

<i>Rodzaj dokumentu:</i>	Sprawozdanie za rok 2022 województwo zachodniopomorskie
<i>Egzamin:</i>	Egzamin maturalny
<i>Przedmiot:</i>	Informatyka
<i>Poziom:</i>	Poziom rozszerzony
<i>Termin egzaminu:</i>	20 maja 2022 r.
<i>Data publikacji dokumentu:</i>	19 września 2022 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 zachodniopomorskiego

Okręgowa Komisja Egzaminacyjna w Poznaniu

Izabela Szafrńska
Anna Sperling

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	14
Wnioski i rekomendacje	30

Opis arkusza egzaminu maturalnego

W roku szkolnym 2021/2022 egzamin maturalny z informatyki był 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 20 zadań o zróżnicowanym stopniu trudności, w tym 3 zadania typu prawda-falsz. Pozostałe zadania 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 znajomości podstawowej wiedzy z różnych obszarów informatyki oraz umiejętności: tworzenia i analizy algorytmów, zapisania w postaci programu komputerowego (w wybranym przez siebie języku programowania) z wykorzystaniem plików z danymi rozwiązań postawionych problemów, wykorzystania arkusza kalkulacyjnego, bazy danych lub własnego programu komputerowego do przeprowadzenia symulacji lub analizy danych.

Za rozwiązanie 8 zadań tworzących arkusz I zestawu egzaminacyjnego zdający mógł uzyskać maksymalnie 15 punktów, a za rozwiązanie 12 zadań z arkusza II – 35 punktów. Razem można było otrzymać 50 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		253
Zdający rozwiązujący zadania w arkuszu standardowym	z liceów ogólnokształcących	80
	z techników	173
	z branżowych szkół II stopnia	0
	ze szkół na wsi	0
	ze szkół w miastach do 20 tys. mieszkańców	26
	ze szkół w miastach od 20 tys. do 100 tys. mieszkańców	68
	ze szkół w miastach powyżej 100 tys. mieszkańców	159
	ze szkół publicznych	193
	ze szkół niepublicznych	60
	kobiety	12
	mężczyźni	241
	bez dysleksji rozwojowej	216
	z dysleksją rozwojową	37

* Dane w tabeli dotyczą tegorocznych absolwentów.

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.), w szczególności nowelizacje ww. rozporządzenia z dnia 16 grudnia 2020 r (Dz.U. poz. 2314) oraz z dnia 17 sierpnia 2021 r (Dz.U. poz. 1525).

Z egzaminu zwolniono 2 osoby – laureatów i finalistów Olimpiady Informatycznej.

TABELA 2. ZDAJĄCY ROZWIĄZUJĄCY ZADANIA W ARKUSZACH DOSTOSOWANYCH

Zdający rozwiązujący zadania w arkuszach dostosowanych	z autyzmem, w tym z zespołem Aspergera	4
	słabowidzący	0
	niewidomi	0
	słabosłyszący	0
	niesłyszący	0
	z niepełnosprawnością ruchową spowodowaną mózgowym porażeniem dziecięcym	0
	Ogółem	4

Przebieg egzaminu

TABELA 3. INFORMACJE DOTYCZĄCE PRZEBIEGU EGZAMINU

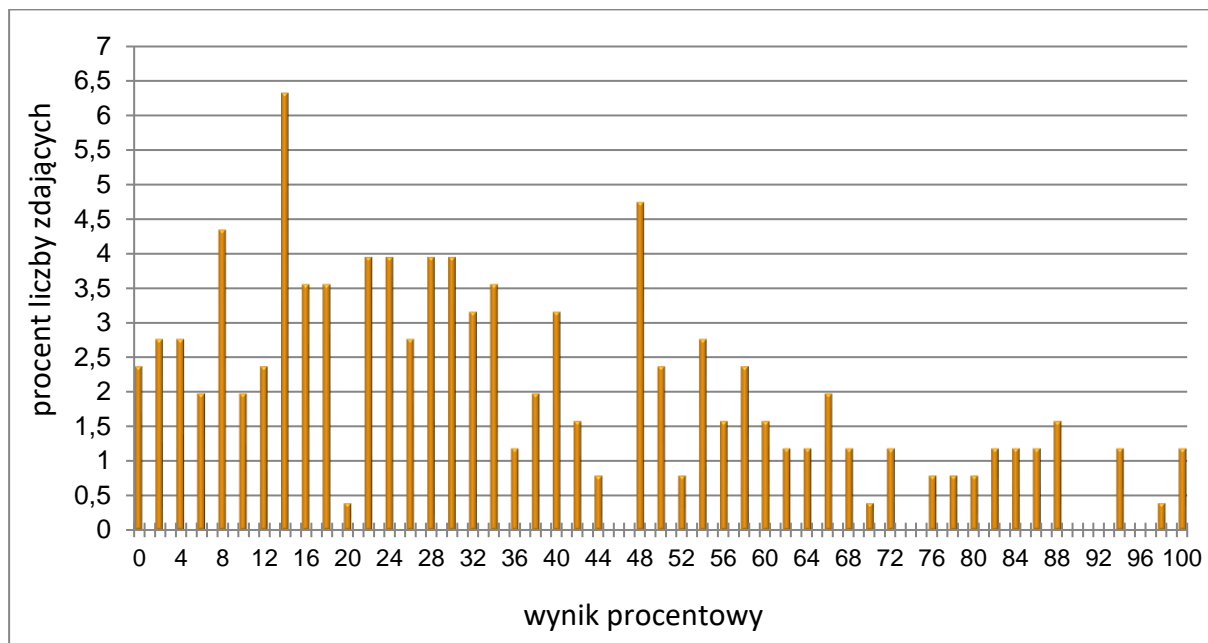
Termin egzaminu		20 maja 2022	
Czas trwania egzaminu dla arkusza standardowego		210 minut	
Liczba szkół		56	
Liczba zespołów egzaminatorów		0	
Liczba egzaminatorów		0	
Liczba obserwatorów ² (§ 8 ust. 1)		1	
Liczba unieważnień ³	w przypadku:		
	art. 44zzv pkt 1	stwierdzenia niesamodzielnego rozwiązywania 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
	inne (np. złe samopoczucie ucznia)		0
Liczba wglądów ³ (art. 44zzz)		1	

² Rozporządzenie Ministra Edukacji Narodowej z dnia 21 grudnia 2016 r. w sprawie szczegółowych warunków i sposobu przeprowadzania egzaminu maturalnego (Dz.U. z 2016 r., poz. 2223, z późn. zm.).

³ Ustawa o systemie oświaty (Dz.U. z 2021 r. poz. 1915, z późn. 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	253	0	100	30	14	36	25
w tym:							
z liceów ogólnokształcących	80	0	100	45	48	45	31
z techników	173	0	88	28	14	32	21

* 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 2022		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.	79
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 [...].	28
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.	54

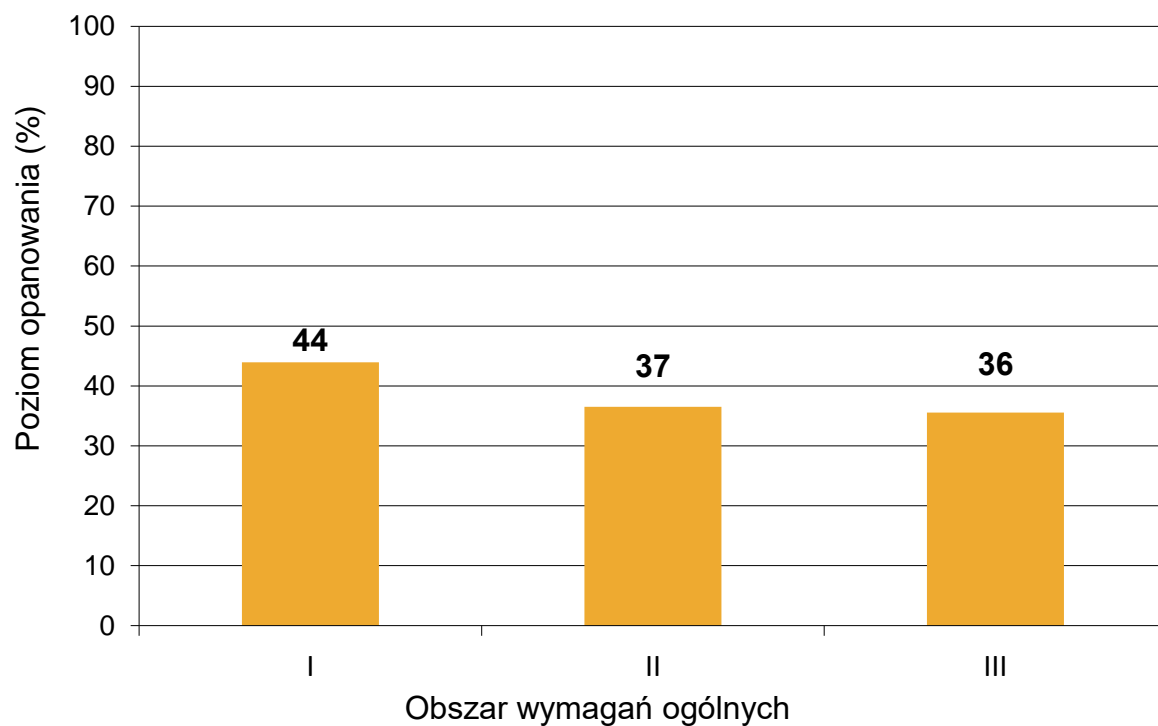
2.2.	III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego.	4. Rozwiązywanie problemów i podejmowanie decyzji [...], stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) posługuje się podstawowymi technikami algorytmicznymi; 15) opisuje własności algorytmów na podstawie ich analizy.	47
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.	27
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, 5) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania.	30
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 [...].	44
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).	61
4.1.	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera,	4. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.	45

	z zastosowaniem podejścia algorytmicznego.	<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) 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: <ol style="list-style-type: none"> a) algorytmy na liczbach całkowitych [...]; 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. 	
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.</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) 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: <ol style="list-style-type: none"> a) algorytmy na liczbach całkowitych [...]; 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; 	21

		23) dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania; 24) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.	
4.3.	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, 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; 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 [...], 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;	17
5.1.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: 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.	83
5.2.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: 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.	42

		PP. 3.3) gromadzi w tabeli arkusza kalkulacyjnego dane pochodzące np. z Internetu, stosuje zaawansowane formatowanie tabeli arkusza, dobiera odpowiednie wykresy do zaprezentowania danych.	
5.3.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: 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.	79
5.4.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: 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.	11
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.	8
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.	47
6.2.	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji,	23

	z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych [...].	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.	
6.3.	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.	29
6.4.	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.	35

WYKRES 2. POZIOM WYKONANIA ZADAŃ W OBSZARZE WYMAGAŃ OGÓLNYCH

Komentarz – na podstawie wyników zdających w kraju

Zadania na egzaminie miały zróżnicowany stopień trudności. Poziom wykonania zadań mieścił się w przedziale od 13% do 83%. Spośród 20 zadań tworzących arkusz trzy zadania należały do kategorii bardzo trudnych (poziom wykonania 13%–19%) oraz trzy zadania do łatwych (poziom wykonania 81%–83%), pozostałe były trudne (poziom wykonania 24%–45%) oraz umiarkowanie trudne (poziom wykonania od 50% do 67%).

W pierwszej części egzaminu zdający mieli do rozwiązania 8 zadań, w tym pierwsze pięć tworzyło dwie wiązki zadań z zakresu tworzenia i analizy algorytmów. Pozostałe trzy zadania to zadania wielokrotnego wyboru typu prawda – fałsz.

Pierwsza wiązka, złożona z dwóch zadań, dotyczyła analizy podanego problemu i zapisu algorytmu w wybranym języku programowania lub pseudokodzie. Dla zdających łatwym okazało się pierwsze z zadań (zadanie 1.1. – poziom wykonania 81%), które było wstępem do zadania 1.2., pomagającym zrozumieć zadany problem. W zadaniu 1.1. należało określić, dla danych w przykładzie ciągów liczbowych, ile należy w każdym z nich wymienić liczb, aby otrzymać ciąg zwany n-permutacją. Definicja n-permutacji podana została na początku zadania.

Zadanie 1.2. wymagało od zdających zapisania algorytmu, generującego jako wynik liczbę liczb, które należy zamienić w danym (za pomocą tablicy) ciągu, aby otrzymać n-permutację. To zadanie było dla zdających trudne – poziom wykonania 31%. Trudności nie sprawiało zrozumienie problemu (o czym świadczą wyniki zadania 1.1.), a sam zapis rozwiązania w postaci algorytmu. Zdający podchodzili do rozwiązania zadania 1.2., najczęściej wybierając jeden z dwóch sposobów. Pierwszy sposób polegał na przeglądaniu kolejnych liczb całkowitych od 1 do n (gdzie n oznacza długość podanego ciągu liczb) i sprawdzaniu, czy każda z nich zawiera się w podanym ciągu (przykład 1).

Przykład 1.

```

k ← 0
dla i = 1, 2, ..., n, wykonuj:
    temp ← 0
    dla j = 1, 2, ..., n, wykonuj:
        jeżeli A[j] ≠ i, to:
            temp ← temp + 1
    jeżeli temp = n, to:
        k ← k + 1
zwróć k

```

Drugi sposób wymagał zliczenia, ile liczb w ciągu jest większych niż n oraz ile jest powtórzeń liczb mniejszych niż n w tym ciągu (Przykład 2).

Przykład 2.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, k=0;
7     cin >> n;
8     int A[n+1];
9     for(int i=1; i<=n; i++) { cin >> A[i];
10         if(A[i] > n) {
11             A[i] = 0;
12             k++;
13         }
14     }
15     for(int i=1; i<=n; i++) {
16         if(A[i] != 0) {
17             for(int j=i+1; j<=n; j++) {
18                 if(A[i] == A[j]) {
19                     A[j] = 0;
20                     k++;
21                 }
22             }
23         }
24     }
25 }
26 cout << k;
27

```

Największy problem sprawiało zdającym zapisanie algorytmu w postaci pseudokodu lub kodu w języku programowania. Należy zwrócić uwagę na to, że zapis algorytmu musiał spełniać wymagania zadania. Niedozwolone było użycie np. wbudowanych funkcji i operatorów istniejących w wybranym języku programowania, innych niż wymienione w zadaniu. Zdający nie zawsze stosowali się do tych wymogów – zdarzały się przypadki użycia np. funkcji sortującej lub operatorów sprawdzających, czy dany element jest w tablicy. W takich przypadkach zdający nie uzyskiwał maksymalnej liczby punktów za rozwiązanie zadania.

Drugą wiązkę arkusza I stanowiły trzy zadania (zadanie 2.1. –2.3.) dotyczące analizy podanego algorytmu. Zadania 2.1. oraz 2.2. okazały się dla zdających umiarkowanie trudne (poziom wykonania odpowiednio 57% i 50%). W pierwszym z nich należało podać wynik działania algorytmu dla podanych słów złożonych z liter *a* i *b* (przez słowo rozumiemy tu dowolny ciąg znaków), w drugim należało podać przykładowe słowa o podanej długości złożone z liter *a* i *b*, dla których algorytm pokaże dany wynik. Trudnym okazało się zadanie 2.3. (poziom wykonania 30%) – wymagało ono dobrego zrozumienia problemu, ponieważ

podanie wyniku dla bardzo długiego słowa złożonego z wielu ciągów liter *a* i *b*, było trudne przy próbach wykonania algorytmu krok po kroku.

Arkusz I zamykały trzy zadania testowe typu prawda – fałsz. Dość trudne okazało się pierwsze z nich (zadanie 3.1. – poziom wykonania 33%), które wymagało znajomości pojęcia złożoności obliczeniowej algorytmu i umiejętności jej określenia na prostym przykładzie. Pozostałe dwa (zadanie 3.2. i 3.3.) były umiarkowanie trudne (poziom wykonania odpowiednio 52% i 67%). Zadanie 3.2. wymagało umiejętności dodania dwóch liczb zapisanych w systemie czwórkowym i zamiany wyniku na zapis binarny, ósemkowy i szesnastkowy. Zadanie 3.3. sprawdzało, czy zdający rozumieją działanie zapytań zapisanych w języku SQL.

Podczas drugiej części egzaminu zdający rozwiązywali zadania z trzech wiązek tworzących arkusz II zestawu egzaminacyjnego. Przy rozwiązywaniu ww. zadań zdający korzystali z zestawów komputerowych, wyposażonych w zadeklarowane przez nich oprogramowanie.

Pierwszą wiązkę arkusza II tworzą trzy zadania (4.1.–4.3.). Rozwiązanie każdego z nich wymagało napisania w zadeklarowanym języku programowania programu, z wykorzystaniem danych zapisanych w pliku tekstowym *liczby.txt*. Do zadań z tej wiązki dołączony był przykładowy plik, który ułatwiał zdającym testowanie napisanych programów, a do każdego z zadań podano odpowiedź, jaką zdający powinien otrzymać w wyniku przeprowadzenia testów poprawności działania swojego programu dla pliku przykładowego. Zadania z tej wiązki okazały się dla zdających trudne i bardzo trudne – należały również do najczęściej pomijanych przez zdających.

Rozwiązanie zadania 4.1. wymagało napisania programu, którego wynikiem miała być liczba takich liczb, których cyfry pierwsza i ostatnia są takie same. W odpowiedzi należało również podać tę liczbę, która spełniała warunek oraz występowała w pliku *liczby.txt* jako pierwsza. Przykładową realizację rozwiązania problemu z zadania 4.1. przedstawia program zawarty w przykładzie 3. Dla ułatwienia zdający, na potrzeby rozwiązania zadania, najczęściej traktowali liczby zapisane w pliku jako łańcuchy znaków.

Przykład 3.

```

1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main()
6  {
7      fstream plik("liczby.txt");
8      string liczba;
9      int ile=0;
10     string pierwsza;
11     for(int i=0;i<200;i++)
12     {
13         plik>>liczba;
14
15         if(liczba[0]==liczba[liczba.length()-1])
16         {
17             ile++;
18             if(ile==1)
19                 pierwsza=liczba;
20         }
21     }
22     cout<<"Zadanie 4.1: "<<ile<<" "<<pierwsza<<endl;
23 }
```


Umiejętność napisania prostego programu, który wyszukuje i zlicza spośród danych zapisanych w pliku liczby spełniające warunki opisane w treści zadania, okazała się najlepiej opanowaną (poziom wykonania – 45%) przez tegorocznych zdających spośród umiejętności sprawdzanych poprzez zadania tej wiązki.

W zadaniu 4.2. (poziom wykonania – 24%) należało napisać program, który wśród liczb zapisanych w pliku *liczby.txt* znajdzie tę, która ma w rozkładzie najwięcej czynników pierwszych, oraz tę, w rozkładzie której występuje najwięcej różnych czynników pierwszych. Przy rozwiązaniu tego zadania zdający wykorzystywali znajomość jednego z podstawowych algorytmów (algorytm rozkładu liczby na czynniki pierwsze), z którym zetknęli się podczas nauki informatyki na lekcjach w szkole. Jedną z takich realizacji jest program, którego fragment jest zawarty w przykładzie 4.

Przykład 4.

```

8      int ma=0, lima;
9      int rma=0, rlima;
10     for(int i=0;i<200;i++)
11     {
12         int a;
13         cin>>a;
14         int b=a;
15         vector<int>czy;
16         int x=2;
17         while(a>1)
18         {
19             if(a%x==0)
20             {
21                 czy.push_back(x);
22                 a=a/x;
23             }
24             else x++;
25         }
26         if(czy.size()>ma)
27         {
28             lima=b;
29             ma=czy.size();
30         }
31         sort(czy.begin(), czy.end());
32         int ile=1, curr=czy[0];
33         for(int y=1;y<czy.size();y++)
34         {
35             if(czy[y]==curr)
36             {
37                 continue;
38             }
39             else
40             {
41                 ile++;
42                 curr=czy[y];
43             }
44         }
45         if(ile>rma)
46         {
47             rma=ile;
48             rlima=b;
49         }
50     }
51     cout<<lima<<" "<<ma<<endl<<rlima<<" "<<rma;
52 }

```

Dla maturzystów, którzy podjęli próbę rozwiązania zadania 4.2., łatwiejszym do realizacji elementem było wyznaczenie liczb, które spełniały warunki zadania, niż podanie liczby ich czynników pierwszych.

Na potrzeby zadania 4.3. zdefiniowano pojęcie *dobrej trójki*. Zgodnie z definicją trójka różnych liczb (x, y, z) jest dobra, jeśli y jest wielokrotnością x , natomiast z jest wielokrotnością y . Następnie poprzez analogię określono właściwości *dobrej piątki*.

Aby rozwiązać zadanie 4.3., należało napisać program (lub programy), które generowały odpowiedzi na pytania postawione w dwóch podpunktach. Pierwszy podpunkt dotyczył liczby dobrych trójek wśród liczb zapisanych w pliku *liczby.txt*. Dodatkowo program miał zapisać wszystkie dobre trójki, każdą w oddzielnym wierszu, w pliku o nazwie *trojki.txt*. Natomiast drugi podpunkt wymagał podania jako odpowiedzi liczby wszystkich dobrych piątek spośród liczb, które wystąpiły w pliku *liczby.txt*.

Przykładowy program, generujący odpowiedzi do podpunktów zadania 4.3. przedstawia poniższy przykład:

Przykład 5.

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main() {
7      int three_count = 0;
8      int five_count = 0;
9      vector<int> list;
10
11     for (int i = 0; i < 200; i++) {
12         int num;
13         cin >> num;
14         list.push_back(num);
15     }
16
17     for (int i = 0; i < list.size(); i++) {
18         int num1 = list[i];
19         for (int j = 0; j < list.size(); j++) {
20             int num2 = list[j];
21             if (num2 > num1 && num2 % num1 == 0) {
22                 for (int k = 0; k < list.size(); k++) {
23                     int num3 = list[k];
24                     if (num3 > num2 && num3 % num2 == 0) {
25                         cout << num1 << " " << num2 << " " << num3 << endl;
26                         three_count++;
27                         for (int l = 0; l < list.size(); l++) {
28                             int num4 = list[l];
29                             if (num4 > num3 && num4 % num3 == 0) {
30                                 for (int m = 0; m < list.size(); m++) {
31                                     int num5 = list[m];
32                                     if (num5 > num4 && num5 % num4 == 0) {
33                                         five_count++;
34                                     }
35                                 }
36                             }
37                         }
38                     }
39                 }
40             }
41         }
42     }
43
44     cout << three_count << endl << five_count << endl;

```

Zadanie 4.3. było jednym z najtrudniejszych spośród wszystkich zadań w całym zestawie egzaminacyjnym (poziom wykonania 19%).

Niektórzy zdający, pomimo wyraźnego polecenia napisania programu, rozwiązywali zadania z wiązki 4.1. –4.3. za pomocą np. arkusza kalkulacyjnego. Zgodnie z zasadami oceniania,

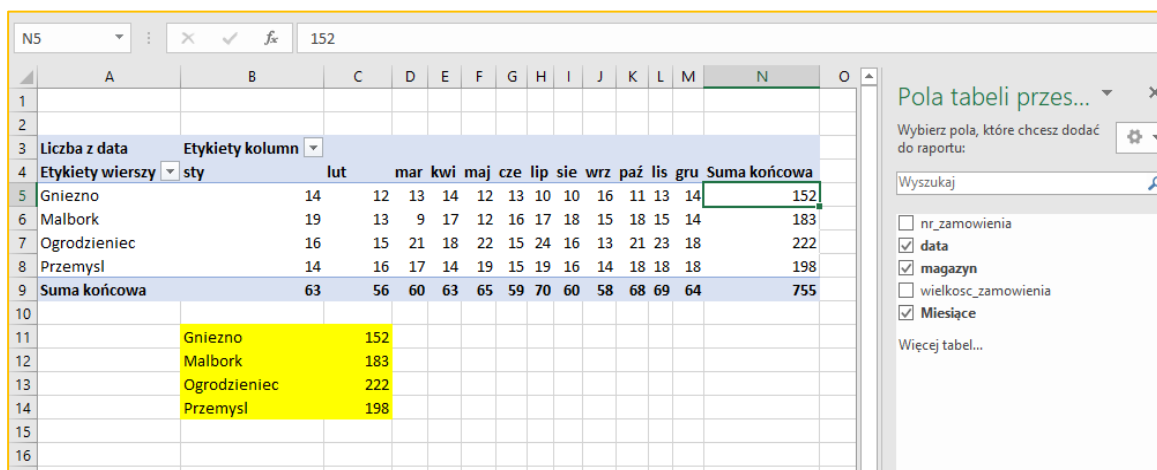
w takim przypadku, uzyskiwali 0 punktów, gdyż nie było właściwej realizacji komputerowej rozwiązania.

Do rozwiązania pozostałych zadań arkusza II części egzaminu, tworzących dwie wiązki, maturzyści mogli wykorzystać dostępne narzędzia informatyczne, to jest arkusz kalkulacyjny, bazę danych lub napisany przez siebie program.

Drugą wiązkę arkusza II części egzaminu tworzy pięć zadań o zróżnicowanym stopniu trudności, przy rozwiązywaniu których zdający mogli wykazać się umiejętnościami wykorzystania typowego programu użytkowego w celu wykonania obliczeń przy pomocy wbudowanych funkcji i zaprojektowanych formuł. W rozwiązaniu zadań 5.1.–5.5. zdający wykorzystywali dane z pliku *soki.txt*, który zawierał informacje o zamówieniach butelkowanego soku owocowego, składanych w pewnym zakładzie przez cztery magazyny. W pliku zapisano informacje dotyczące każdego zamówienia: numer zamówienia, datę złożenia zamówienia, nazwę magazynu, który złożył dane zamówienie oraz wielkość zamówienia (liczbę butelek).

W zadaniu 5.1. należało podać zestawienie zawierające informację o liczbie składanych zamówień przez każdy z magazynów. Rozwiązanie zadania zdający uzyskiwali najczęściej poprzez wykorzystanie tabeli przestawnej (przykład 6).

Przykład 6.



Etykiety wierszy	sty	lut	mar	kwi	maj	cze	lip	sie	wrz	paź	lis	gru	Suma końcowa
Gniezno	14	12	13	14	12	13	10	10	16	11	13	14	152
Malbork	19	13	9	17	12	16	17	18	15	18	15	14	183
Ogrodzieniec	16	15	21	18	22	15	24	16	13	21	23	18	222
Przemysl	14	16	17	14	19	15	19	16	14	18	18	18	198
Suma końcowa	63	56	60	63	65	59	70	60	58	68	69	64	755

Gniezno	152
Malbork	183
Ogrodzieniec	222
Przemysl	198

Zadanie to było dla tegorocznych maturzystów najłatwiejszym spośród wszystkich zadań zestawu egzaminacyjnego (poziom wykonania 83%).

Zadanie 5.2. (poziom wykonania – 45%) wymagało podania liczby dni najdłuższego okresu, kiedy zamówienia z Ogrodzieńca (jeden z magazynów) wpływały codziennie. W odpowiedzi należało również uwzględnić daty: pierwszego i ostatniego dnia tego okresu. By uzyskać wymagane informacje, zdający najczęściej najpierw za pomocą filtra wyszukiwali zamówienia z magazynu w Ogrodzieńcu, a następnie badali różnicę pomiędzy datami kolejnych zamówień z tego magazynu (przykład 7), co umożliwiło wskazanie ciągu następujących po sobie dni, w których były składane zamówienia. Kolejny krok to określenie warunku, zliczającego okres występowania kolejnych dni (dla różnicy równej 1) z wykorzystaniem wbudowanych w arkusz kalkulacyjny funkcji (przykład 8).

Przykład 7.

D161						
=B161-B160						
	A	B	C	D	E	F
1	nr_zamow	data	magazyn	różnica	jeżeli	
158	537	20.09.2021	Ogrodzieniec	3	1	
159	539	21.09.2021	Ogrodzieniec	1	2	
160	547	26.09.2021	Ogrodzieniec	5	1	
161	553	30.09.2021	Ogrodzieniec	4	1	
162	560	03.10.2021	Ogrodzieniec	3	1	
163	564	05.10.2021	Ogrodzieniec	2	1	
164	569	07.10.2021	Ogrodzieniec	2	1	
165	572	08.10.2021	Ogrodzieniec	1	2	
166	574	09.10.2021	Ogrodzieniec	1	3	
167	577	10.10.2021	Ogrodzieniec	1	4	
168	579	11.10.2021	Ogrodzieniec	1	5	
169	580	12.10.2021	Ogrodzieniec	1	6	
170	582	13.10.2021	Ogrodzieniec	1	7	
171	583	14.10.2021	Ogrodzieniec	1	8	
172	591	17.10.2021	Ogrodzieniec	3	1	
173	593	18.10.2021	Ogrodzieniec	1	2	
174	596	20.10.2021	Ogrodzieniec	2	1	
175	597	21.10.2021	Ogrodzieniec	1	2	
176	599	22.10.2021	Ogrodzieniec	1	3	

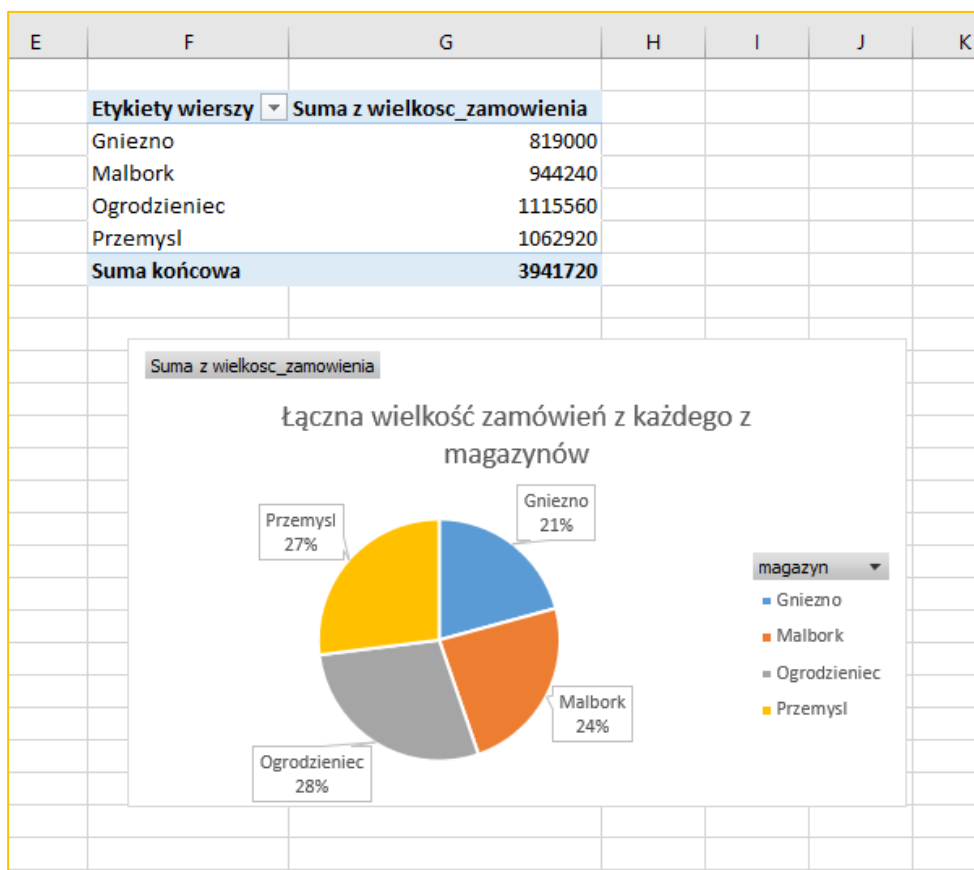
Przykład 8.

E171						
=JEŻELI(D171=1;E170+1;1)						
	A	B	C	D	E	F
1	nr_zamow	data	magazyn	różnica	jeżeli	
158	537	20.09.2021	Ogrodzieniec	3	1	
159	539	21.09.2021	Ogrodzieniec	1	2	
160	547	26.09.2021	Ogrodzieniec	5	1	
161	553	30.09.2021	Ogrodzieniec	4	1	
162	560	03.10.2021	Ogrodzieniec	3	1	
163	564	05.10.2021	Ogrodzieniec	2	1	
164	569	07.10.2021	Ogrodzieniec	2	1	
165	572	08.10.2021	Ogrodzieniec	1	2	
166	574	09.10.2021	Ogrodzieniec	1	3	
167	577	10.10.2021	Ogrodzieniec	1	4	
168	579	11.10.2021	Ogrodzieniec	1	5	
169	580	12.10.2021	Ogrodzieniec	1	6	
170	582	13.10.2021	Ogrodzieniec	1	7	
171	583	14.10.2021	Ogrodzieniec	1	8	
172	591	17.10.2021	Ogrodzieniec	3	1	

Po znalezieniu najdłuższego ciągu następujących po sobie dni zamówień wystarczyło w odpowiedzi podać otrzymaną długość okresu oraz odczytać datę pierwszego i ostatniego zamówienia.

Jako rozwiązanie zadania 5.3. (poziom wykonania – 81%) zdający mieli podać zestawienie zawierające łączną wielkość zamówień z każdego z magazynów oraz na podstawie uzyskanych danych utworzyć procentowy wykres kołowy. By uzyskać szukane wartości zdający, tak jak w zadaniu 5.1., wykorzystywali najczęściej tabelę przestawną, obliczając łączną wielkość zamówień z każdego z magazynów, a następnie wykonywali wykres kołowy (przykład 9).

Przykład 9.



Zadanie 5.3. okazało się dla zdających jednym z trzech najłatwiejszych zadań spośród wszystkich zadań części I i II zestawu egzaminacyjnego.

Dwa ostatnie zadania (zadania 5.4. i 5.5.), wchodzące w skład tej wiązki, wymagały przeprowadzenia symulacji realizacji zamówień przez zakład główny oraz jego filię, przy założeniu warunków podanych we wstępie do tych zadań. Problemy postawione w tych zadaniach okazały się dla zdających najtrudniejszymi do rozwiązania nie tylko wśród zadań tej wiązki, ale również spośród wszystkich zadań w całym zestawie egzaminacyjnym. Zadania tego typu, podobnie jak zadania programistyczne, najczęściej sprawiają trudności zdającym.

Rozwiązanie zadania 5.4. (poziom wykonania – 17%) polegało na znalezieniu daty oraz numeru zamówienia, które jako pierwsze zostało zrealizowane przez filię zakładu głównego. W odpowiedzi należało również uwzględnić liczbę zamówień z całego roku przekazanych do filii oraz łączną liczbę butelek przekazanych do magazynów przez zakład filialny. W symulacji jako podstawę obliczeń należało uwzględnić liczbę butelek znajdującą się w zakładzie głównym w dniu 2 stycznia 2021 r. (30 000 butelek), wysokość produkcji w głównym zakładzie w zależności od dnia tygodnia oraz zasady realizacji zamówień.

Poprawną realizację, uwzględniającą wszystkie warunki zadania, ilustrują przykłady 10–13.

Przykład 10.

F152									
=JEŻELI(LUB([@[dzień tygodnia]]=6;[@[dzień tygodnia]]=7);5000;12000)									
	A	B	C	D	E	F	G	H	I
1	nr_zamowienia	data	magazyn	wielkosc_zamowienia	dzień tygodnia	liczba wyprodukowanych butelek	nowy dzień	główny zakład	po zamówieniu
146	145	15.03.2021	Ogrodzieniec	6860	1	12000	1	17780	10920
147	146	16.03.2021	Ogrodzieniec	2020	2	12000	1	22920	20900
148	147	16.03.2021	Przemysl	3650	2	12000	0	20900	17250
149	148	17.03.2021	Ogrodzieniec	9720	3	12000	1	29250	19530
150	149	18.03.2021	Przemysl	7840	4	12000	1	31530	23690
151	150	18.03.2021	Ogrodzieniec	6780	4	12000	0	23690	16910
152	151	18.03.2021	Gniezno	3490	4	12000	0	16910	13420
153	152	18.03.2021	Malbork	9980	4	12000	0	13420	3440
154	153	19.03.2021	Malbork	7850	5	12000	1	15440	7590
155	154	19.03.2021	Gniezno	9770	5	12000	0	7590	FILIA
156	155	20.03.2021	Gniezno	750	6	5000	1	12590	11840
157	156	20.03.2021	Malbork	8900	6	5000	0	11840	2940
158	157	20.03.2021	Ogrodzieniec	9410	6	5000	0	2940	FILIA
159	158	21.03.2021	Gniezno	9310	7	5000	1	7940	FILIA
160	159	21.03.2021	Ogrodzieniec	2480	7	5000	0	7940	5460

Przykład 11.

H152									
=JEŻELI([@[nowy dzień]]=1;JEŻELI(I151<>"FILIA";I151+[@[liczba wyprodukowanych butelek]]);H151+[@[liczba wyprodukowanych butelek]]);JEŻELI(I151<>"FILIA";I151;H151))									
	A	B	C	D	E	F	G	H	I
1	nr_zamowienia	data	magazyn	wielkosc_zamowienia	dzień tygodnia	liczba wyprodukowanych butelek	nowy dzień	główny zakład	po zamówieniu
146	145	15.03.2021	Ogrodzieniec	6860	1	12000	1	17780	10920
147	146	16.03.2021	Ogrodzieniec	2020	2	12000	1	22920	20900
148	147	16.03.2021	Przemysl	3650	2	12000	0	20900	17250
149	148	17.03.2021	Ogrodzieniec	9720	3	12000	1	29250	19530
150	149	18.03.2021	Przemysl	7840	4	12000	1	31530	23690
151	150	18.03.2021	Ogrodzieniec	6780	4	12000	0	23690	16910
152	151	18.03.2021	Gniezno	3490	4	12000	0	16910	13420
153	152	18.03.2021	Malbork	9980	4	12000	0	13420	3440
154	153	19.03.2021	Malbork	7850	5	12000	1	15440	7590
155	154	19.03.2021	Gniezno	9770	5	12000	0	7590	FILIA
156	155	20.03.2021	Gniezno	750	6	5000	1	12590	11840
157	156	20.03.2021	Malbork	8900	6	5000	0	11840	2940
158	157	20.03.2021	Ogrodzieniec	9410	6	5000	0	2940	FILIA
159	158	21.03.2021	Gniezno	9310	7	5000	1	7940	FILIA
160	159	21.03.2021	Ogrodzieniec	2480	7	5000	0	7940	5460
161	160	21.03.2021	Przemysl	1740	7	5000	0	5460	3720

Przykład 12.

=JEZELI([@główny_zakład]-[@wielkosc_zamowienia]<0;"FILIA";[@główny_zakład]-[@wielkosc_zamowienia])											
	A	B	C	D	E	F	G	H	I	J	K
1	nr_zamowienia	data	magazyn	wielkosc_zamowienia	dzień tygodnia	liczba wyprodukowanych butelek	nowy dzień	główny zakład	po zamówieniu		
146	145	15.03.2021	Ogrodzieniec	6860	1	12000	1	17780	10920		
147	146	16.03.2021	Ogrodzieniec	2020	2	12000	1	22920	20900		
148	147	16.03.2021	Przemysl	3650	2	12000	0	20900	17250		
149	148	17.03.2021	Ogrodzieniec	9720	3	12000	1	29250	19530		
150	149	18.03.2021	Przemysl	7840	4	12000	1	31530	23690		
151	150	18.03.2021	Ogrodzieniec	6780	4	12000	0	23690	16910		
152	151	18.03.2021	Gniezno	3490	4	12000	0	16910	13420		
153	152	18.03.2021	Malbork	9980	4	12000	0	13420	3440		
154	153	19.03.2021	Malbork	7850	5	12000	1	15440	7590		
155	154	19.03.2021	Gniezno	9770	5	12000	0	7590	FILIA		
156	155	20.03.2021	Gniezno	750	6	5000	1	12590	11840		
157	156	20.03.2021	Malbork	8900	6	5000	0	11840	2940		
158	157	20.03.2021	Ogrodzieniec	9410	6	5000	0	2940	FILIA		
159	158	21.03.2021	Gniezno	9310	7	5000	1	7940	FILIA		

Przykład 13.

=SUMA.JEZELI(I:I;"FILIA";D:D)										
	C	D	E	F	G	H	I	J	K	L
	magazyn	wielkosc_zamowienia	dzień tygodnia	liczba wyprodukowanych butelek	nowy dzień	główny zakład	po zamówieniu			
1	Ogrodzieniec	1290	6	5000	1	35000	33710			
1	Przemysl	4420	6	5000	0	33710	29290			
1	Gniezno	5190	6	5000	0	29290	24100			
1	Malbork	950	7	5000	1	29100	28150			
1	Gniezno	6000	7	5000	0	28150	22150			
1	Przemysl	8530	7	5000	0	22150	13620			
1	Malbork	1140	1	12000	1	25620	24480		zamowienia	37
1	Przemysl	2460	1	12000	0	24480	22020		butelki	285230
1	Gniezno	7520	2	12000	1	34020	26500			
1	Przemysl	7920	2	12000	0	26500	18580			
1	Ogrodzieniec	1430	2	12000	0	18580	17150			

Zadanie 5.5. (poziom wykonania – 13%) okazało się dla tegorocznych maturzystów najtrudniejszym zadaniem w całym zestawie egzaminacyjnym. Aby uzyskać maksymalną liczbę punktów za rozwiązanie tego zadania, należało podać, jaką najmniejszą liczbę butelek soku powinien wyprodukować zakład główny w dni robocze, aby mógł zrealizować wszystkie zamówienia samodzielnie. Rozwiązując zadanie 5.5., zdający często modyfikowali rozwiązanie zadania 5.4., poprzez dopasowanie liczby wyprodukowanych butelek (przykład 14.).

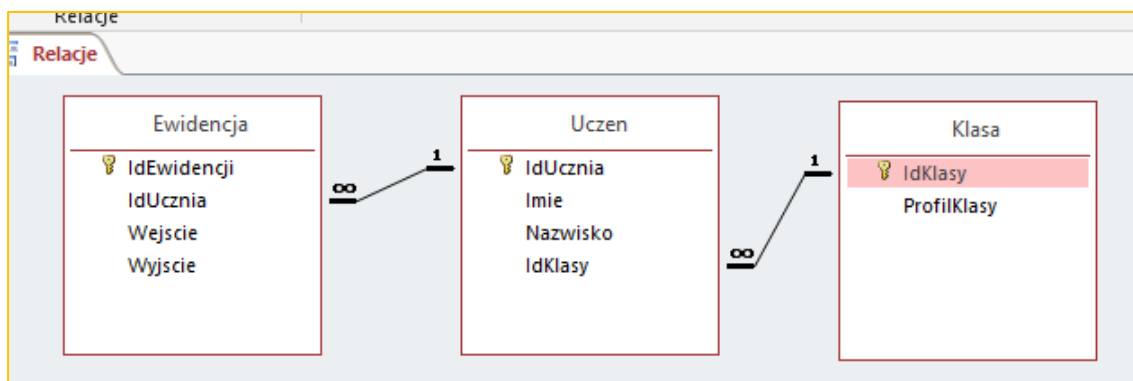
Przykład 14.

=JEŻELI(LUB([@[dzień tygodnia]]=6;[@[dzień tygodnia]]=7);5000;13179)									
C	D	E	F	G	H	I	J	K	L
ryn	wielkosc_zamowienia	dzień tygodnia	liczba wyprodukowanych butelek	nowy dzień	główny zakład	po zamówieniu		zamowienia	
zieniec	1290	6	5000	1	35000	33710		butelki	0
nysl	4420	6	5000	0	33710	29290			
no	5190	6	5000	0	29290	24100			
ork	950	7	5000	1	29100	28150			
no	6000	7	5000	0	28150	22150			
nysl	8530	7	5000	0	22150	13620			
ork	1140	1	13179	1	26799	25659			
nysl	2460	1	13179	0	25659	23199			
no	7520	2	13179	1	36378	28858			
nysl	7920	2	13179	0	28858	20938			
zieniec	1430	2	13179	0	20938	19508			
ork	1500	3	13179	1	32687	31187			
zieniec	5540	3	13179	0	31187	25647			
no	7340	3	13179	0	25647	18307			

Ostatnią wiązkę zadań arkusza II tworzą cztery zadania (6.1.–6.4.), w rozwiązaniu których należało wykorzystać dostępne narzędzia informatyczne oraz dane pochodzące z automatycznego systemu kontroli wejść i wyjść w pewnej szkole, zawarte w trzech plikach: *klasa.txt*, *uczen.txt* oraz *ewidencja.txt*. Zdający, rozwiązując zadania z tej wiązki, najczęściej korzystali z narzędzi bazodanowych – programu bazodanowego (np. MS Access, Apache OpenOffice Base, LibreOffice Base). 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.

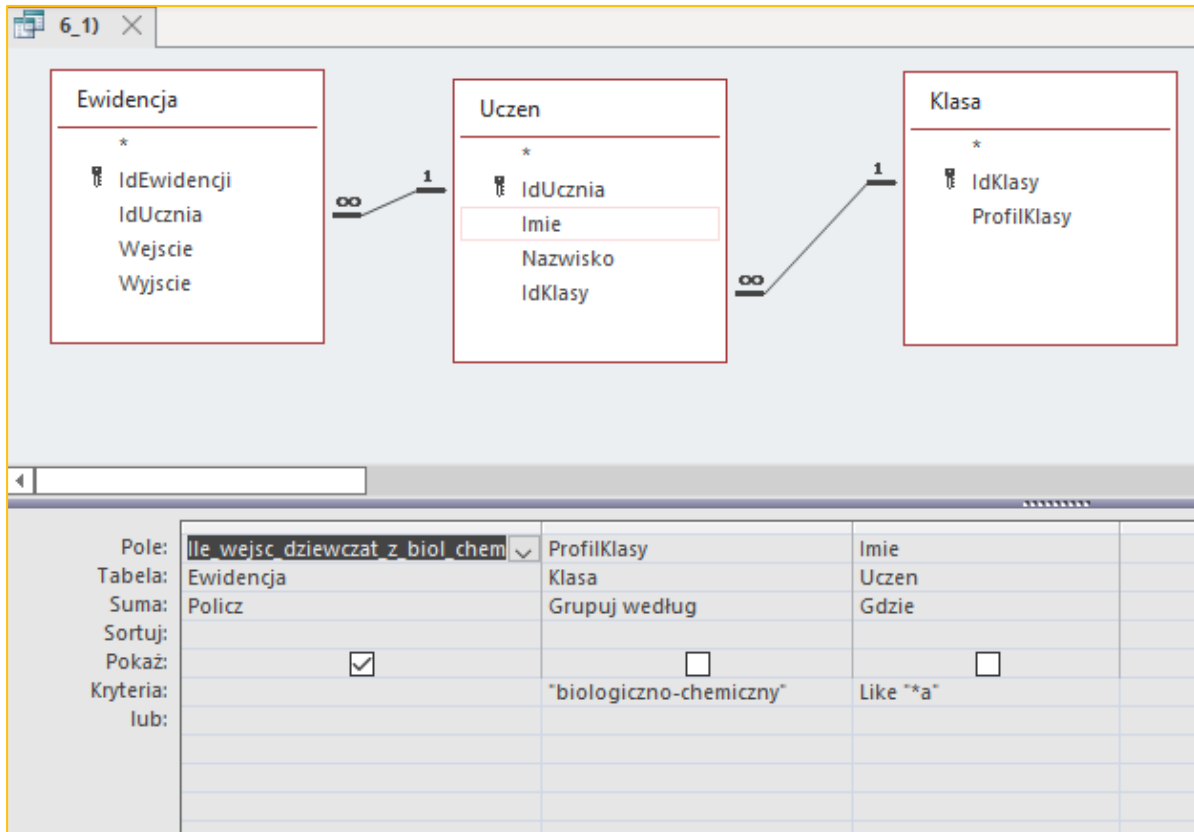
Zadania z tej wiązki w większości okazały się dość trudne dla zdających (poziom wykonania od 29% do 58%).

Rozwiązując zadania w programie bazodanowym, należało, przed przystąpieniem do realizacji zadań 6.1.–6.4., najpierw zaimportować dane z plików tekstowych oraz odpowiednio połączyć tabele relacjami. Następnie należało odpowiedzieć na pytania, tworząc odpowiednie kwerendy. Poniżej przykład widoku relacji ukazujący powiązania między tabelami (przykład 15.).

Przykład 15.

Rozwiązanie zadania 6.1. (poziom wykonania – 58%) polegało na podaniu liczby wejść dziewcząt z klas o profilu biologiczno-chemicznym. Odpowiedź zdający uzyskiwali poprzez zliczenie zgrupowanych danych w polu *ProfilKlasy* z uwzględnieniem warunku dotyczącego ostatniej litery w imieniu uczennic (przykład 16.). To zadanie okazało się najłatwiejszym zadaniem w wiązce.

Przykład 16.



Odpowiednik w języku SQL.

```
SELECT Count(Ewidencja.IdEwidencji) AS
Ile_wejsc_dziewczat_z_biol_chemu
FROM (Klasa INNER JOIN Uczen ON Klasa.IdKlasy = Uczen.IdKlasy) INNER
JOIN Ewidencja ON Uczen.IdUcznia = Ewidencja.IdUcznia
WHERE (((Uczen.Imie) Like "*a"))
GROUP BY Klasa.ProfilKlasy
HAVING (((Klasa.ProfilKlasy)="biologiczno-chemiczny"));
```

Część zdających, by uzyskać rozwiązania najpierw tworzyła kwerendę pomocniczą, która spośród wszystkich uczniów wchodzących do szkoły wyodrębniała uczennice z klas biologiczno-chemicznych. Natomiast zliczenie wejść tych uczennic następowało w drugiej kwerendzie, która wykorzystywała pierwszą jako źródło danych

W zadaniu 6.2. (poziom wykonania – 29%) jako rozwiązanie należało utworzyć zestawienie zawierające informacje o liczbie uczniów, którzy w poszczególnych dniach nie spóźnili się do szkoły. Odpowiedź można było uzyskać budując kwerendę na podstawie jednej tabeli, z zapisaniem warunku dotyczącego daty i godziny wejścia oraz zliczenie wartości według pola *Wejście* (przykład 17. z wykorzystaniem wbudowanych funkcji daty i czasu) lub *iducznia*.

Przykład 17.

Pole:	Dzien_miesiaca: Day[Ewidencja.Wejscie]	Ile_uczniow_nie_spoznilo_sie: IdUcznia	Hour[Ewidencja].[We]	Minute[Ewidencja].[V	Second[Ewidencja].[V
Tabela:	Ewidencja	Ewidencja			
Suma:	Grupuj według	Policz	Gdzie	Gdzie	Gdzie
Sortuj:					
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kryteria:			8	0	0
lub:			<8		

Odpowiednik w języku SQL.

```
SELECT Day(Ewidencja.Wejscie) AS Dzien_miesiaca,
Count(Ewidencja.IdUcznia) AS Ile_uczniow_nie_spoznilo_sie
FROM Ewidencja
WHERE (((Hour([Ewidencja].[Wejscie]))=8) AND
((Minute([Ewidencja].[Wejscie]))=0) AND
((Second([Ewidencja].[Wejscie]))=0)) OR
(((Hour([Ewidencja].[Wejscie]))<8))
GROUP BY Day(Ewidencja.Wejscie);
```

Analiza rozwiązań pozwala zauważyć, że zdający mieli problem z „wyłuskaniem” informacji dotyczących godziny wejścia do szkoły, która była zapisana w jednym polu wraz z datą. By poradzić sobie z tym problemem część zdających traktowała dane w polu *Ewidencja.Wejscie* jako tekstowe i wykorzystywała wbudowane funkcje do odcięcia potrzebnego fragmentu ciągu znaków. W niektórych rozwiązaniach zdający nie uwzględniali właściwej godziny wejścia (nie spóźnili się ci uczniowie, którzy weszli przed godziną 8.00 oraz ci, którzy weszli równo o godzinie 8.00), co skutkowało nieprawidłowymi wartościami w zestawieniu.

Jako rozwiązanie zadania 6.3. (poziom wykonania – 35%) należało podać identyfikatory, imiona i nazwiska trzech osób, które w okresie monitorowanego czasu przebywały najdłużej na terenie szkoły. Przykład 18. zawiera przykładową kwerendę, która generuje odpowiedź na pytanie postawione w tym zadaniu.

Przykład 18.

Pole:	IdUcznia	Imie	Nazwisko	Suma([Ewidencja].[Wyjscie]-[Ewidencja].[Wejscie])
Tabela:	Uczen	Uczen	Uczen	
Suma:	Grupuj według	Grupuj według	Grupuj według	Wyrażenie
Sortuj:				Malejąco
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Kryteria:				
lub:				

Odpowiednik w języku SQL.

```
SELECT TOP 3 Uczn.IdUcznia, Uczn.Imie, Uczn.Nazwisko
FROM Uczn INNER JOIN Ewidencja ON Uczn.IdUcznia = Ewidencja.IdUcznia
GROUP BY Uczn.IdUcznia, Uczn.Imie, Uczn.Nazwisko
ORDER BY Sum([Ewidencja].[Wyjscie]-[Ewidencja].[Wejscie]) DESC;
```

W rozwiązaniu wykorzystano grupowanie danych oraz wbudowaną funkcję umożliwiającą zsumowanie wartości uzyskanych w wyniku zbudowanego wewnątrz tej funkcji wyrażenia.

Zadanie 6.4. także okazało się dla zdających trudne (poziom wykonania – 42%).

Rozwiązanie tego zadania polegało na wyszukaniu imion i nazwisk wszystkich uczniów, którzy byli nieobecni 6.04.2022 r.

Rozwiązując to zadanie, maturzyści najczęściej dzielili rozwiązanie na dwa etapy (dwie kwerendy). Pierwszy etap polegał na utworzeniu kwerendy, która w wyniku dawała informację o wszystkich uczniach obecnych w wyznaczonym dniu. Natomiast druga kwerenda wyszukująca niepasujące dane (poprzez wykorzystanie warunku *Is Null*) sprawdzała, identyfikatory których uczniów nie wystąpiły w wyniku wykonania pierwszej kwerendy. Poniżej przykład opisanego rozwiązania.

Przykład 19, 20. Kwerenda pomocnicza – wyszukanie wszystkich uczniów obecnych w wyznaczonym dniu (dwa sposoby).

Przykład 19.

Pole:	IdUcznia	Wyr1: Left([Ewidencja].[Wejscie];2)
Tabela:	Uczen	
Sortuj:		
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:		Like "06"
lub:		

Przykład 20.

Pole:	Dzien: Day([wejscie])	IdUcznia
Tabela:		Ewidencja
Sortuj:		
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:	6	
lub:		

Kwerenda końcowa (Przykład 21).

Przykład 21.

The screenshot displays a database query interface. At the top, two tables are shown: 'Uczen' and 'zadanie4_pom'. An arrow indicates a join between them. The 'Uczen' table has fields: IdUcznia (primary key), Imie, Nazwisko, and IdKlasy. The 'zadanie4_pom' table has fields: IdUcznia and Wyr1. Below the tables, a query results table is visible with the following structure:

Pole:	Imie	Nazwisko	IdUcznia
Tabela:	Uczen	Uczen	zadanie4_pom
Sortuj:			
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Kryteria:			Is Null
lub:			

Wnioski i rekomendacje

1. Na podstawie analizy wyników egzaminu maturalnego z informatyki w roku 2022 można stwierdzić, że chociaż zadania tworzące tegoroczny zestaw egzaminacyjny były dla zdających trudne – jak w poprzednich latach – to wynik ogólny nieco się poprawił i jest najwyższy od 2015 roku.
2. Zróżnicowanie pod względem trudności zadań występuje w każdej z wiązek wchodzących w skład arkusza egzaminacyjnego. Zdający dobrze sobie radzą z typowymi zadaniami wymagającymi np. użycia arkusza kalkulacyjnego, prostymi zapytaniem w bazie danych, a także analizą ich zapisu w języku SQL.
3. Podobnie jak w poprzednich latach, największe problemy zdający mają z zadaniami z zakresu algorytmiki i programowania oraz zadaniami wymagającymi wykonywania symulacji w arkuszu kalkulacyjnym.
4. 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. 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 zapis algorytmów w pseudojęzyku lub języku programowania – w tym należy zwrócić uwagę na zapis algorytmów bez używania wbudowanych funkcji bibliotecznych.
5. Istotną rolą osób przygotowujących młodzież do egzaminu maturalnego z informatyki jest zwrócenie uwagi 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.
6. Warto 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.