komputery	Komputery
Suma:	Grupuj według	Policz
Sortuj:		
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:		
lub:		

Sekcja	PoliczOfNur
A	20
B	16
C	21
D	14
E	28
F	26
G	19
H	26
I	20
J	22
K	16
L	11
M	19
N	21
O	23
P	15
Q	14
R	18
S	19

Zapis SQL:

```
SELECT Komputery.Sekcja, Count (Komputery.Numer_komputera) AS
PoliczOfNumer_komputera
FROM Komputery
GROUP BY Komputery.Sekcja;
```

Kwerenda końcowa dająca odpowiedź do zadania 6.3. (Przykład 7.)

Przykład 7.

The screenshot shows a database query interface with two tables, 6_3_rob0 and 6_3_rob0_2, and a query result table.

Table 6_3_rob0:

6_3_rob0
*
Miesiac
Dzien
Sekcja
PoliczOfNumer_komp

Table 6_3_rob0_2:

6_3_rob0_2
*
Sekcja
PoliczOfNumer_kompu

Query Parameters:

Pole:	Dzien	Miesiac	Sekcja
Tabela:	6_3_rob0	6_3_rob0	6_3_rob0
Sortuj:			
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:			
lub:			

Query Result:

Dzien	Miesiac	Sekcja
23	12	Q

Zapis SQL:

```
SELECT 6_3_rob0.Dzien, 6_3_rob0.Miesiac, 6_3_rob0.Sekcja
FROM
6_3_rob0_2 INNER JOIN 6_3_rob0 ON (6_3_rob0.PoliczOfNumer_komputera
= 6_3_rob0_2.PoliczOfNumer_komputera) AND (6_3_rob0_2.Sekcja =
6_3_rob0.Sekcja);
```

Wynikiem jest data 23.12.2015 i sekcja Q.

Wielu zdających pomijało ostatnią kwerendę, odczytując sekcję z posortowanej kwerendy 1 i ręcznie porównując liczbę komputerów obliczoną w kwerendzie 2. Ułatwiało takie postępowanie fakt, że tylko w jednym dniu nastąpiła awaria wielu komputerów.

Inne podobne rozwiązanie: (kwerendy 3a i 3b są odpowiednikami kwerend 1 i 2) – pojawia się tutaj wyrażenie (wyr3) sprawdzające, czy liczba komputerów w sekcji równa jest liczbie komputerów, które uległy awarii. (Przykład 8.)

Przykład 8.

Pole:	Wyr1	Wyr2	Wyr3: Iif([3a].[PoliczOfNumer_komputera]=[3b].[PoliczOfNumer_komputera],	Sekcja
Tabela:	3a	3a		3b
Sortuj:			Malejąco	
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:			1	
Iub:				

Zadanie to było dość często pomijane. Zdający, które je rozwiązywali, nie mieli zwykle problemów z obliczeniem liczby komputerów w każdej sekcji, trudniejsze było policzenie liczby komputerów, które uległy awarii jednego dnia, w jednej sekcji, ze względu na fakt, że „czas awarii” podany został w formacie oprócz daty zawierającej godzinę – trzeba było więc najpierw uzyskać samą datę (np. przy pomocy funkcji *DateValue*).

Zadanie 6.4.

W tym zadaniu należało znaleźć awarię, której usunięcie trwało najdłużej. Podać należało numer zgłoszenia znalezionej awarii, czas jej wystąpienia i czas (datę) zakończenia naprawy.

Tu rozwiązanie można było otrzymać za pomocą jednej kwerendy.

Aby obliczyć czas trwania naprawy, można było użyć funkcji *DateDiff*, ale wystarczyło także zwykłe odejmowanie.

W poniższym przykładzie zdający uporządkował wyniki według długości trwania naprawy malejąco, następnie, aby otrzymać tylko te dane, o które pytano w zadaniu, ograniczył liczbę wyświetlanych wierszy do jednego (Przykład 9.).

Przykład 9.

The screenshot displays the Microsoft Access interface. At the top, there is a ribbon with various query-related tools. Below the ribbon, the 'Typ kwerendy' (Query Type) is set to 'Kwerenda' (Query). The 'Konfiguracja kwerendy' (Query Configuration) section shows three tables: 'Komputery', 'Awarie', and 'Naprawy'. 'Komputery' is linked to 'Awarie' with a 1-to-many relationship, and 'Awarie' is linked to 'Naprawy' with a 1-to-many relationship. The 'Naprawy' table is highlighted in the design view, and a calculated field is defined: $[Naprawy]![Czas_naprawy] - [Awarie]![Czas_awarii]$. Below the design view, the 'Kryteria' (Criteria) section shows the calculated field is set to 'Malejąco' (Descending). The 'Wyniki' (Results) section shows a table with the following data:

Numer_zgloszenia	Czas_awarii	Czas_naprawy
2087	06.11.2015 12:38:46	13.11.2015 12:38:32

Tak jak w przypadku zadania 6.3. wystąpiła duża frakcja opuszczeń. Część osób rozwiązujących to zadanie szukała najpierw dla każdego zgłoszenia daty ostatniej naprawy – nie było to jednak konieczne, bo jeśli szukamy największej różnicy między wystąpieniem awarii a czasem zakończenia naprawy, to nie ma znaczenia, ile było napraw dla danego zgłoszenia.

Jeśli odpowiedzi na to pytanie były umieszczane, to w przeważającej większości były prawidłowe. Niektórzy zdający przy rozwiązywaniu tego zadania wspomagali się arkuszem kalkulacyjnym.

Zdarzały się przypadki błędnej interpretacji. Zdający w przykładzie poniższym (Przykład 10.) zsumował odstępy czasu między wystąpieniem awarii i każdą naprawą, mimo że w treści zadania wyraźnie jest napisane, że czas liczymy od wystąpienia awarii do momentu zakończenia ostatniej naprawy:

Przykład 10.

The screenshot shows the 'Kryteria' (Criteria) section of a query design view. The calculated field is defined as: $Suma([Naprawy]![Czas_naprawy] - [Awarie]![Czas_awarii])$. The criteria are set to 'Malejąco' (Descending). The 'Wyniki' (Results) section shows a table with the following data:

Numer_zgloszenia	Suma([Naprawy]![Czas_naprawy]-[Awarie]![Czas_awarii])
2087	06.11.2015 12:38:46 - 13.11.2015 12:38:32

Zadanie 6.5.

To zadanie było najtrudniejszym zadaniem na tegorocznym egzaminie z informatyki.

Poprawne rozwiązanie wymagało znalezienia liczby komputerów, które nie uległy awarii o priorytecie większym lub równym 8 (czyli uległy awarii o priorytecie mniejszym niż 8), a także liczby komputerów, które w ogóle nie uległy awarii.

Przykładowe rozwiązanie 1.

Krok 1

Wyszukanie komputerów, które uległy awarii o priorytecie równym lub większym niż 8. (Przykład 11.)

Przykład 11.

Pole:	Priorytet	Numer_komputera	
Tabela:	Awarie	Awarie	
Sortuj:			
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Kryteria:	>=8		
lub:			

Zapis SQL:

```
SELECT Awarie.Priorytet, Awarie.Numer_komputera
FROM Awarie
WHERE ((Awarie.Priorytet)>=8);
```

Warto zauważyć, że w wyniku tej kwerendy numery komputerów, które miały więcej niż jedną awarię spełniającą kryteria, się powtarzają.

Krok 2

Wyszukanie komputerów, które nie uległy awarii o priorytecie równym lub większym niż 8 (w tym także tych, które wcale nie uległy awarii), czyli których nie ma w wyniku poprzedniej kwerendy. (Przykład 12.)

Przykład 12.

Pole:	Komputery.*	Numer_komputera	
Tabela:	Komputery	6_5pomoc	
Sortuj:			
Pokaż:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kryteria:		Is Null	
Iub:			

Jak widać podczas konstruowania kwerendy zmieniono właściwości sprzężenia tak by wybierane były wszystkie elementy z tabeli *komputery* i tylko pasujące z kwerendy pomocniczej, a następnie dopisano warunek *is Null* dla pola z kwerendy pomocniczej. (Przykład 13.)

Przykład 13.

komputery.*	Numer_komputera				
komputery	6_5pomoc				
<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Is Null				

Zapis SQL

```
SELECT Komputery.*
FROM Komputery LEFT JOIN 6_5pomoc ON Komputery.Numer_komputera =
6_5pomoc.Numer_komputera
WHERE (((6_5pomoc.Numer_komputera) Is Null));
```

Można było wybrać tylko numer komputera z tabeli *komputery*, ale nadmiar nie przeszkadza. Wynikiem jest lista 149 komputerów spełniających warunki zadania. (Przykład 14.)

Przykład 14.

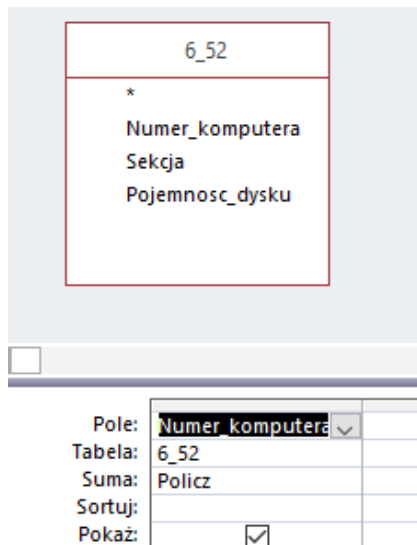
Numer_kom	Sekcja	Pojemnosc
8	A	300
11	M	500
17	K	300
19	H	300
26	T	700
44	Z	700
46	N	300
49	C	300
55	G	290
58	T	500
62	B	300
65	A	500
67	W	400
68	L	320
69	T	200

Zapis SQL z zagnieżdżonym zapytaniem pomocniczym z kroku 1 (takie rozwiązanie można zastosować np. używając MySQL zamiast MSAccess)

```
SELECT Komputery.*
FROM Komputery LEFT JOIN
(
    SELECT Awarie.Priorytet, Awarie.Numer_komputera
    FROM Awarie WHERE Awarie.Priorytet>=8
) AS `6_5pomoc`
ON Komputery.Numer_komputera = 6_5pomoc.Numer_komputera
WHERE (6_5pomoc.Numer_komputera) Is Null;
```

Zdający na koniec wykonał jeszcze kwerendę zliczającą ilość komputerów w kwerendzie pomocniczej 6_52 aby otrzymać wynik 149. (Przykład 15.)

Przykład 15.

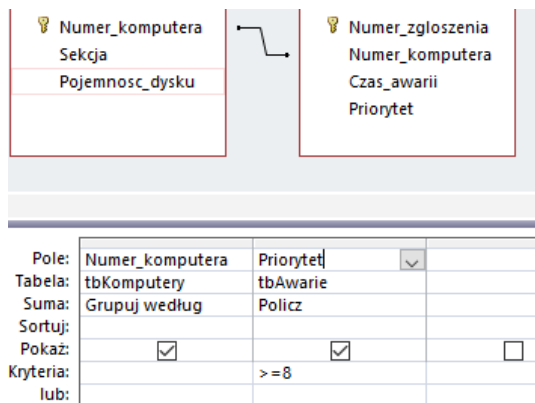


Odpowiednik w języku SQL (w formie jednego zapytania z podzapytaniem):

```
SELECT COUNT(6_52.Numer_komputera)
FROM
(
  SELECT Komputery.*
  FROM Komputery LEFT JOIN
  (
    SELECT Awarie.Priorytet, Awarie.Numer_komputera
    FROM Awarie
    WHERE Awarie.Priorytet >= 8 ) AS `6_5pomoc`
  ON Komputery.Numer_komputera = 6_5pomoc.Numer_komputera
  WHERE (((6_5pomoc.Numer_komputera) Is Null))
) AS `6_52`;
```

Poniżej (Przykład 16.) przykład błędnie wykonanej kwerendy mającej wyszukiwać numery komputerów, które uległy awarii (i prawdopodobnie od razu eliminować powtórzenia):

Przykład 16.



Najczęstsze błędy popełniane przez zdających to:

- zliczanie także komputerów, które miały awarię, o priorytecie równym 8 (otrzymany wtedy wynik 221 oceniany był na 1 punkt),
- zliczanie tylko tych komputerów, które uległy awarii o priorytecie poniżej 8, bez uwzględnienia ilości komputerów, które wcale nie uległy awarii (mimo, że mowa jest o nich w treści zadania)
- błędny sposób zliczania ilości komputerów, które uległy awarii o priorytecie większym lub równym 8,
- policzenie prawidłowo liczby komputerów, które uległy awarii o priorytecie co najmniej równym 8 (jest ich 351), odjęcie ich liczby od liczby wszystkich komputerów, a następnie niepotrzebne doliczenie komputerów, które nie uległy awarii (jest ich 46),
- problemy z policzeniem ilości komputerów tak, by wyeliminować powtórzenia (wiele komputerów ulegało awarii kilkakrotnie),
- podawanie w wyniku liczby komputerów, które w ogóle nie uległy awarii – niezgodnie z treścią zadania.

3. Wnioski i rekomendacje

Poniżej przedstawiono kilka wskazówek, w jaki sposób ukierunkować przygotowania uczniów do matury.

1. We wszystkich zadaniach częste błędy zdających wynikają ze zbyt pobieżnego czytania treści zadań. Należy uczulić uczniów, by przeczytali dokładnie i ze zrozumieniem każde zadanie zanim przystąpią do jego rozwiązywania.
2. Możliwie często zaleca się stosowanie tej samej formy odpowiedzi, jaka jest używana w zadaniach maturalnych oraz zwrócenie uwagi na to, że każda załączona odpowiedź musi być poparta realizacją komputerową rozwiązania.
3. Zdającym nadal duże problemy sprawia rozwiązywanie zadań z zakresu algorytmiki i programowania. Podczas omawiania na lekcjach algorytmów klasycznych warto pokazać więcej ich zastosowań w różnych sytuacjach. Zalecane jest także częstsze używanie podczas zapisu algorytmów na lekcji pseudojęzyka programowania.
4. Bardzo ważne jest zaangażowanie uczniów w rozwiązywanie jak największej liczby zadań z zakresu programowania. Istnieją portale internetowe, na których można ćwiczyć swoje umiejętności, sprawdzać wyniki programów online, a także rywalizować z innymi użytkownikami. Warto uczniom pokazać takie możliwości.
5. Warto więcej uwagi poświęcić rozwiązywaniu zadań z zakresu baz danych.
6. Poprawiły się wyniki uczniów w zakresie posługiwania się arkuszem kalkulacyjnym, ale nadal duże problemy sprawiają zadania wymagające symulacji i stosowania podejścia algorytmicznego.