

<i>Rodzaj dokumentu:</i>	Sprawozdanie za rok 2021 województwo wielkopolskie
<i>Egzamin:</i>	Egzamin maturalny
<i>Przedmiot:</i>	Informatyka
<i>Poziom:</i>	Poziom rozszerzony
<i>Termin egzaminu:</i>	19 maja 2021 r.
<i>Data publikacji dokumentu:</i>	17 września 2021 r.

Opracowanie

Iwona Arcimowicz (Centralna Komisja Egzaminacyjna)

Izabela Szafrńska (Okręgowa Komisja Egzaminacyjna w Poznaniu)

Redakcja

dr Wioletta Kozak (Centralna Komisja Egzaminacyjna)

Opracowanie techniczne

Andrzej Kaptur (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 wielkopolskiego

Okręgowa Komisja Egzaminacyjna w Poznaniu

Izabela Szafrńska

Anna Sperling

Michał Pawlak

Andrzej Popiół

Centralna Komisja Egzaminacyjna

ul. Józefa Lewartowskiego 6, 00-190 Warszawa

tel. 22 536 65 00, fax 22 536 65 04

e-mail: sekretariat@cke.gov.pl

www.cke.gov.pl

Spis treści

Opis arkusza maturalnego	4
Dane dotyczące populacji zdających	4
Przebieg egzaminu	5
Podstawowe dane statystyczne	6
Komentarz	15
Wnioski i rekomendacje	33

Opis arkusza egzaminu maturalnego

W roku szkolnym 2020/2021 egzamin maturalny z informatyki został przeprowadzany na podstawie wymagań egzaminacyjnych określonych w załączniku nr 2 do rozporządzenia Ministra Edukacji Narodowej z dnia 20 marca 2020 r.¹

Egzamin maturalny z informatyki składał się z dwóch części: pisemnej (arkusz I) oraz praktycznej (arkusz II). Tegoroczny zestaw egzaminacyjny zawierał 22 zadania o zróżnicowanym stopniu trudności, w tym 3 zadania typu *prawda-falsz*. Pozostałe zadania otwarte tworzyły 5 wiązek, w których poszczególne polecenia (od 2 do 5 poleceń) odnosiły się do tego samego materiału źródłowego, dedykowanego danej wiązce.

Rozwiązanie zadań egzaminacyjnych wymagało od zdających umiejętności: tworzenia i analizy algorytmów, zapisania (w wybranym przez siebie języku programowania) w postaci programu komputerowego, z wykorzystaniem plików z danymi, rozwiązań postawionych problemów, wykorzystania arkusza kalkulacyjnego, bazy danych lub własnego programu komputerowego do przeprowadzenia symulacji lub analizy danych, znajomości podstawowej wiedzy z różnych obszarów informatyki.

Za rozwiązanie 8 zadań tworzących arkusz I zestawu egzaminacyjnego zdający mógł uzyskać maksymalnie 15 punktów, a za rozwiązanie 14 zadań z arkusza II - 35 punktów.

Egzamin trwał 60 minut w części I oraz 150 minut w części II.

Dane dotyczące populacji zdających

TABELA 1. ZDAJĄCY ROZWIĄZUJĄCY ZADANIA W ARKUSZU STANDARDOWYM*

Liczba zdających		942
Zdający rozwiązujący zadania w arkuszu standardowym	z liceów ogólnokształcących	379
	z techników	563
	ze szkół na wsi	17
	ze szkół w miastach do 20 tys. mieszkańców	171
	ze szkół w miastach od 20 tys. do 100 tys. mieszkańców	322
	ze szkół w miastach powyżej 100 tys. mieszkańców	432
	ze szkół publicznych	928
	ze szkół niepublicznych	14
	kobiety	72
	mężczyźni	870
	bez dysleksji rozwojowej	825
	z dysleksją rozwojową	117

* Dane w tabeli dotyczą tegorocznych absolwentów.

Z egzaminu zwolniono 1 osobę – finalistę Olimpiady Informatycznej.

¹ Załącznik nr 2 do rozporządzenia Ministra Edukacji Narodowej z dnia 20 marca 2020 r. w sprawie szczególnych rozwiązań w okresie czasowego ograniczenia funkcjonowania jednostek systemu oświaty w związku z zapobieganiem, przeciwdziałaniem i zwalczaniem COVID-19 (Dz.U. poz.493, z późn. zm.).

TABELA 2. ZDAJĄCY ROZWIĄZUJĄCY ZADANIA W ARKUSZACH DOSTOSOWANYCH

Zdający rozwiązujący zadania w arkuszach dostosowanych	z autyzmem, w tym z zespołem Aspergera	7
	słabowidzący	0
	niewidomi	0
	słabosłyszący	1
	niestyszący	0
	z niepełnosprawnością ruchową spowodowaną mózgowym porażeniem dziecięcym	1
	Ogółem	9

Przebieg egzaminu

TABELA 3. INFORMACJE DOTYCZĄCE PRZEBIEGU EGZAMINU

Termin egzaminu		19 maja 2021	
Czas trwania egzaminu dla arkusza standardowego		210 minut	
Liczba szkół		123	
Liczba zespołów egzaminatorów		1	
Liczba egzaminatorów		13	
Liczba obserwatorów ² (§ 8 ust. 1)		0	
Liczba unieważnień ³	w przypadku:		
	art. 44zzv pkt 1	stwierdzenia niesamodzielnego rozwiązywania zadań przez zdającego	0
	art. 44zzv pkt 2	wniesienia lub korzystania przez zdającego w sali egzaminacyjnej z urządzenia telekomunikacyjnego	0
	art. 44zzv pkt 3	zakłócenia przez zdającego prawidłowego przebiegu egzaminu	0
	art. 44zzw ust. 1	stwierdzenia podczas sprawdzania pracy niesamodzielnego rozwiązywania zadań przez zdającego	0
	art. 44zzy ust. 7	stwierdzenie naruszenia przepisów dotyczących przeprowadzenia egzaminu maturalnego	0
	art. 44zzy ust. 10	niemożność ustalenia wyniku (np. zaginięcie karty odpowiedzi)	0
Liczba wglądów ³ (art. 44zzz)		17	

² Rozporządzenie 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.).

³ Ustawa o systemie oświaty (Dz.U. z 2020 r. poz. 1327, ze zm.).

Podstawowe dane statystyczne

Wyniki zdających

WYKRES 1. ROZKŁAD WYNIKÓW ZDAJĄCYCH

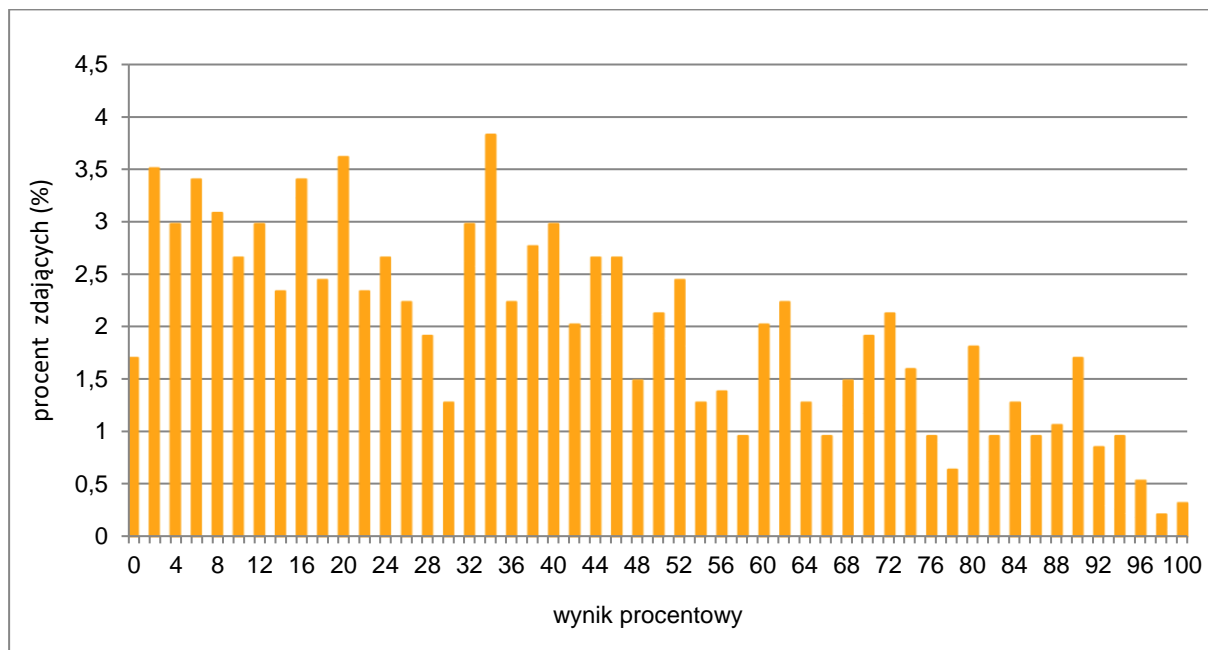


TABELA 4. WYNIKI ZDAJĄCYCH – PARAMETRY STATYSTYCZNE*

Zdający	Liczba zdających	Minimum (%)	Maksimum (%)	Mediana (%)	Modalna (%)	Średnia (%)	Odchylenie standardowe (%)
ogółem	942	0	100	36	34	39	27
w tym:							
z liceów ogólnokształcących	379	0	100	27	27	27	27
z techników	563	0	100	25	25	25	25

* Dane dotyczą wszystkich tegorocznych absolwentów. Parametry statystyczne są podane dla grup liczących 30 lub więcej zdających.

Poziom wykonania zadań

TABELA 5. POZIOM WYKONANIA ZADAŃ

Nr zad.	Wymagania egzaminacyjne 2021		Poziom wykonania zadania (%)
	Wymagania ogólne	Wymagania szczegółowe	
1.1.	III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego.	4. 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) posługuje się podstawowymi technikami algorytmicznymi.	57
1.2.	III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego.	4. 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; 3) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 4) posługuje się podstawowymi technikami algorytmicznymi; 5) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania; 6) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 10) opisuje podstawowe algorytmy i stosuje: a) algorytmy na liczbach całkowitych, np.: – reprezentacja liczb w dowolnym systemie pozycyjnym [...].	29
2.1.	III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego.	4. 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) posługuje się podstawowymi technikami algorytmicznymi; 15) opisuje własności algorytmów na podstawie ich analizy.	46

2.2.	III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego.	4. 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) posługuje się podstawowymi technikami algorytmicznymi; 15) opisuje własności algorytmów na podstawie ich analizy.	35
2.3.	III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego.	4. 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) posługuje się podstawowymi technikami algorytmicznymi; 15) opisuje własności algorytmów na podstawie ich analizy.	41
3.1.	III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego.	4. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin, 8) stosuje rekurencję w prostych sytuacjach problemowych.	69
3.2.	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.	1. Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej. Zdający: 1) przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb [...].	48
3.3.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł [...].	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Zdający: 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL).	47
4.1.	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera,	4. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Zdający:	41

	z zastosowaniem podejścia algorytmicznego.	<p>1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 3) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 4) posługuje się podstawowymi technikami algorytmicznymi; 5) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania; 6) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 10) opisuje podstawowe algorytmy i stosuje: a) algorytmy na tekstach [...]; 21) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 23) dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; 24) 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>4. 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) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 4) posługuje się podstawowymi technikami algorytmicznymi; 5) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania; 6) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 10) opisuje podstawowe algorytmy i stosuje: a) algorytmy na tekstach [...]; 21) 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>	30

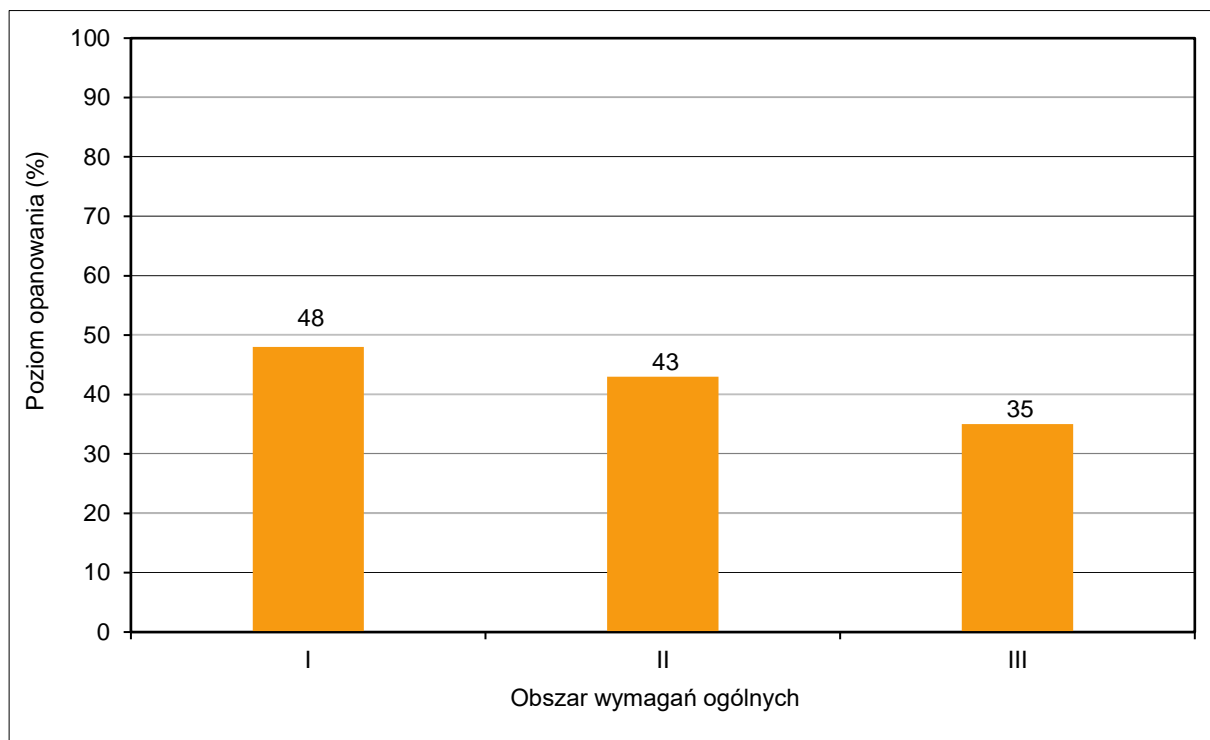
		<p>23) dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania;</p> <p>24) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.</p>	
4.3.	<p>III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.</p>	<p>4. 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) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji;</p> <p>4) posługuje się podstawowymi technikami algorytmicznymi;</p> <p>5) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;</p> <p>6) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;</p> <p>10) opisuje podstawowe algorytmy i stosuje:</p> <p>a) algorytmy na tekstach [...];</p> <p>21) 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>23) dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania;</p> <p>24) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.</p>	26
4.4.	<p>III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.</p>	<p>4. 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) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji;</p> <p>4) posługuje się podstawowymi technikami algorytmicznymi;</p> <p>5) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;</p> <p>6) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania</p>	21

		<p>problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;</p> <p>10) opisuje podstawowe algorytmy i stosuje:</p> <p>a) algorytmy na tekstach [...];</p> <p>21) 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>23) dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania;</p> <p>24) 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: tekstów, danych liczbowych [...].</p>	<p>3. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych [...]. Zdający:</p> <p>1) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. PP 3.3) gromadzi w tabeli arkusza kalkulacyjnego dane pochodzące np. z internetu, stosuje zaawansowane formatowanie tabeli arkusza, dobiera odpowiednie wykresy do zaprezentowania danych.</p>	52
5.2.	<p>II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: tekstów, danych liczbowych [...].</p>	<p>3. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych [...]. Zdający:</p> <p>1) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. PP 3.3) gromadzi w tabeli arkusza kalkulacyjnego dane pochodzące np. z internetu, stosuje zaawansowane formatowanie tabeli arkusza, dobiera odpowiednie wykresy do zaprezentowania danych.</p>	66
5.3.	<p>II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: tekstów, danych liczbowych [...].</p>	<p>3. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych [...]. Zdający:</p> <p>1) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. PP 3.3) gromadzi w tabeli arkusza kalkulacyjnego dane pochodzące np. z internetu, stosuje zaawansowane formatowanie tabeli arkusza, dobiera odpowiednie wykresy do zaprezentowania danych.</p>	58
5.4.	<p>II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: tekstów, danych liczbowych [...].</p>	<p>3. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych [...]. Zdający:</p> <p>1) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. PP 3.3) gromadzi w tabeli arkusza kalkulacyjnego dane pochodzące np. z internetu, stosuje zaawansowane formatowanie tabeli arkusza, dobiera odpowiednie wykresy do zaprezentowania danych.</p>	42

5.5.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych [...].	3. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych [...]. Zdający: 1) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. PP 3.3) gromadzi w tabeli arkusza kalkulacyjnego dane pochodzące np. z internetu, stosuje zaawansowane formatowanie tabeli arkusza, dobiera odpowiednie wykresy do zaprezentowania danych.	27
6.1.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych [...].	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.	59
6.2.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych [...].	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.	46
6.3.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych [...].	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.	54
6.4.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.	31

	komputera: rysunków, tekstów, danych liczbowych [...].	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 [...].	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.	12

WYKRES 2. POZIOM WYKONANIA ZADAŃ W OBSZARZE WYMAGAŃ OGÓLNYCH



Komentarz – na podstawie wyników zdających w kraju

Analiza jakościowa zadań

Do egzaminu maturalnego z informatyki na poziomie rozszerzonym w 2021 roku przystąpiło 8965 abiturientów liceów ogólnokształcących i techników tj. 3,3% ogółu zdających egzamin maturalny. Średni wynik egzaminu z informatyki to 38% (48% dla absolwentów liceów ogólnokształcących i 29% dla absolwentów techników). Wykorzystany do przeprowadzenia egzaminu maturalnego zestaw zadań okazał się dla zdających trudny. Pomimo wyniku wyższego niż w roku poprzednim, tylko 46% maturzystów, którzy przystąpili do tego egzaminu, uzyskało wynik wyższy od wyniku średniego.

Podobnie jak w latach poprzednich, zadania egzaminacyjne sprawdzały wiadomości i umiejętności ujęte w trzech obszarach wymagań ogólnych:

- I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych (zadanie 3.2.).
- II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych (zadania: 3.3., 5.1., 5.2., 5.3., 5.4., 5.5., 6.1., 6.2., 6.3., 6.4., 6.5.).
- III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego (zadania: 1.1., 1.2., 2.1., 2.2., 2.3., 3.1., 4.1., 4.2., 4.3., 4.4.).

Arkusze obu części egzaminu zawierały po 3 wiązki. W pierwszej części zestawu egzaminacyjnego znajdowały się dwie wiązki (zadanie 1. i 2.), złożone z kilku zadań, które wymagały od zdającego:

- znajomości technik algorytmicznych i algorytmów,
- znajomości i rozumienia zgodności algorytmu ze specyfikacją,
- analizy algorytmu i liczby operacji w nim wykonywanych,
- wykorzystania klasycznych algorytmów do rozwiązania prostych zadań.

Ostatnią wiązkę w arkuszu I tworzyły trzy odrębne zadania zamknięte, w których należało określić wartość logiczną każdego z trzech zdań w nim podanych. Zadania tej wiązki sprawdzały znajomość zagadnień z zakresu:

- analizy działania algorytmu dla wskazanych danych,
- zasad konwersji liczb pomiędzy różnymi systemami pozycyjnymi,
- metod wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL).

Zadania z drugiej części egzaminu na poziomie rozszerzonym wymagały od rozwiązujących wiedzy i umiejętności z zakresu:

- modelowania zjawisk i procesów z różnych dziedzin życia,
- wykorzystania metod informatyki do rozwiązywania problemów,

- doboru właściwego programu (użytkowego lub własnoręcznie napisanego) do rozwiązywanego zadania,
- projektowania i przeprowadzania wszystkich etapów na drodze do otrzymania informatycznego rozwiązania problemu, m.in. posługiwania się arkuszem kalkulacyjnym w celu zobrazowania graficznie informacji adekwatnie do jej charakteru,
- analizy problemu i zbioru danych, którego rozwiązanie wymaga zaprojektowania i utworzenia relacyjnej bazy danych (tabeli i relacji między nimi) z uwzględnieniem zawartych informacji,
- wyszukiwania informacji w relacyjnej bazie danych z zastosowaniem różnych technik (w tym zapytania) oraz metod optymalizujących wyszukiwanie.

W tegorocznym arkuszu egzaminacyjnym, na poziomie całego kraju, nie było dla zdających ani zadań bardzo łatwych, ani łatwych. Największą grupę – 63,6% – stanowią zadania trudne (14 zadań). Jedno zadanie było bardzo trudne (4,6%), a 31,8% stanowią zadania umiarkowanie trudne. Najtrudniejsze dla tegorocznych maturzystów okazały się zadania reprezentujące III obszar wymagań ogólnych - poziom wykonania zadań w tym obszarze wynosi 35%. Zadania z pozostałych obszarów okazały się nieco łatwiejsze. Poziom wykonania zadań z obszaru I wymagań ogólnych to 47%, a z obszaru II – 41%.

Spośród zadań tworzących tegoroczny zestaw egzaminacyjny z informatyki najtrudniejszym dla maturzystów było zadanie 6.5. (poziom wykonania 11%), a najłatwiejszym (poziom wykonania 67%) zadanie 3.3. typu *prawda-falsz*.

Pierwszą wiązkę z arkusza I tworzyły dwa zadania (zadanie 1.1. i 1.2), w rozwiązaniu których należało wykorzystać podaną definicję cyfrowego dopełnienia nieujemnej liczby całkowitej. Podana we wstępie do zadań definicja mówiła o tym, że cyfrowym dopełnieniem liczby n nazywamy liczbę całkowitą d , której zapis dziesiętny otrzymujemy z zapisu dziesiętnego liczby n przez zamianę każdej cyfry tego zapisu na cyfrę, będącą jej uzupełnieniem do 9.

W zadaniu 1.1. należało podać czterocyfrową liczbę n taką, że wartość bezwzględna różnicy liczby n i jej cyfrowego dopełnienia d jest: a) najmniejsza, b) największa. Poprawnymi odpowiedziami dla podpunktu a) są liczby: $n=5000$ lub $n=4999$, natomiast dla podpunktu b) – $n=8999$ lub $n=1000$. Zadanie to okazało się dla tegorocznych maturzystów umiarkowanie trudne (poziom wykonania 56%). Analiza rozwiązań wskazuje na to, że łatwiejsze było dla zdających wskazanie największej liczby spełniającej warunki zadania niż najmniejszej.

Drugie zadanie (1.2.) z tej wiązki polegało na napisaniu algorytmu, który dla dodatniej liczby całkowitej n obliczy jej cyfrowe dopełnienie d , przy założeniu, że najbardziej znacząca cyfra liczby n jest większa od 0 i mniejsza od 9. Poprawne algorytmy zawierają przykłady 1. i 2.

Zadanie 1.2. było najtrudniejszym (poziom wykonania 29%) i najczęściej opuszczanym przez zdających zadaniem w arkuszu I zestawu egzaminacyjnego. Błędne rozwiązania tego zadania w przypadku części zdających wynikały z niedokładnego przeczytania treści i ograniczenia rozwiązania tylko do liczb czterocyfrowych (tak jak w zadaniu 1.1.).

Dla $n = 0$ tablica T jest pusta (nie zawiera żadnego elementu).

Wykonaj analizę poniżej zapisanej funkcji $d(x)$, która rozszerza tablicę T o liczbę całkowitą x , a następnie przeprowadza pewną reorganizację zawartości tej tablicy.

$d(x)$:

$n \leftarrow n + 1$

$T[n] \leftarrow x$

$s \leftarrow n$

dopóki $((s \text{ div } 2) \geq 1)$ **oraz** $(T[s] > T[s \text{ div } 2])$ **wykonuj**

$pom \leftarrow T[s]$

$T[s] \leftarrow T[s \text{ div } 2]$

$T[s \text{ div } 2] \leftarrow pom$

$s \leftarrow s \text{ div } 2$

Uwaga: w tym zadaniu przyjmujemy, że:

- tablica T może być powiększana;
- jeśli wartość lewego argumentu operatora **oraz** jest równa *falsz*, to wartość prawego argumentu nie jest wyliczana;
- *div* jest operatorem oznaczającym część całkowitą z dzielenia.

W zadaniu 2.1. należało podać wynik działania funkcji dla podanych zawartości T i z podanym parametrem x . Zadanie 2.2. wymagało od rozwiązujących podania zawartości tablicy T po wykonaniu kolejnych sześciu wywołań funkcji d z podanymi parametrami (6, -4, 12, 27, 26, 8) przy początkowo pustej tablicy T – to zadanie okazało się najtrudniejsze w tej wiązce zadań (poziom wykonania 33%). Zadanie 2.3. polegało na analizie działania fragmentu algorytmu i podaniu liczby operacji w nim wykonywanych przy konkretnych założeniach.

Ostatnia wiązka pierwszej części arkusz egzaminacyjny to trzy zadania typu *prawda-falsz*:

- Zadanie 3.1. dotyczące analizy działania funkcji rekurencyjnej (poziom wykonania 67%),
- Zadanie 3.2., które wymagało od maturzystów umiejętności przedstawiania liczb w różnych systemach pozycyjnych (poziom wykonania 47%),
- Zadanie 3.3., w którym należało wykazać się znajomością języka SQL i umiejętnością wykorzystania go do wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (poziom wykonania 45%).

Zadanie 3.1. okazało się dla zdających zadaniem najłatwiejszym nie tylko w I części egzaminu, ale także w całym zestawie egzaminacyjnym.

Arkusz II zestawu egzaminacyjnego również tworzyły trzy wiązki zadań. W tej części egzaminu, przy rozwiązywaniu zadań, zdający korzystali z zestawów komputerowych, wyposażonych w zadeklarowane przez nich oprogramowanie.

W skład pierwszej wiązki arkusza II wchodziły cztery zadania (4.1 – 4.4.). Każde z nich wymagało napisania w zadeklarowanym języku programowania programu, z wykorzystaniem

danych zapisanych w pliku tekstowym *instrukcje.txt*. Zdający mieli do dyspozycji plik przykładowy oraz podaną dla niego odpowiedź do każdego z wymienionych zadań. Plik przykładowy ułatwiał testowanie pisanych rozwiązań. Wszystkie zadania z tej wiązki okazały się dla zdających trudne i należały do najczęściej pomijanych przez zdających.

Wynikiem działania programu, który miał generować rozwiązanie problemu postawionego w zadaniu 4.1., była długość napisu otrzymanego po wykonaniu wszystkich instrukcji z pliku *instrukcje.txt*. Umiejętność napisania programu, który symuluje pracę automatu tworzącego napis z liter alfabetu (bez polskich znaków), zgodnie z instrukcjami zapisanymi w pliku, oraz obliczenie długości powstałego w ten sposób napisu, okazała się najlepiej opanowaną (poziom wykonania 39%) przez tegorocznych maturzystów spośród umiejętności sprawdzanych poprzez zadania tej wiązki.

W zadaniu 4.2. (poziom wykonania 30%) należało napisać program, który w pliku *instrukcje.txt* znajdzie najdłuższy ciąg występujących kolejno po sobie instrukcji tego samego rodzaju. Jako wynik działania programu powinien być podany rodzaj instrukcji oraz liczba jej wystąpień w znalezionym ciągu.

Zadanie 4.3. (poziom wykonania 27%) wymagało napisania programu, który podałyby literę najczęściej dopisywaną, czyli najczęściej występującą w instrukcji DOPISZ, oraz liczbę jej wystąpień.

Jako rozwiązanie zadania 4.4. (poziom wykonania 22%) zdający mieli podać napis, który powstanie po wykonaniu wszystkich instrukcji z pliku *instrukcje.txt*. Poprawne rozwiązanie tego zadania wymagało od maturzystów prawidłowego oprogramowania działania wszystkich czterech rodzajów instrukcji występujących w pliku *instrukcje.txt*. Niewielu maturzystów poradziło sobie z tym zadaniem. Wśród rozwiązań pojawiły się takie, w których zdający popełniali błąd przy definiowaniu jednej lub więcej niż jednej instrukcji zapisanych w pliku. Na przykład oprogramowując instrukcję PRZESUN dokonywali przesunięcia wszystkich wystąpień podanej litery, a nie tylko jej pierwszego wystąpienia.

Niektórzy zdający, pomimo wyraźnego polecenia napisania programu, rozwiązywali zadania z wiązki 4.1.- 4.4. za pomocą np. arkusza kalkulacyjnego. Zgodnie z zasadami oceniania, w takim przypadku, uzyskiwali 0 punktów, gdyż nie było właściwej realizacji komputerowej rozwiązania.

Drugą i trzecią wiązkę zadań arkusza II części egzaminu stanowiły zadania, które maturzyści mogli rozwiązać z wykorzystaniem, dostępnych narzędzi informatycznych, to jest arkusza kalkulacyjnego, bazy danych lub napisanego przez siebie programu.

Rozwiązując zadania 5.1. – 5.5. zdający mogli wykazać się umiejętnościami wykorzystania typowego programu użytkowego w celu wykonania obliczeń przy pomocy wbudowanych funkcji i zaprojektowanych formuł. W rozwiązaniu zadań maturzyści wykorzystywali dane z pliku *wodociągi.txt*, który zawierał informacje, zebrane przez wodociągi miejskie, o poborze wody przez wszystkich klientów za rok 2019. W pliku zapisano informacje dotyczące gospodarstwa domowego każdego klienta: dziesięcioznakowy kod klienta oraz 12 liczb całkowitych oznaczających ilości zużytej wody w m³ przez kolejnych 12 miesięcy. Kod klienta składał się z pięciocyfrowego numeru klienta, dwucyfrowej liczby oznaczającej liczbę osób pozostających we wspólnym gospodarstwie domowym oraz trzyliterowego kodu dzielnicy miasta.

W zadaniu 5.1. (poziom wykonania 50%), jako rozwiązanie, należało podać zestawienie zawierające pięciocyfrowe numery 10 klientów, którzy w ciągu roku zużyli w swoim gospodarstwie domowym średnio najwięcej wody na jedną osobę, oraz ich średnie zużycie wody na jedną osobę, w zaokrągleniu do dwóch miejsc po przecinku. Zestawienie, zawierające numery klientów i średnie zużycie wody na jedną osobę, powinno być uporządkowane nierosnąco według średniej. Zadanie to okazało się dla zdających umiarkowanie trudne.

By z danych w pliku uzyskać numery klientów, liczbę osób mieszkających w danym gospodarstwie domowym oraz zużycie wody na jedną osobę, w zaokrągleniu do dwóch miejsc po przecinku, zdający wykorzystywali wbudowane funkcje arkusza kalkulacyjnego (przykład 3., przykład 4., przykład 5.).

Przykład 3.

N2 X ✓ f_x =LEWY(A2;5)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	KodKlienta	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	Kod	Ilość osób
2	0793503REM	7	7	9	12	12	15	19	13	8	8	5	9	07935	3
3	0064504WIL	9	9	11	14	12	19	24	17	12	13	10	9	00645	4
4	0809004PRA	10	9	9	14	15	19	22	15	12	13	11	9	08090	4
5	0508003URU	7	8	7	8	11	14	19	13	9	8	7	9	05080	3
6	0573803WLO	7	6	7	12	12	13	16	13	10	9	6	7	05738	3

Przykład 4.

O2 X ✓ f_x =WARTOŚĆ.LICZBOWA(PRAWY(LEWY(A2;7);2))

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	KodKlienta	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	Kod	Ilość osób	średnia	
2	0793503REM	7	7	9	12	12	15	19	13	8	8	5	9	07935	3	38,33	
3	0064504WIL	9	9	11	14	12	19	24	17	12	13	10	9	00645	4	37,50	
4	0809004PRA	10	9	9	14	15	19	22	15	12	13	11	9	08090	4	37,25	
5	0508003URU	7	8	7	8	11	14	19	13	9	8	7	9	05080	3	37,00	

Przykład 5.

P2 X ✓ f_x =ZAKR(SUMA(B2:M2)/O2;2)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	KodKlienta	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	Kod	Ilość osób	średnia	
2	0793503REM	7	7	9	12	12	15	19	13	8	8	5	9	07935	3	41,33	
3	0508003URU	7	8	7	8	11	14	19	13	9	8	7	9	05080	3	40,00	
4	0064504WIL	9	9	11	14	12	19	24	17	12	13	10	9	00645	4	39,75	
5	0809004PRA	10	9	9	14	15	19	22	15	12	13	11	9	08090	4	39,50	
6	0573803WLO	7	6	7	12	12	13	16	13	10	9	6	7	05738	3	39,33	
7	0834905WVES	12	12	12	15	17	25	29	19	17	14	11	13	08349	5	39,20	
8	0885004BIA	8	8	9	14	15	18	22	16	14	13	9	10	08850	4	39,00	
9	0220204REM	9	10	11	14	12	19	19	18	14	11	7	9	02202	4	38,25	
10	0946804PRA	10	7	4	10	15	16	24	17	14	14	10	10	09468	4	37,75	
11	0686604REM	5	8	9	13	13	18	19	17	14	14	9	8	06866	4	36,75	

Z rozwiązaniem tego zadania poradził sobie co drugi maturzysta. Część maturzystów, podając rozwiązanie tego zadania w pliku tekstowym, ograniczała je tylko do pięciu zamiast dziesięciu pierwszych klientów. Popelniany w ten sposób błąd najprawdopodobniej wynikał z niedokładnego przeczytania polecenia zadania.

Zadanie 5.2. (poziom wykonania 61%) – najłatwiejsze w tej wiązce – wymagało od zdających utworzenia zestawienia, zawierającego dane o całkowitym rocznym zużyciu wody dla każdej dzielnicy przez wszystkich mieszkańców. Do wyłuskania nazwy każdej z dzielnic, zapisanych w kodach klientów, zdający mogli wykorzystać np. wbudowaną funkcję arkusza

kalkulacyjnego, działającą na tekstach, umożliwiającą wycięcie z danego tekstu ciągu znaków o określonej długości, licząc od końca tekstu (przykład 6.).

Przykład 6.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	KodKlienta	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	dziel	wod	
2	0000104WIL	10	10	5	8	14	18	21	17	10	9	4	10	WIL	136	
3	0000201ZOL	2	2	1	3	3	4	4	4	2	2	2	2	ZOL	31	
4	0000302MOK	4	4	2	6	5	7	10	8	4	6	3	3	MOK	62	
5	0000402MOK	3	3	3	5	4	7	9	7	4	6	2	2	MOK	55	
6	0000503BIE	7	3	7	8	7	10	13	10	6	9	7	7	BIE	94	
7	0000601BIA	1	1	2	3	2	3	5	3	2	3	2	2	BIA	29	
8	0000703WIL	3	7	6	10	6	10	13	11	9	6	4	7	WIL	92	
9	0000804BIE	7	7	7	8	12	18	19	13	14	9	9	4	BIE	127	
10	0000902OCH	3	2	4	6	5	6	10	8	6	6	4	3	OCH	63	

Po otrzymaniu nazwy dzielnicy oraz obliczeniu sumy zużycia wody dla każdego gospodarstwa, rozwiązanie zadania zdający uzyskiwali najczęściej poprzez wykorzystanie tabeli przestawnej.

Przykład 7.

	A	B	C	D	E
1					
2					
3	Etykiety wierszy	Suma z wod			
4	BEM	54080			
5	BIA	61614			
6	BIE	56368			
7	MOK	55889			
8	OCH	59273			
9	PRA	57241			
10	REM	58971			
11	SRO	58124			
12	TAR	60234			
13	URU	59597			
14	URY	50116			
15	WAW	57674			
16	WES	60372			
17	WIL	55476			
18	WLO	66372			
19	WOL	60523			
20	ZOL	62312			
21	Suma końcowa	994236			
22					
23					

Jako rozwiązanie zadania 5.3. (poziom wykonania 51%) zdający mieli podać maksymalne miesięczne zużycie wody w każdej z dzielnic. By uzyskać szukane wartości zdający, tak jak w zadaniu 5.2., wykorzystywali najczęściej tabelę przestawną, obliczając zużycie wody w każdym miesiącu łącznie przez wszystkich mieszkańców danej dzielnicy, a następnie wbudowaną funkcję arkusza kalkulacyjnego *max()* do wyznaczenia najwyższej wartości zużycia wody. Część zdających jako odpowiedź podawała pełne zestawienie zużycia wody w każdym miesiącu dla każdej z dzielnic, zamiast tylko wartości maksymalnych. W takim wypadku, zgodnie z zasadami oceniania, zdający uzyskiwali 1 punkt za to zadanie. Zadanie 5.3. okazało się dla tegorocznych maturzystów zadaniem umiarkowanie trudnym.

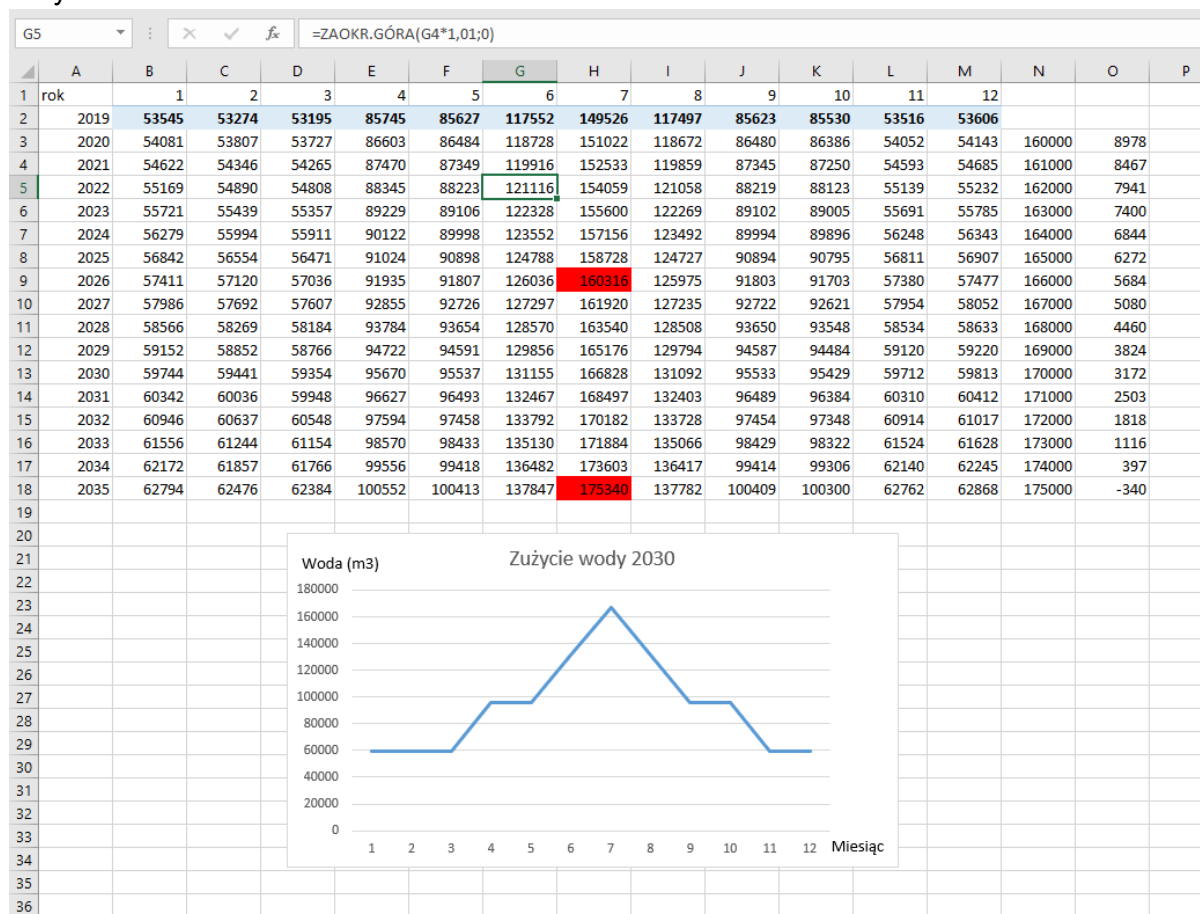
Rozwiązanie zadania 5.4. (poziom wykonania 43%) polegało na przeprowadzeniu symulacji obrazującej przewidywane zużycie wody w każdym z kolejnych miesięcy od stycznia 2020

roku do grudnia 2030 roku. W oparciu o obliczenia zdający mieli podać rok i miesiąc, w którym pierwszy raz zabraknie wody w mieście (przewidywane zużycie będzie większe niż maksymalny przepływ sieci) oraz sporządzić wykres liniowy, obrazujący przewidywane zużycie wody w każdym z kolejnych miesięcy w 2030 roku. W symulacji jako podstawę obliczeń należało uwzględnić obecny maksymalny miesięczny przepływ (wydajność sieci, 160 000 m³), sumaryczne zużycie wody w każdym z 12 miesięcy oraz przyjąć założenie, że sumaryczne miesięczne zużycie wody będzie rosło o 1% rok do roku każdego miesiąca (w m³ z zaokrągleniem w górę do najbliższej liczby całkowitej).

Natomiast w zadaniu 5.5. (poziom wykonania 26%) należało podać rok i miesiąc, kiedy pierwszy raz zabraknie wody w mieście po uwzględnieniu inwestycji, która począwszy od 2021 roku corocznie w styczniu pozwoli na zwiększanie maksymalnego przepływu o 1000 m³.

Zdający często łączyli realizację komputerową rozwiązania zadań 5.4. - 5.5.w jednym arkuszu. Przykład 8. przedstawia właśnie takie rozwiązanie.

Przykład 8.



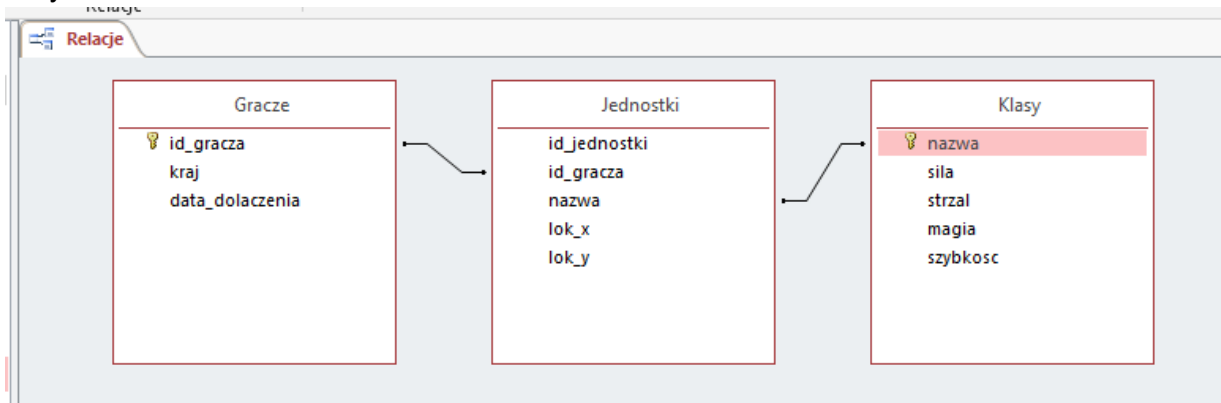
Zadanie 5.5. okazało się dla tegorocznych maturzystów jednym z trzech najtrudniejszych zadań w całym zestawie egzaminacyjnym.

Ostatnią wiązkę zadań arkusza II tworzą zadania, w rozwiązaniu których należało wykorzystać dostępne narzędzia informatyczne oraz dane opisujące aktualny stan wirtualnej planszy gry internetowej *Bitwa o Kamienną Bramę*, zawarte w trzech plikach: *gracze.txt*, *klasy.txt* oraz *jednostki.txt*. Zdający, rozwiązując zadania (6.1. – 6.5) z tej wiązki, najczęściej

korzystali z narzędzi bazodanowych – programu bazodanowego (np. MS Access, Apache OpenOffice Base, LibreOffice Base). Część zdających wykorzystywała strukturalny język programowania. Zdarzali się również maturzyści, którzy próbowali znaleźć odpowiedź na postawione pytania z wykorzystaniem arkusza kalkulacyjnego lub własnego programu, napisanego w wybranym przez siebie języku programowania.

Wykonując zadanie w programie bazodanowym, należało, przed przystąpieniem do realizacji zadań 6.1. – 6.5., najpierw zaimportować dane z plików tekstowych odpowiednio połączyć tabele relacjami, a następnie odpowiedzieć na pytania, tworząc odpowiednie kwerendy. Poniżej przykład widoku relacji ukazujący powiązania między tabelami (Przykład 9.).

Przykład 9.



Rozwiązanie zadania 6.1. (poziom wykonania 55%) polegało na podaniu nazw 5 krajów, z których najwięcej graczy dołączyło do gry w 2018 roku. Oprócz nazwy należało w odpowiedzi również uwzględnić liczbę graczy, którzy dołączyli do gry w tym roku. Odpowiedź zdający uzyskiwali na podstawie danych zawartych w jednej tabeli (*gracze*) poprzez zliczenie zgrupowanych danych w polu *kraj* z uwzględnieniem warunku dotyczącego roku 2018 (wykorzystanie wbudowanej funkcji `Year([data_dolaczenia])`).

Przykład 10.

Pole:	kraj	kraj	Year([data_dolaczenia])
Tabela:	Gracze	Gracze	
Suma:	Grupuj według	Policz	Grupuj według
Sortuj:		Malejąco	
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:			2018
lub:			

Odpowiednik w języku SQL.

```

Zadanie6_1
SELECT TOP 5 Gracze.kraj, Count(Gracze.kraj) AS PoliczOfkraj, Year([data_dolaczenia])
FROM Gracze
GROUP BY Gracze.kraj, Year([data_dolaczenia])
HAVING (((Year([data_dolaczenia]))=2018))
ORDER BY Count(Gracze.kraj) DESC;
    
```

Część maturzystów rozwiązując to zadanie zapomniała o uwzględnieniu w kwerendzie ograniczenia do roku 2018 i podawała zestawienie największej liczby graczy z poszczególnych państw. Za takie rozwiązanie zdający uzyskiwali 1 z 2 punktów.

Przykład 11

The screenshot shows a database management tool interface. At the top, there is a tab labeled 'Zadanie6_1'. Below it, a table structure for 'Gracze' is displayed with columns: id_gracza (primary key), kraj, and data_dolaczenia. Below the table structure, there is a query configuration window with the following settings:

Pole:	kraj	kraj
Tabela:	Gracze	Gracze
Suma:	Grupuj według	Policz
Sortuj:		Malejąco
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:		
lub:		

Zadanie to okazało się najłatwiejszym dla zdających podejmujących się rozwiązywania zadań w oparciu o zbudowaną bazę danych. Co drugi maturzysta przystępujący do egzaminu maturalnego z informatyki poradził sobie z tym zadaniem.

W zadaniu 6.2. (poziom wykonania 44%) jako rozwiązanie należało podać sumę wartości pola strzał (*strzał*) dla każdej klasy jednostek, które mają „elf” jako część nazwy. Odpowiedź uzyskiwało się budując kwerendę na podstawie trzech tabel, z zapisaniem warunku dotyczącego części nazwy oraz zsumowania pogrupowanych wartości według pola *nazwa*.

Przykład 12.

The screenshot shows a database query tool interface. At the top, there is a schema diagram with three tables: **Gracze**, **Jednostki**, and **Klasy**. **Gracze** has fields: `id_gracza` (primary key), `kraj`, and `data_dolaczenia`. **Jednostki** has fields: `id_jednostki` (primary key), `id_gracza`, `nazwa`, `lok_x`, and `lok_y`. **Klasy** has fields: `nazwa` (primary key), `sila`, `strzal`, `magia`, and `szybkosc`. Relationships are shown as 1 to many between Gracze and Jednostki, and many to 1 between Jednostki and Klasy.

Below the schema is a query editor. The 'Pole:' field contains `nazwa`. The 'Tabela:' field contains `Jednostki`. The 'Suma:' field contains `Grupuj według`. The 'Sortuj:' field is empty. The 'Pokaż:' field has a checked checkbox. The 'Kryteria:' field contains `Like '*elf*'`. The 'lub:' field is empty. Below the editor is a table with columns for 'nazwa', 'strzal', and other fields, with checkboxes in the 'Pokaż:' row.

Analiza rozwiązań pozwala zauważyć, że o ile wyszukanie nazw klasy jednostek zawierających w nazwie słowo „elf” nie stanowiło problemu dla zdających, o tyle obliczenie poprawnej sumy wartości pola `strzał` było zadaniem trudnym do wykonania. Najczęściej błąd wynikał z uwzględnienia tylko jednej tabeli (klasy) w tym zapytaniu.

Przykład 13.

The screenshot shows a database query tool interface. The 'Zadanie6_2' window displays a query for the **Klasy** table. The 'Pole:' field contains `nazwa`. The 'Tabela:' field contains `Klasy`. The 'Suma:' field contains `Grupuj według`. The 'Sortuj:' field is empty. The 'Pokaż:' field has a checked checkbox. The 'Kryteria:' field contains `Like '*elf*'`. The 'lub:' field is empty. Below the editor is a table with columns for 'nazwa' and 'strzal', with checkboxes in the 'Pokaż:' row.

Jako rozwiązanie zadania 6.3. (poziom wykonania 50%) należało podać w porządku rosnącym numery graczy, którzy nie mają artylerzystów (jednostek o nazwie artylerzysty). Najczęściej zdający rozwiązywali to zadanie tworząc zapytanie pomocnicze, które wyszukiwało graczy mających artylerzystów, a następnie w tabeli `gracze` wyszukiwano uczestników gry, których numery nie wystąpiły w wyniku kwerendy pomocniczej. Poniżej przykład takiego rozwiązania.

Przykład 14.

Kwerenda pomocnicza - wyszukanie graczy posiadających artylerzystów

Pole:	nazwa	id_gracza		
Tabela:	Jednostki	Jednostki		
Suma:	Grupuj według	Grupuj według		
Sortuj:				
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kryteria:	"artylerzysta"			
lub:				

Kwerenda końcowa

Pole:	id_gracza			
Tabela:	Gracze			
Sortuj:				
Pokaż:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kryteria:	Not In ([Zadanie6_3pom].[id_gracza])			
lub:				

Zadanie 6.4. okazało się dla zdających trudne (poziom wykonania 27%). Rozwiązanie tego zadania polegało na wyszukaniu jednostek, które mogą w jednej turze dojść do *Bramy*, położonej w miejscu o współrzędnych (100,100) i podzieleniu ich na poszczególne klasy. Jako odpowiedź należało podać zestawienie, które dla każdej klasy poda jej nazwę oraz liczbę jednostek z tej klasy mogących w jednej turze osiągnąć *Bramę*. Zdający otrzymali pomocniczą informację, że jednostka w ciągu jednej tury może przemieścić się z punktu (x,y) do punktu (x1,y1), jeśli $|x - x1| + |y - y1| \leq \text{szybkosc}$.

Rozwiązując to zadanie maturzyści najczęściej dzielili rozwiązanie na kilka etapów, tworząc kilka kwerend, np. najpierw obliczali liczbę kroków, jaką mogą wykonać poszczególne jednostki w jednej turze, następnie sprawdzali stosunek tej wartości do *szybkości*, a w końcowym kroku grupowali i zliczali wystąpienia jednostki spełniającej warunek określony we wcześniejszej kwerendzie. Poniżej przykład opisanego rozwiązania.

Przykład 15.

Pole:	id_jednostki	Wyr1: Abs(100-[Jednostki]![lok_x])	Wyr2: Abs(100-[Jednostki]![lok_y])	Wyr3: [Wyr1]+[Wyr2]	nazwa
Tabela:	Jednostki				Jednostki
Sortuj:					
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:					
lub:					

Pole:	Wyr4: [Wyr3] <= [Klasy]![szybkosc]	nazwa	id_jednostki
Tabela:		Zadanie6_4pomoc	Zadanie6_4pomoc
Sortuj:			
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:	-1		
lub:			

Pole:	nazwa	nazwa
Tabela:	Zadanie6_4pomoc2	Zadanie6_4pomoc2
Suma:	Grupuj według	Policz
Sortuj:		
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:		
lub:		

Niektórzy zdający posłużyli się przy rozwiązywaniu tego zadania arkuszem kalkulacyjnym i jego funkcjami. W poniższym przykładzie zdający początkowo stworzył kwerendę w bazie danych, wyznaczając jednostki, dla których spełniony jest warunek $|x-x_1| + |y-y_1| \leq \text{szybkość}$. Następnie wyeksportował otrzymany wynik kwerendy do arkusza kalkulacyjnego, w którym zastosował sumy częściowe do uzyskania wyniku końcowego.

Przykład 16.

zad64pom

Diagram showing two tables: **Jednostki** and **Klasy**.

Jednostki attributes: id_jednostki (PK), id_gracza, nazwa, lok_x, lok_y.

Klasy attributes: nazwa (PK), sila, strzal, magia, szybkoosc.

Relationship: 1 to many (∞).

Criteria: Pole: Wyr1: Abs(100-[lok_x])+Abs(100-[lok_y]); Tabela: Wyr1; Sortuj: ; Pokaż: ; Kryteria: <=[Klasy].[szybkosc]; lub: ;

Pole:	Wyr1: Abs(100-[lok_x])+Abs(100-[lok_y])	szybkosc	nazwa
Tabela:		Klasy	Klasy
Sortuj:			Rosnąco
Pokaż:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kryteria:	<=[Klasy].[szybkosc]		
lub:			

C3 =SUMY.CZĘŚCIOWE(3;C2:C2)

	A	B	C
1	Wyr1	szybkosc	nazwa
2	6	8	architekt
3		architekt Liczba	1
4	8	10	artylerzysta
5	6	10	artylerzysta
6	7	10	artylerzysta
7	3	10	artylerzysta
8		artylerzysta Liczba	4
9	8	8	balista
10	8	8	balista
11		balista Liczba	2
12	12	12	ciemny elf
13		ciemny elf Liczba	1

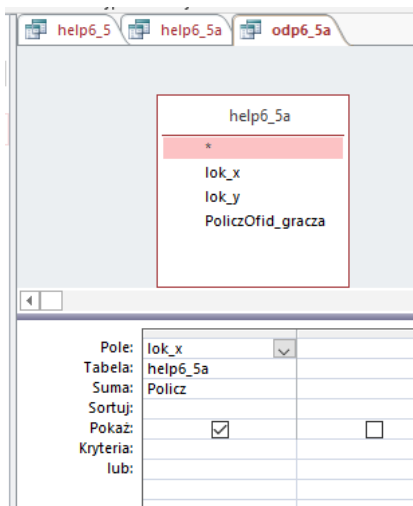
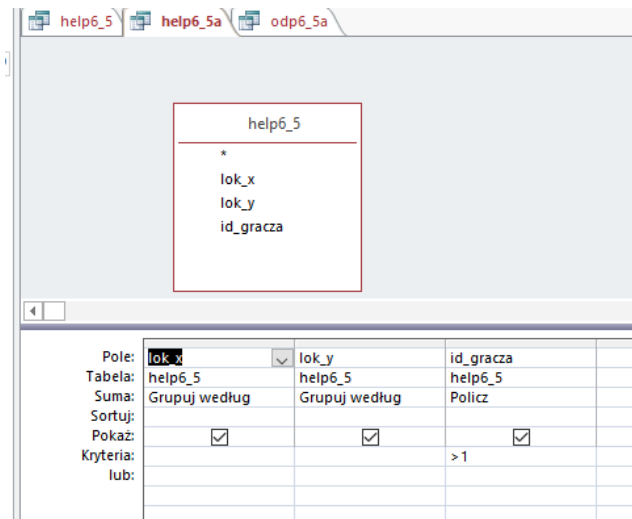
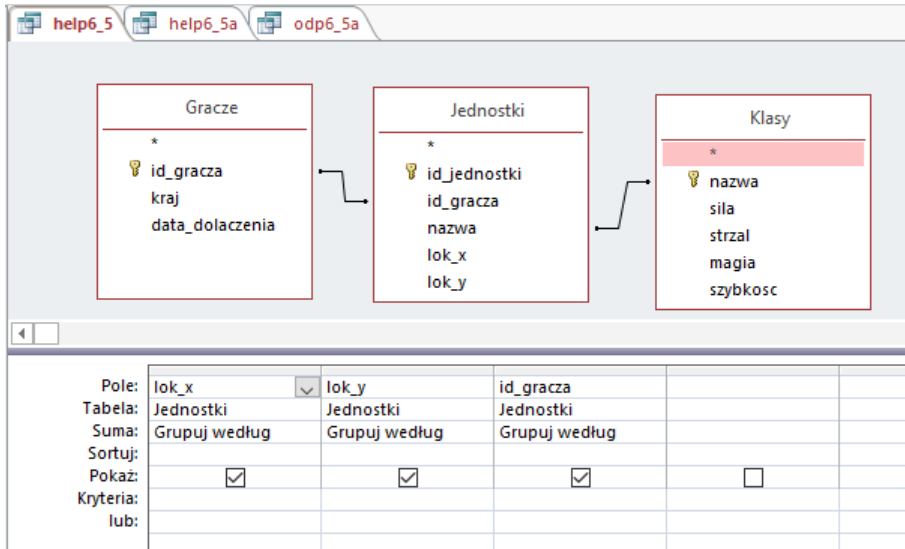
Zdecydowana większość zdających nie potrafiła właściwie zinterpretować i zapisać wszystkich warunków określonych w treści zadania, co skutkowało błędnymi rozwiązaniami.

Dla tegorocznych maturzystów najtrudniejszym zadaniem w całym zestawie egzaminacyjnym było zadanie 6.5. (poziom wykonania 11% - zadanie bardzo trudne), które, tak jak wszystkie zadania z wiązki 6.1. – 6.4., sprawdzało umiejętność wyszukiwania i przetwarzania informacji w relacyjnych bazach danych. W zadaniu 6.5. zdający mieli obliczyć, ile bitew ma miejsce na planszy oraz podać, w ilu bitwach biorą udział gracze z Polski. W rozwiązaniu należało wziąć pod uwagę, że w jednej lokalizacji może się znajdować więcej niż jedna jednostka tego samego gracza. Rozwiązując to zadanie zdający dzielili rozwiązanie na kilka etapów, tworząc kilka kwerend. Poniższy przykład prezentuje

rozwiązanie w kilku kwerendach, oddzielnie dla części pierwszej (liczba bitew na planszy) i części drugiej (liczba bitew, w których biorą udział Polacy).

Przykład 17.

Podpunkt a)



Podpunkt b)

Pole:	lok_x	lok_y	id_gracza		
Tabela:	Jednostki	Jednostki	Jednostki		
Suma:	Grupuj według	Grupuj według	Grupuj według		
Sortuj:					
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kryteria:					
lub:					

Pole:	lok_x	lok_y	
Tabela:	help6_5a	help6_5a	
Suma:	Grupuj według	Grupuj według	
Sortuj:			
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Kryteria:			
lub:			

Pole:	lok_x
Tabela:	help6_5b
Suma:	Policz
Sortuj:	
Pokaż:	<input checked="" type="checkbox"/>
Kryteria:	
lub:	

Zdający przy rozwiązywaniu tego zadania korzystali najczęściej z własności relacyjnych baz danych lub arkusza kalkulacyjnego. Wśród zdających byli również tacy, którzy problem postawiony w tym zadaniu rozwiązywali z pomocą napisanego przez siebie programu. Przykład takiego programu poniżej.

Przykład 18.

```

#include<bits/stdc++.h>
using namespace std;

typedef long long ll;
string gracze[230];
ll ilosc[230][230];
ll gracz_pierwszy[230][230];
ll Polacy = 0;
ll bitwy = 0;
bool bitwa[230][230];
bool bitwa_Polacy[230][230];
bool czy_zawodnik[230][230][230];
ifstream fin("./Dane_pomocnicze/graczy.txt");
ifstream fin1("./Dane_pomocnicze/jednostki.txt");

inline void debug()
{
    ll pomoc = 0;
    for(int i = 0; i < 210; i++)
        for(int j = 0; j < 210; j++)
            if(ilosc[i][j] > 1)
                pomoc++;
    ll pom1 = 0;
    for(int i = 0; i < 210; i++)
        for(int j = 0; j < 210; j++)
            if(bitwa[i][j])
                pom1++;
    cout << " *" << pomoc << " " << pom1 << "* ";
}
inline void init()
{
    for(int i = 0; i < 200; i++)
    {
        ll a;
        string s;
        fin >> a >> s;
        gracze[a] = s;
    }
    for(int i = 0; i < 10001; i++)
    {
        ll id_gracza, x,y;
        fin1 >> id_gracza >> x >> y;
        if(czy_zawodnik[x][y][id_gracza] == false)
            ilosc[x][y]++;
        if(ilosc[x][y] == 1)
        {
            gracz_pierwszy[x][y] = id_gracza;
            czy_zawodnik[x][y][id_gracza] = true;
        }
        if(ilosc[x][y] > 1 and czy_zawodnik[x][y][id_gracza] == false)
        {
            czy_zawodnik[x][y][id_gracza] = true;
        }
    }
}

```

```

        if(bitwa[x][y] == false)
        {
            bitwa[x][y] = true;
            bitwy++;
        }
        if(bitwa_Polacy[x][y] == false and (gracze[id_gracza] == "Polska"
or gracze[gracz_pierwszy[x][y]] == "Polska"))
        {
            bitwa_Polacy[x][y] = true;
            Polacy++;
        }
    }
}
//debug();
}
inline void rozwiazanie2()
{
    for(int i = 0; i < 200; i++)
    {
        ll a;
        string s;
        fin >> a >> s;
        gracze[a] = s;
    }
    for(int i = 0; i < 10001; i++)
    {
        ll id_gracza, x,y;
        fin1 >> id_gracza >> x >> y;
        ilosc[x][y]++;
        if(ilosc[x][y] == 1)
            gracz_pierwszy[x][y]=id_gracza,czy_zawodnik[x][y][id_gracza]=true;
        if(ilosc[x][y] > 1 and czy_zawodnik[x][y][id_gracza] == false)
        {
            czy_zawodnik[x][y][id_gracza] = true;
            if(bitwa[x][y] == false)
                bitwy++, bitwa[x][y] = true;
            if((gracze[id_gracza] == "Polska" or
gracze[gracz_pierwszy[x][y]] == "Polska") and bitwa_Polacy[x][y] == false)
                Polacy++, bitwa_Polacy[x][y] = true;
        }
    }
    debug();
}
int main()
{
    init();
    cout << Polacy << " " << bitwy;
}

```

Rozwiązując zadanie 6.5. zdający najczęściej popełniali błędy polegające na: braku uwzględnienia faktu, że w tej samej lokalizacji może być więcej niż jedna jednostka jednej osoby, obliczaniu liczby bitew wyłącznie z udziałem Polaków, policzenia dwukrotnie tej samej bitwy (jeśli wzięło w niej udział dwóch Polaków).

Wnioski i rekomendacje

1. Na podstawie analizy wyników egzaminu maturalnego z informatyki w roku 2021 można stwierdzić, że zadania tworzące tegoroczny zestaw egzaminacyjny były dla zdających trudne. Zróżnicowanie pod względem trudności występuje w każdej z wiązek wchodzących w skład arkusza egzaminacyjnego. Podobnie jak w roku poprzednim, zdający mieli problem z rozwiązaniem zadań z zakresu opracowywania i przetwarzania informacji z różnych źródeł oraz przeprowadzania analizy statystycznej danych w arkuszu kalkulacyjnym lub bazie danych. Tegoroczni maturzyści potrafili wykorzystać możliwości arkusza kalkulacyjnego lub relacyjnej bazy danych w prostych i niezłożonych zadaniach, jednak zbudowanie złożonej kwerendy lub wykonanie symulacji w arkuszu kalkulacyjnym, uwzględniającej kilka warunków stanowiło dla zdających nadal duży problem.
2. Podobnie jak w latach poprzednich, spośród zadań występujących w arkuszu egzaminacyjnym, zdający najczęściej nie podejmowali próby rozwiązania zadań wymagających napisania algorytmu lub programu realizującego rozwiązanie problemu postawionego w treści zadania. By maturzyści nie bali się rozwiązywania tego typu zadań, wskazane jest zwiększenie podczas zajęć lekcyjnych liczby ćwiczeń wymagających wykorzystania w praktyce ujętych w podstawie programowej algorytmów. W przypadku ćwiczeń z rozwiązywania zadań z części pierwszej egzaminu warto zwrócić uwagę na poprawny zapisu algorytmów w pseudojęzyku programowania.
3. Analiza rozwiązań zdających uwidacznia problem związany z pobieżnym, nieuważnym czytaniem treści poleceń występujących w zadaniach. Skutkiem tego często są nieprawidłowe odpowiedzi lub rozwiązania całych zadań, nieuwzględniające wszystkich warunków podanych w zadaniu. Należy zwrócić uwagę uczniów podczas zajęć, by uważnie czytali/analizowali treść zadań, a przy udzielaniu odpowiedzi sprawdzali, czy ich rozwiązanie jest rzeczywiście rozwiązaniem postawionego w zadaniu problemu, czy wynik podany jest z dokładnością żądaną w zadaniu, a np. typ wykonanego wykresu odpowiada wskazanemu w treści zadania.
4. Istotną rolę osób przygotowujących młodzież do egzaminu maturalnego z informatyki jest zwrócenie uwagi uczniów, przyszłych maturzystów, na zapisywanie plików z realizacjami komputerowymi rozwiązań zadań z drugiej części egzaminu. Realizacja komputerowa rozwiązania stanowi integralną część rozwiązania zadania, wymaganą przy sprawdzaniu rozwiązania przez egzaminatora. Brak plików z realizacją komputerową rozwiązania lub brak części obliczeń np. z powodu nadpisania obliczeń poprzednich skutkuje uzyskaniem przez zdającego 0 punktów za dane zadanie, pomimo zapisania w pliku tekstowym poprawnych odpowiedzi. Warto również zwrócić uwagę zdających na polecenia w treści zadania, które wskazują, z jakich narzędzi można korzystać podczas rozwiązywania danego zadania. Jeżeli polecenie brzmi „napisz program”, to zdający, który rozwiąże to zadanie za pomocą np. arkusza kalkulacyjnego i poda poprawne odpowiedzi otrzyma 0 punktów, gdyż nie zrealizował w sposób właściwy polecenia z zadania.