

<i>Rodzaj dokumentu:</i>	Sprawozdanie za rok 2024
<i>Egzamin:</i>	Egzamin maturalny
<i>Przedmiot:</i>	Informatyka
<i>Poziom:</i>	Poziom rozszerzony
<i>Termin egzaminu:</i>	22 maja 2024 r.
<i>Data publikacji dokumentu:</i>	20 września 2024 r.

Opracowanie

Iwona Arcimowicz (Centralna Komisja Egzaminacyjna)

Ewa Kałucka (Okręgowa Komisja Egzaminacyjna w Jaworznie)

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

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	37

Opis arkusza egzaminu maturalnego

W roku szkolnym 2023/2024 egzamin maturalny z informatyki został przeprowadzany na podstawie wymagań egzaminacyjnych określonych w rozporządzeniu Ministra Edukacji i Nauki z dnia 10 czerwca 2022 r.¹

Arkusz egzaminacyjny z informatyki na poziomie rozszerzonym zawierał ogółem 22 zadania w tym 20 zadań w 6 wiązkach tematycznych.

Zadania sprawdzały wiadomości oraz umiejętności ujęte w trzech obszarach wymagań ogólnych:

- I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji (6 zadań łącznie za 14 punktów, w tym jedno zadanie zamknięte za 1 punkt).
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi (16 zadań łącznie za 35 punktów, w tym 15 zadań praktycznych – wymagających użycia komputera i zapisania programu, wykorzystania arkusza kalkulacyjnego lub bazy danych).
- III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi.

Egzamin trwał 210 minut. Za rozwiązanie wszystkich zadań można było otrzymać 50 punktów.

Dane dotyczące populacji zdających

TABELA 1. ZDAJĄCY ROZWIĄZUJĄCY ZADANIA W ARKUSZU STANDARDOWYM*

Liczba zdających (Formuła 2023)		8 808
Zdający rozwiązujący zadania w arkuszu standardowym	z liceów ogólnokształcących	4 316
	z techników	4 489
	ze szkół na wsi	122
	ze szkół w miastach do 20 tys. mieszkańców	1 239
	ze szkół w miastach od 20 tys. do 100 tys. mieszkańców	2 767
	ze szkół w miastach powyżej 100 tys. mieszkańców	4 680
	ze szkół publicznych	8 149
	ze szkół niepublicznych	659
	kobiety	927
	mężczyźni	7 881
	bez dysleksji rozwojowej	7 297
	z dysleksją rozwojową	1 511
	obywatele Ukrainy ²	12

* Dane w tabeli dotyczą tegorocznych absolwentów.

Z egzaminu w Formule 2023 i Formule 2015 zwolniono 58 osób – laureatów i finalistów Olimpiady Informatycznej.

¹ Rozporządzenie Ministra Edukacji i Nauki z dnia 10 czerwca 2022 r. w sprawie wymagań egzaminacyjnych dla egzaminu maturalnego przeprowadzanego w roku szkolnym 2022/2023 i 2023/2024 (poz. 1246).

² Dz.U. z 2024 r. poz. 167, z późn. zm.

TABELA 2. ZDAJĄCY ROZWIĄZUJĄCY ZADANIA W ARKUSZACH DOSTOSOWANYCH

Zdający rozwiązujący zadania w arkuszach dostosowanych	z autyzmem, w tym z zespołem Aspergera	203
	słabowidzący	14
	niewidomi	0
	słabosłyszący	18
	nieśłyszący	2
	z niepełnosprawnością ruchową spowodowaną mózgowym porażeniem dziecięcym	4
	z zaburzeniem widzenia barw	3
Ogółem	244	

Przebieg egzaminu

TABELA 3. INFORMACJE DOTYCZĄCE PRZEBIEGU EGZAMINU

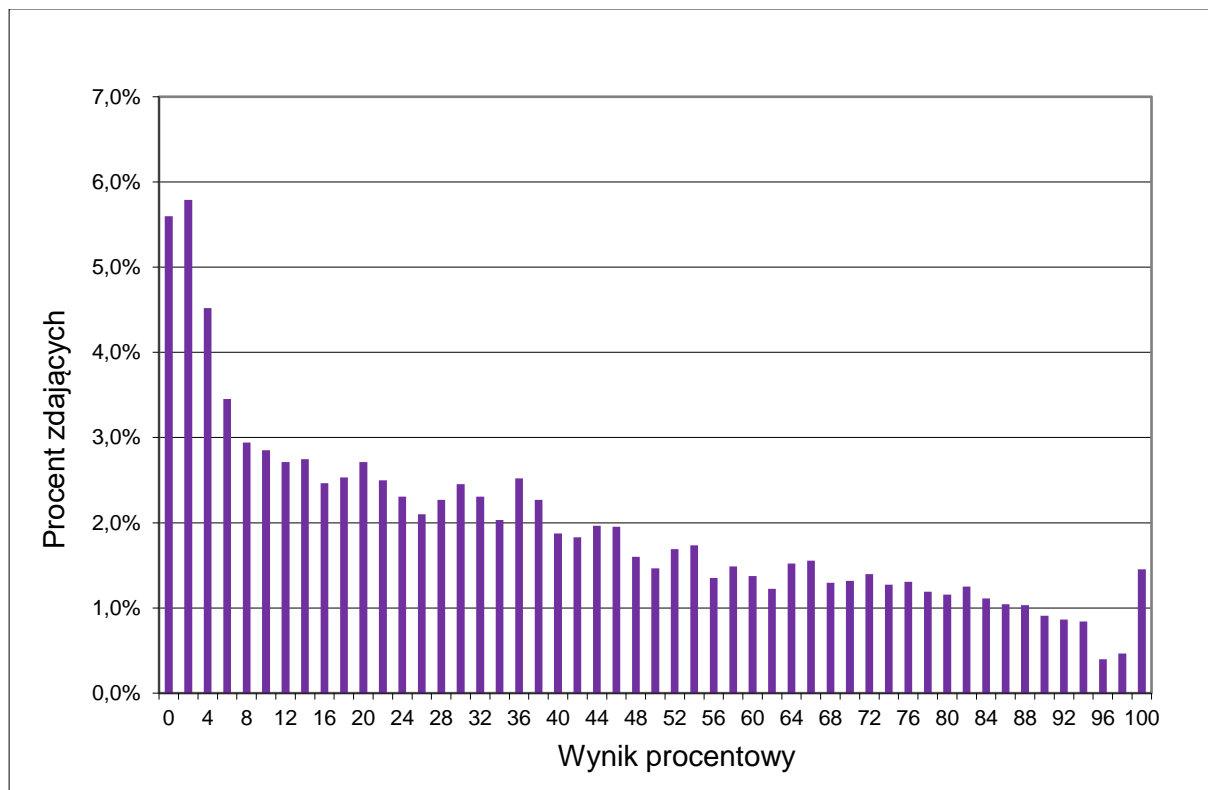
Termin egzaminu		22 maja 2024	
Czas trwania egzaminu dla arkusza standardowego		210 minut	
Liczba szkół		1 486	
Liczba zespołów egzaminatorów		10	
Liczba egzaminatorów		123	
Liczba obserwatorów ³ (§ 8 ust. 1)		17	
Liczba unieważnień ⁴	w przypadku:		
	art. 44zzv pkt 1	stwierdzenia niesamodzielnego rozwiązywania zadań przez zdającego	0
	art. 44zzv pkt 2	wniesienia lub korzystania przez zdającego w sali egzaminacyjnej z urządzenia telekomunikacyjnego	1
	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	4
	art. 44zzy ust. 10	niemożność ustalenia wyniku (np. zaginięcie karty odpowiedzi)	0
Liczba wglądów ⁴ (art. 44zzz)		226	

³ Rozporządzenie Ministra Edukacji Narodowej z dnia 1 sierpnia 2022 r. w sprawie egzaminu maturalnego (Dz.U. poz. z 2024 poz. 302).

⁴ Ustawa z dnia 7 września 1991 r. o systemie oświaty (Dz.U. z 2024 r. poz. 750).

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	8808	0	100	32	2	36	28
w tym:							
z liceów ogólnokształcących	4316	0	100	44	0	45	29
z techników	4489	0	100	20	2	28	25
z branżowych szkół II stopnia	3	-	-	-	-	-	-

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

Poziom wykonania zadań

TABELA 5. POZIOM WYKONANIA ZADAŃ

Wymagania egzaminacyjne			
Nr zad.	Wymagania ogólne	Wymagania szczegółowe <i>Gdy wymaganie dotyczy treści zakresu podstawowego szkoły ponadpodstawowej – dopisano (P).</i>	Poziom wykonania zadania (%)
1.1.	I. Rozumienie, analizowanie i rozwiązywanie problemów.	Zdający: I.2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych. P.I.4) [...] analizuje algorytmy na podstawie ich gotowych implementacji; P.I.5) sprawdza poprawność działania algorytmów dla przykładowych danych.	41
1.2.	I. Rozumienie, analizowanie i rozwiązywanie problemów. II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: I.2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych. P.I.4) [...] analizuje algorytmy na podstawie ich gotowych implementacji; P.I.5) sprawdza poprawność działania algorytmów dla przykładowych danych. P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin [...].	33
1.3.	I. Rozumienie, analizowanie i rozwiązywanie problemów. II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: I.2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych. P.I.4) [...] analizuje algorytmy na podstawie ich gotowych implementacji; P.I.5) sprawdza poprawność działania algorytmów dla przykładowych danych. P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin [...].	22
2.1.	I. Rozumienie, analizowanie i rozwiązywanie problemów. II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: P.I.1) planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania); P.I.2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach: [...] zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi [...], P.I.4) [...] analizuje algorytmy na podstawie ich gotowych implementacji. P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje	52

		z parametrami i bez parametrów [...].	
2.2.	I. Rozumienie, analizowanie i rozwiązywanie problemów. II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: P.I.1) planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania); P.I.2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach: [...] zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi [...], P.I.4) [...] analizuje algorytmy na podstawie ich gotowych implementacji. P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów [...].	45
3.1.	I. Rozumienie, analizowanie i rozwiązywanie problemów. II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: I.2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych; I.3) objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność; I.5) przedstawia sposoby reprezentowania w komputerze znaków, liczb [...]. I+II.1) zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach [...]. P.I.2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach: [...] zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi [...]. P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów [...].	33
3.2.	II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2). II.1) projektuje i tworzy rozbudowane programy w procesie rozwiązywania problemów, wykorzystuje w programach dobrane do algorytmów struktury danych, w tym struktury dynamiczne i korzysta z dostępnych bibliotek dla tych struktur;	43

		<p>II.2) stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów;</p> <p>II.3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów.P.I.2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy:</p> <p>a) na liczbach: [...] zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi [...].</p> <p>P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów [...].</p>	
3.3.	<p>II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.</p>	<p>Zdający:</p> <p>P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2).</p> <p>II.1) projektuje i tworzy rozbudowane programy w procesie rozwiązywania problemów, wykorzystuje w programach dobrane do algorytmów struktury danych, w tym struktury dynamiczne i korzysta z dostępnych bibliotek dla tych struktur;</p> <p>II.2) stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów;</p> <p>II.3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów.P.I.2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy:</p> <p>a) na liczbach: [...] zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi [...].</p> <p>P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów [...].</p>	32
4.1.	<p>I. Rozumienie, analizowanie i rozwiązywanie problemów.</p> <p>II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.</p>	<p>Zdający:</p> <p>P.I.5) sprawdza poprawność działania algorytmów dla przykładowych danych;</p> <p>P.I.2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach [...].</p> <p>P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów [...].</p>	33

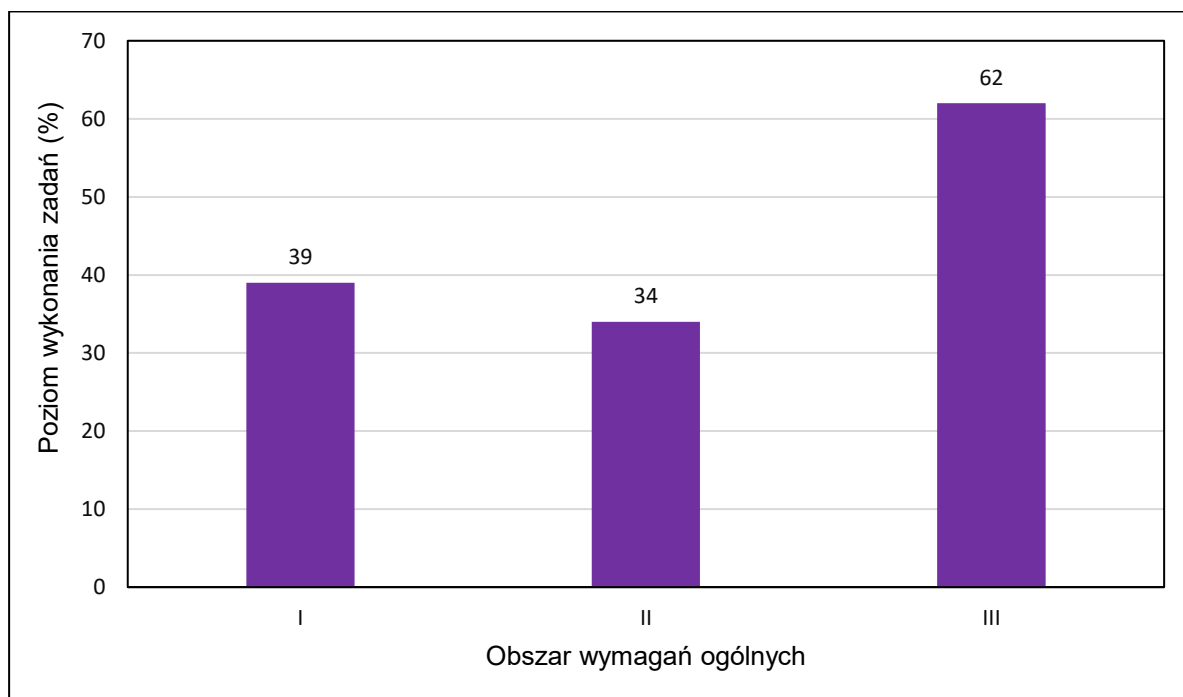
4.2.	I. Rozumienie, analizowanie i rozwiązywanie problemów. II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: I.2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych. II.3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów. P.I.2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: c) porządkowania ciągu liczb [...]. P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów [...].	27
4.3.	I. Rozumienie, analizowanie i rozwiązywanie problemów. II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: I.2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych. II.3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów. P.I.2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach [...]. P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów [...]. I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: a) rozkładania liczby na czynniki pierwsze, c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie.	16
4.4.	I. Rozumienie, analizowanie i rozwiązywanie problemów. II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: I.2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych. II.3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów. P.I.2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach [...]. P.II.1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów [...].	9

		I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: a) rozkładania liczby na czynniki pierwsze, c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie.	
5.	III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi.	Zdający: [...] opisuje model warstwowy sieci komputerowej oraz model sieci internet, opisuje podstawowe funkcje urządzeń i protokoły stosowane w przepływie informacji i w zarządzaniu siecią.	62
6.	I. Rozumienie, analizowanie i rozwiązywanie problemów.	Zdający: P.I.2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach: [...] zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi [...].	45
7.1.	II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: II.4) przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym: b) stosuje zaawansowane funkcje arkusza kalkulacyjnego w zależności od rodzaju danych [...]. P.II.3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami: b) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych.	59
7.2.	II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: II.4) przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym: b) stosuje zaawansowane funkcje arkusza kalkulacyjnego w zależności od rodzaju danych [...]. P.II.3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami: b) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych.	52
7.3.	II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: II.4) przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:	43

		<p>b) stosuje zaawansowane funkcje arkusza kalkulacyjnego w zależności od rodzaju danych [...].</p> <p>P.II.3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:</p> <p>b) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych.</p>	
7.4.	<p>II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.</p>	<p>Zdający:</p> <p>II.4) przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:</p> <p>b) stosuje zaawansowane funkcje arkusza kalkulacyjnego w zależności od rodzaju danych [...].</p> <p>P.II.3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:</p> <p>b) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych.</p>	11
8.1.	<p>II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.</p>	<p>Zdający:</p> <p>II.4) przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:</p> <p>c) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie.</p> <p>P.II.3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:</p> <p>c) wyszukuje informacje, korzystając z bazy danych opartej na co najmniej dwóch tabelach, definiuje relacje, stosuje filtrowanie, formułuje kwerendy.</p>	61
8.2.	<p>II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.</p>	<p>Zdający:</p> <p>II.4) przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:</p> <p>c) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie.</p>	39

		P.II.3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami: c) wyszukuje informacje, korzystając z bazy danych opartej na co najmniej dwóch tabelach, definiuje relacje, stosuje filtrowanie, formułuje kwerendy.	
8.3.	II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: II.4) przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym: c) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz siecią aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie. P.II.3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami: c) wyszukuje informacje, korzystając z bazy danych opartej na co najmniej dwóch tabelach, definiuje relacje, stosuje filtrowanie, formułuje kwerendy.	44
8.4.	II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.	Zdający: II.4) przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym: c) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz siecią aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie. P.II.3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami: c) wyszukuje informacje, korzystając z bazy danych opartej na co najmniej dwóch tabelach, definiuje relacje, stosuje filtrowanie, formułuje kwerendy.	36

WYKRES 2. POZIOM WYKONANIA ZADAŃ W OBSZARZE WYMAGAŃ OGÓLNYCH



Komentarz

W roku 2024 po raz drugi odbył się egzamin maturalny z informatyki w formule 2023. W tym roku do egzaminu z informatyki przystąpiło 8808 osób, w tym 4316 absolwentów liceów oraz, po raz pierwszy w tej formule, 4489 absolwentów techników.

Analiza jakościowa zadań

Tegoroczny arkusz składał się z 22 zadań. 20 zadań ujęto w 6 wiązek tematycznych, pozostałe dwa zadania były odrębnymi zadaniami. Trzy zadania okazały się dla zdających bardzo trudne (poziom wykonania⁵ niższy niż 19%), 14 zadań było dla zdających trudnych (poziom wykonania każdego z tych zadań mieścił się w zakresie od 20% do 49%), 5 zadań okazało się umiarkowanie trudnymi (poziom wykonania w zakresie od 50% do 69%). Poziom wykonania żadnego zadania nie przekroczył 70%, co oznacza, że w arkuszu nie było zadań, które dla zdających były łatwe lub bardzo łatwe.

Rozkład punktacji na poszczególnych poziomach trudności wyglądał następująco: za zadania bardzo trudne można było uzyskać w sumie 9 punktów (18% maksymalnej liczby punktów do zdobycia), za zadania trudne można było uzyskać w sumie 32 punkty (64% maksymalnej liczby punktów do zdobycia), a za zadania umiarkowanie trudne można było uzyskać w sumie 9 punktów (18% maksymalnej liczby punktów do zdobycia). Zdecydowaną większość zadań stanowiły zadania praktyczne – wymagające użycia komputera. Za ich rozwiązanie można było zdobyć w sumie 34 punkty, czyli 68% wszystkich możliwych punktów do zdobycia. Za poprawne rozwiązanie każdego z tych zadań zdający otrzymywali punkty tylko wówczas, gdy zapisana odpowiedź w pliku tekstowym wynikała z komputerowej realizacji zadania. Brak plików zawierających komputerową realizację zadań był jednoznaczny z brakiem rozwiązania tych zadań.

Zadania, z którymi zdający poradzili sobie najslabiej

W tej części komentarza omówimy zadania, z którymi zdający poradzili sobie najslabiej. Przyjmiemy do analizy zadania, których poziom wykonania jest niższy niż 35%. Takich zadań w arkuszu jest 9.

Zadania, których poziom wykonania jest niższy niż 35%, wymieniając kolejno od najtrudniejszego, to (w nawiasach oprócz poziomu wykonania zapisano jakiej tematyki dotyczy zadanie oraz rodzaj zadania):

1. Zadanie 4.4. (9%, zadanie programistyczne, praktyczne),
2. Zadanie 7.4. (11%, zadanie dotyczące symulacji w arkuszu kalkulacyjnym, praktyczne),
3. Zadanie 4.3. (16%, zadanie programistyczne, praktyczne),
4. Zadanie 1.3. (22%, zadanie z zakresu algorytmiki, otwarte),
5. Zadanie 4.2. (27%, zadanie programistyczne, praktyczne),
6. Zadanie 3.3. (32%, zadanie programistyczne, praktyczne),
7. Zadanie 4.1. (33%, zadanie programistyczne, praktyczne),

⁵ Poziom wykonania zadania to parametr, który określa się jako iloraz (wyrażony w procentach) średniego wyniku za dane zadanie i maksymalnej liczby punktów możliwych do uzyskania za to zadanie.

8. Zadanie 1.2. (33%, zadanie z zakresu algorytmiki, otwarte),
9. Zadanie 3.1. (33%, zadanie z zakresu algorytmiki, otwarte).

Jak wynika z podanego zestawienia wśród najtrudniejszych zadań przeważają (5 z 9) zadania wymagające napisania programu komputerowego w wybranym (i dopuszczonym do egzaminu w komunikacie dyrektora CKE) języku programowania. W tej grupie zawarta jest cała wiązka zadań 4.1.–4.4. Trzy z pozostałych zadań to zadania otwarte wymagające analizy lub zapisania algorytmu w arkuszu egzaminacyjnym, a jedno (w tym jedno z najtrudniejszych) to zadanie wymagające wykonania symulacji w arkuszu kalkulacyjnym, które zostanie omówione dokładniej w dalszej części.

Do rozwiązania wiązki zadań programistycznych 4.1.–4.4. należało wykorzystać jeden plik z danymi, w którym w dwóch wierszach zapisano liczby rozdzielone znakami spacji. W pierwszym wierszu zapisano 3000 liczb pierwszych z przedziału od 2 do 2000, w drugim – 20 liczb całkowitych z przedziału od 2 do 1 000 000 000. Pierwszą trudnością dla zdających było poprawne odczytanie danych zapisanych w ten sposób.

W zadaniu 4.1. należało policzyć, ile liczb z pierwszego wiersza jest dzielnikiem jakiegokolwiek liczby spośród liczb zapisanych w drugim wierszu. Było to najłatwiejsze zadanie w tej wiązce (poziom wykonania – 33%). W zadaniu 4.2. (poziom wykonania – 27%) należało podać liczbę, która znajdzie się na sto pierwszym miejscu po uporządkowaniu liczb z pierwszego wiersza w pliku. To zadanie wymagało umiejętności wykorzystania dowolnego algorytmu sortowania lub znajomości funkcji wbudowanych sortujących dane. Dużo trudniejsze dla zdających były zadania 4.3 i 4.4. (poziom wykonania odpowiednio 16% i 9%).

W zadaniu 4.3. (poziom wykonania 16%) należało dla każdej liczby z drugiego wiersza pliku rozstrzygnąć, czy da się ją przedstawić jako iloczyn jedynie liczb z pierwszego wiersza, przy czym liczba wystąpień danej liczby w iloczynie nie mogła przekraczać liczby jej wystąpień w pierwszym wierszu pliku. Wśród rozwiązań zdających można zaobserwować dwa podstawowe sposoby wykonania zadania. Pierwszy sposób polegał na rozłożeniu kolejnych liczb z drugiego wiersza na czynniki pierwsze (pamiętajmy, że wszystkie liczby z górnego wiersza pliku to liczby pierwsze), a następnie, dla każdego czynnika danej liczby porównywano liczbę jego wystąpień w rozkładzie na czynniki pierwsze z liczbą wystąpień tego czynnika w pierwszym wierszu pliku. Takie podejście narzuca się jako pierwsze, jednak powoduje, że zdający musi umieć nie tylko rozłożyć liczbę na czynniki pierwsze, ale także zdecydować jak je zapamiętać (np. z pomocą listy) i zliczyć. Drugi najczęstszy sposób rozwiązania polegał na tym, że dla każdej liczby z drugiego wiersza pobierane były kolejne liczby pierwsze z górnego wiersza i – jeśli liczba pierwsza była dzielnikiem danej liczby – obliczano iloraz, dla którego pobierano kolejną liczbę z górnego wiersza. Jeśli w wyniku kolejnego dzielenia otrzymano iloraz równy 1, oznaczało to, że sprawdzana liczba spełnia warunek postawiony w zadaniu.

Przykładowe poprawne rozwiązanie zadania 4.3. w języku Python z wykorzystaniem pierwszego sposobu przedstawiono na przykładzie 1.


```

def z3():
    pierwsza = []
    druga = []
    x = 0
    with open("liczby.txt") as plik:
        for linia in plik:
            if (x == 0):
                for i in range(3000):
                    pierwsza.append(int(linia.strip().split()[i]))
            else:
                for i in range(20):
                    druga.append(int(linia.strip().split()[i]))
            x += 1
    for i in range(len(druga)):
        czynniki=[]
        a=2
        odp=druga[i]
        while(druga[i]>1):
            while(druga[i]%a==0):
                czynniki.append(a)
                druga[i]/=a
            a+=1
        w=True
        for j in czynniki:
            if j not in pierwsza or czynniki.count(j)>pierwsza.count(j):
                w=False
                break

        if(w):
            print(odp)

```

Przykład 1.

Na przykładzie 2. przedstawiono rozwiązanie zadania sposobem pierwszym w języku c++.

```
int cos;

for(int f =0; f<20; f++)
{
    cos = tab2[f];
    for(int l =0; l<2000; l++)
    {
        mapka[l]=0;
    }

    for(int i =0; i<3000; i++)
    {
        mapka[tabp[i]]++;
    }

    int i =2;
    while (cos>1&& i<2000)
    {

        if(cos%i==0&&mapka[i]>0)
        {
            mapka[i]--;
            cos = cos/i;
        }
        else {
            i++;
        }

    }
    if(cos==1)
    {
        cout << tab2[f] << endl;
    }
}
```

Przykład 2.

Fragmety poprawnych rozwiązań zadania 4.3. w języku c++ z wykorzystaniem sposobu drugiego przedstawiono na przykładzie 3.

```

for(int i=0; i<20; i++)
{
    for(int j=0; j<3000; j++)
    {
        if(wiersz2[i]%wiersz1[j]==0)
        {
            wiersz2[i]/=wiersz1[j];
            //cout << wiersz2[i] << endl;
        }
    }
}
for(int i=0; i<20; i++)
{
    if(wiersz2[i]==1)
    {
        cout << wiersz2bezzmian[i] << endl;
    }
}

```

Przykład 3.

Przykład 4. przedstawia fragment rozwiązania, w którym zdający napisał oddzielną funkcję, która dla podanej liczby sprawdza, czy liczba ta jest iloczynem liczb z pierwszego wiersza pliku (wykonywane jest dzielenie danej liczby przez kolejne elementy z pierwszego wiersza pliku, pod warunkiem że są one dzielnikami tej liczby).

```

bool iloczyn(int x)
{
    for(int i=0; i<N; i++)
    {
        if(x%a[i]==0)    x/=a[i];
        if(x==1)        return 1;
    }
    return 0;
}

```

```

for(int j=0; j<M; j++)
{
    if(iloczyn(b[j]))    cout<<b[j]<<endl;
}

```

Przykład 4.

W zadaniu 4.4. (poziom wykonania – 9%) należało znaleźć w ciągu liczb z pierwszego wiersza spójny fragment, który zawiera co najmniej 50 elementów i którego średnia arytmetyczna jest największa. Jednym z najprostszych sposobów rozwiązania polegał na tym, że dla każdej liczby z pierwszego wiersza obliczano sumę najpierw 50 kolejnych liczb (zaczynając od aktualnej) i obliczano ich średnią, a następnie do obliczonej w poprzednim

kroku sumy dodawano kolejne liczby i za każdym razem obliczano i porównywano średnie arytmetyczne. Zasadniczą trudnością dla zdających było zrozumienie treści zadania. Wielu zdających pominięło to zadanie.

Przykład 5. przedstawia poprawne rozwiązanie zadania 4.4. w języku Python.

```
pierwsza = []
druga = []
x = 0
with open("liczby.txt") as plik:
    for linia in plik:
        if (x == 0):
            for i in range(3000):
                pierwsza.append(int(linia.strip().split()[i]))
        else:
            for i in range(20):
                druga.append(int(linia.strip().split()[i]))
    x += 1
maxsrednia = 0
for i in range(len(pierwsza)):
    ciag=[]
    for j in range(i,len(pierwsza)):
        ciag.append(pierwsza[j])
        if(len(ciag)>=50):
            srednia=0
            for a in ciag:
                srednia+=a
            srednia=srednia/len(ciag)
            if(srednia>maxsrednia):
                maxsrednia=srednia
            wynik=ciag
print(maxsrednia)
print(len(wynik))
print(wynik[0])
```

Przykład 5.

Poprawne rozwiązanie zadania 4.4. napisane w języku c++ przedstawiono na przykładzie 6.

```

fstream plik;
plik.open("liczby.txt");
float tab[3000];
int passa,pocz,maxpassa;
float srednia,maxsr=0,s;
for(int i=0; i<3000; i++)
{
    plik>>tab[i];
}
for(int i=0; i<3000; i++)
{
    passa=0;
    s=0;
    for(int j=i; j<3000; j++)
    {
        passa++;
        s=s+tab[j];
        if(passa>=50){
            if((s/(j-i+1))>maxsr){
                pocz=tab[i];
                maxsr=s/(j-i+1);
                maxpassa=passa;
            }
        }
    }
}
cout<<maxsr<<" "<<maxpassa<<" "<<pocz<<endl;

```

Przykład 6.

Wśród najtrudniejszych zadań dla zdających znalazły się także zadania 1.2. i 1.3. (poziom wykonania odpowiednio 33% i 22%) z pierwszej wiązki zadań w arkuszu, która dotyczyła analizy algorytmu przedstawionego na rysunku 1.

Zadanie 1. Plansza

Dana jest prostokątna plansza złożona z n wierszy i m kolumn zawierająca $n * m$ pól. Wiersze są ponumerowane od góry kolejnymi liczbami $1, 2, \dots, n$, natomiast kolumny od lewej do prawej kolejnymi liczbami $1, 2, \dots, m$. Każde pole jest albo białe, albo czarne.

Planszę możemy opisać jako tablicę dwuwymiarową $A[1..n][1..m]$, w której $A[i][j] = 0$, jeśli pole w i -tym wierszu i j -tej kolumnie jest czarne, natomiast $A[i][j] = 1$, jeśli to pole jest białe. Pola w lewym górnym rogu oraz prawym dolnym rogu zawsze są białe (czyli $A[1][1] = 1$ oraz $A[n][m] = 1$).

Rozważmy następujący algorytm, w którym jest wykorzystywana pomocnicza tablica $P[1..n][1..m]$, przyjmująca wartości logiczne (PRAWDA albo FAŁSZ).

Specyfikacja:

Dane:

n, m – liczby całkowite dodatnie, wymiary planszy
 $A[1..n][1..m]$ – opis planszy

Wynik:

PRAWDA albo FAŁSZ

$P[1][1] \leftarrow$ PRAWDA

dla $i = 1, 2, \dots, n$ **wykonuj**

dla $j = 1, 2, \dots, m$ **wykonuj**

jeżeli $A[i][j] = 0$

$P[i][j] \leftarrow$ FAŁSZ

w przeciwnym przypadku

jeżeli $i = 1$ **oraz** $j \neq 1$

$P[i][j] \leftarrow P[i][j-1]$

jeżeli $i \neq 1$ **oraz** $j = 1$

$P[i][j] \leftarrow P[i-1][j]$

jeżeli $i \neq 1$ **oraz** $j \neq 1$

$P[i][j] \leftarrow P[i][j-1]$ **lub** $P[i-1][j]$

podaj wynik $P[n][m]$

Rysunek 1.

W zadaniu 1.2. (poziom wykonania – 33%) należało podać przykład planszy o 5 wierszach i 5 kolumnach z najwyżej dwoma czarnymi polami, dla której wynikiem działania algorytmu będzie FAŁSZ, oraz przykład planszy 4x4 z co najmniej 9 czarnymi polami, dla której wynikiem będzie PRAWDA. Aby poprawnie rozwiązać to zadanie należało dokładnie przeanalizować działanie algorytmu i wywnioskować, że algorytm daje w wyniku prawdę jedynie w przypadku, gdy możliwe jest przejście od lewego górnego pola do prawego dolnego wyłącznie po polach białych, poruszając się tylko w prawo albo w dół.

Przykłady 7. i 8. przedstawiają poprawne uzupełnienia tabel przez zdających.

- a) o 5 wierszach i 5 kolumnach, na której **co najwyżej 2 pola** są czarne, a wynikiem działania algorytmu jest FAŁSZ

- b) o 4 wierszach i 4 kolumnach, na której **co najmniej 9 pól** jest czarnych, a wynikiem działania algorytmu jest PRAWDA.

Przykład 7.

- a) o 5 wierszach i 5 kolumnach, na której **co najwyżej 2 pola** są czarne, a wynikiem działania algorytmu jest FAŁSZ

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	0
1	1	1	0	1

0 - CZARNE
1 - BIAŁE

- b) o 4 wierszach i 4 kolumnach, na której **co najmniej 9 pól** jest czarnych, a wynikiem działania algorytmu jest PRAWDA.

1	1	1	1
0	0	0	1
0	0	0	1
0	0	0	1

Przykład 8.

W zadaniu 1.3. (poziom wykonania – 22%) należało podać maksymalną liczbę czarnych pól, jaką może mieć plansza $n \times n$, aby wynikiem działania algorytmu była prawda.

Przykładowe poprawne rozwiązania zadania 1.3. przedstawiono na poniższych przykładach:

Zadanie 1.3. (0–1)

Dana jest kwadratowa plansza o n wierszach i n kolumnach.

Podaj, jaka jest największa możliwa liczba czarnych pól na tej planszy, dla których wynikiem działania algorytmu jest PRAWDA.

Odpowiedź: $N^2 - 2N + 1 = (N-1)(N-1) = (N-1)^2$

Przykład 9.

Zadanie 1.3. (0–1)

Dana jest kwadratowa plansza o n wierszach i n kolumnach.

Podaj, jaka jest największa możliwa liczba czarnych pól na tej planszy, dla których wynikiem działania algorytmu jest PRAWDA.

Odpowiedź: $(n-1) \cdot (n-1)$

Przykład 10.

Zadanie 1.3. (0–1)

Dana jest kwadratowa plansza o n wierszach i n kolumnach.

Podaj, jaka jest największa możliwa liczba czarnych pól na tej planszy, dla których wynikiem działania algorytmu jest PRAWDA.

Odpowiedź: $(n-1)^2$

Przykład 11.

Ostatnim omawianymi zadaniami z grupy najtrudniejszych dla zdających są zadania 3.1. i 3.3. (poziom wykonania odpowiednio 33% i 32%). Zadanie 3.1. wymagało zapisania w arkuszu egzaminacyjnym algorytmu w postaci pseudokodu lub programu w wybranym przez zdającego (w deklaracji) języku programowania, natomiast zadanie 3.3. było zadaniem praktycznym wymagającym napisania programu. Obydwa zadania dotyczyły wprowadzonego na początku tej wiązki zadań pojęcia *nieparzystego skrótu* liczby (rysunek 2.):

Zadanie 3. Nieparzysty skrót

Nieparzystym skrótem dodatniej liczby całkowitej n nazwiemy dodatnią liczbę całkowitą m , która powstaje przez usunięcie cyfr parzystych z zapisu dziesiętnej liczby n .

Nieparzysty skrót liczby całkowitej n nie istnieje, gdy jej zapis dziesiętny składa się tylko z cyfr parzystych.

Przykład:

Nieparzystym skrótem liczby 294762 jest liczba 97.

Nieparzystym skrótem liczby 39101 jest liczba 3911.

Nieparzysty skrót liczby 224 nie istnieje.

Rysunek 2.

Zadanie 3.1. polegało na zapisaniu w postaci pseudokodu lub języka programowania algorytmu wyznaczającego nieparzysty skrót danej liczby n . W zadaniu dodano założenie, że podana liczba posiada taki skrót. Dodatkowo w zadaniu znalazła się uwaga informująca, jakich instrukcji wolno, a jakich nie wolno używać w zapisie algorytmu (rysunek 3.):

Uwaga: Twój algorytm może używać **wyłącznie zmiennych przechowujących liczby całkowite** oraz może operować **wyłącznie na liczbach całkowitych**. W zapisie możesz wykorzystać tylko operacje arytmetyczne: dodawanie, odejmowanie, mnożenie, dzielenie, dzielenie całkowite, resztę z dzielenia oraz porównywanie liczb, instrukcje sterujące, przypisania do zmiennych lub samodzielnie napisane funkcje, wykorzystujące wyżej wymienione operacje. **Zabronione** jest używanie funkcji wbudowanych oraz operatorów innych niż wymienione.

Rysunek 3.

Warto dokładniej przyjrzeć się tej uwadze, ponieważ tego typu instrukcje pojawiają się przy wszystkich zadaniach wymagających zapisu algorytmu w arkuszach z informatyki. Niestosowanie się do takich uwag może spowodować obniżenie, a nawet wyzerowanie oceny za zadanie. Rozważmy kolejne zapisy:

- „Twój algorytm może używać wyłącznie zmiennych przechowujących liczby całkowite oraz może operować wyłącznie na liczbach całkowitych” – ta uwaga wskazuje, że w zapisie tego algorytmu nie można używać np. zmiennej tekstowej, znakowej, ale także tablicy czy listy
- „w zapisie możesz wykorzystać tylko operacje arytmetyczne: dodawanie, odejmowanie, mnożenie, dzielenie, dzielenie całkowite, resztę z dzielenia oraz porównywanie liczb, instrukcje sterujące, przypisania do zmiennych lub samodzielnie napisane funkcje, wykorzystujące wyżej wymienione operacje” – w tym miejscu zostały wymienione wszystkie dozwolone operacje, co oznacza, że skoro nie zostały wymienione operacje takie jak potęgowanie i pierwiastkowanie, to nie są one dozwolone; jest dozwolone porównywanie liczb – ale nie zmiennych innego typu itd.
- „zabronione jest używanie funkcji wbudowanych oraz operatorów innych niż wymienione” – ten zapis oprócz tego, że zabrania użycia praktycznie wszystkich funkcji wbudowanych (np. służących do zamiany typów lub choćby pierwiastkowania) wzmacnia poprzedni zapis wyliczający operacje dozwolone, to jest: skoro nie wymieniono np. jakiejś operacji arytmetycznej – to nie jest ona dozwolona.

Podsumowując, w zadaniu 3.1., aby pobrać kolejne cyfry liczby n , należało wykorzystać operacje arytmetyczne na tej liczbie – resztę z dzielenia przez 10 (w wyniku której otrzymuje się ostatnią cyfrę liczby) oraz dzielenie całkowite przez 10 („obcięcie” ostatniej cyfry). Niedozwolone było traktowanie danej liczby jako tekstu i pobieranie jej cyfr jako znaków. Podobnie podczas tworzenia wynikowej liczby m konieczne było używanie podstawowych operacji arytmetycznych, a zabronione było np. dopisywanie cyfr do tekstu. Część zdających, zapisując rozwiązanie tego zadania, ignorowała zapisy umieszczone w uwadze i używała niedozwolonych w tym zadaniu typów zmiennych (typ tekstowy, tablica, lista).

Podajemy poniżej kilka przykładów poprawnego rozwiązania tego zadania.

Przykłady 12. i 13. przedstawiają zapisy algorytmu w pseudokodzie.

```
p ← 1, liczba, liczba ← 0
liczba
dopóki n ≠ 0 wykonuj:
    jeżeli n mod 2 = 1
        liczba ← liczba + (n mod 10) * p
        p ← p * 10
    n ← n div 10
zwróć
m ← liczba
zwróć m i zakończ
```

Przykład 12.

```
b ← 1
m ← 0
dopóki n > 0 wykonuj
    a ← n mod 10
    jeżeli (a mod 2) = 1
        m ← m + a * b
        b ← b * 10
    n ← n div 10
zwróć m
```

Przykład 13.

Zapis algorytmu w języku Python przedstawiono na przykładzie 14.

```

m = 0
x = 7
y = 0

while n != 0:
    y = n % 10
    if y % 2 == 1:
        m = m + y * x
        x = x * 10
    n = n // 10

print(m)

```

Przykład 14.

Na przykładzie 15. znajduje się zapis algorytmu w c++.

```

int oddshortcut(int n) {
    int m = 0;
    int i = 1;
    while(n > 0) {
        int last = n % 10;
        if(last % 2 != 0) {
            m += i * last;
            i *= 10;
        }
        n /= 10;
    }
    return m;
}

```

Przykład 15.

Zadanie 3.3. było zadaniem programistycznym, w którym należało wśród podanych w pliku tekstowym liczb znaleźć takie, dla których największym wspólnym dzielnikiem danej liczby i jej nieparzystego skrótu jest liczba 7. W tym zadaniu należało pokonać dwie trudności: pierwsza, to poprawne wyznaczenie nieparzystego skrótu dla każdej liczby z pliku, druga – poprawne obliczenie największego wspólnego dzielnika liczby i jej skrótu. Przy okazji warto zauważyć, że w zadaniach programistycznych nie ma założeń wykluczających stosowanie

np. funkcji wbudowanych, ani ograniczających sposoby zapisu algorytmów. Część zdających miała problemy z wyznaczeniem skrótu, ale część z nich nie poradziła sobie także z obliczeniem największego wspólnego dzielnika mimo, że algorytmy obliczające jego wartość powinny być znane już uczniom szkół podstawowych.

Na przykładzie 16. przedstawiono fragment przykładowego poprawnego rozwiązania w języku Python. Warto zauważyć, że gdyby funkcja wyznaczająca nieparzysty skrót, prawidłowo zastosowana w tym przykładzie, została przedstawiona jako zapis algorytmu w zadaniu 3.1. – nie spełniałaby warunków zadania, ponieważ w jej zapisie korzysta się z niedozwolonych w zadaniu 3.1. funkcji wbudowanych i typów.

```
def NWD(a, b):
    if b == 0:
        return a
    else:
        return NWD(b, a % b)
def NieparzystySkrot(liczba):
    m = ""
    for i in liczba:
        if int(i) % 2 != 0:
            m += i

    return int(m)
def Zadanie3_3(dane, wyjscie):
    print("Zadanie 3.3", file=wyjscie)
    for liczba in dane:
        skrot = NieparzystySkrot(liczba)
        dzielnik = NWD(int(liczba), int(skrot))

        if dzielnik == 7:
            print(liczba, file=wyjscie)
```

Przykład 16.

Rozwiązanie zadania w języku c++ przedstawiono na przykładzie 17.

```

int skrot(int n)
{
    int m = 0;
    int cnt = 1;
    while (n>0)
    {
        if((n%10)%2==1)
        {
            m+=(n%10)*cnt;
            cnt*=10;
        }
        n/=10;
    }
    return m;
}

```

```

file.open("skrot.txt");
file2.open("skrot2.txt");
for(int i=0; i<200; i++)
{
    file >> num;
    file2 >> num2;
    if(hasAllEvenDig(num))
    {
        max3 = max(max3, num);
        ans2++;
    }
    if(__gcd(num2, skrot(num2)) == 7)
    {
        cout << num2 << endl;
    }
}

```

Przykład 17.

Zadania, z którymi zdający poradzili sobie najlepiej

W tej części komentarza omówimy zadania, z którymi zdający poradzili sobie najlepiej. Przyjmiemy do analizy zadania, których poziom wykonania jest wyższy niż 50%. Takich zadań w arkuszu jest 5.

Zadania, których poziom wykonania jest wyższy niż 50%, wymieniając kolejno od najłatwiejszego, to (w nawiasach oprócz poziomu wykonania zapisano jakiej tematyki dotyczy zadanie oraz rodzaj zadania):

1. Zadanie 5. (62%, zadanie teoretyczne dotyczące sieci komputerowych, zamknięte)
2. Zadanie 8.1. (61%, zadanie bazodanowe, praktyczne)
3. Zadanie 7.1. (59%, zadanie dotyczące arkusza kalkulacyjnego, praktyczne)
4. Zadanie 7.2. (52%, zadanie dotyczące arkusza kalkulacyjnego, praktyczne)
5. Zadanie 2.1. (52%, zadanie z zakresu algorytmiki, otwarte).

Najłatwiejszym dla zdających zadaniem okazało się zadanie 5 (poziom wykonania – 62%) – zadanie zamknięte dotyczące wiedzy o zastosowaniu podstawowych protokołów wykorzystywanych w sieci Internet (HTTP, FTP i DHCP).

Na drugim miejscu pod względem najlepszych wyników znalazło się zadanie 8.1. (poziom wykonania – 61%), które było pierwszym zadaniem w wiązce zadań bazodanowych. Sprawdzało ono umiejętność tworzenia prostej kwerendy z zastosowaniem sumowania i sortowania wyników.

Kolejne dwa zadania 7.1. i 7.2. (poziom wykonania odpowiednio 59% i 52%) sprawdzały umiejętność posługiwania się funkcjami arkusza kalkulacyjnego. W zadaniu 7.1. należało wykonać zestawienie liczby kilogramów jabłek odmian zimowych kupionych przez poszczególnych klientów, zastosować odpowiednie sortowanie i podać w odpowiedzi trzy największe wyniki tego zestawienia. Najwygodniejszym i najczęściej stosowanym przez zdających sposobem wykonania tego zadania było wykorzystanie tabeli przestawnej. W celu rozwiązania zadania 7.2. należało połączyć dane z dwóch plików (jabłka.txt i cennik.txt), obliczyć całkowity przychód hurtowni jabłek, o której mowa w zadaniu, oraz podać nazwę odmiany jabłek generującą najwyższy przychód. W tym przypadku rozwiązania zdających przedstawiają głównie dwa sposoby postępowania. Pierwszy sposób, to połączenie danych o sprzedaży jabłek z danymi z cennika np. za pomocą funkcji *wyszukaj.pionowo*, a następnie zastosowanie narzędzia tabeli przestawnej do obliczenia przychodu ze sprzedaży poszczególnych odmian. Drugi sposób to zastosowanie tabeli przestawnej w pierwszej kolejności (aby otrzymać zestawienie ilości sprzedanych jabłek każdej odmiany), a następnie obliczenie przychodu ze sprzedaży poszczególnych odmian przez pomnożenie wyników z tabeli przestawnej przez wartości z pliku cennik.

Przykład 18. przedstawia rozwiązanie zadania 7.2. pierwszym opisanym sposobem:

SUMA				=WYSZUKAJ.PIONOWO(B2;cennik;2)*E2		
	A	B	C	WYSZUKAJ.PIONOWO(szukana_wartość; tabela_tablica; nr_indeksu_kolumny; [przeszukiwany_zakres])		S
1	data	odmiana	kod nip	kg	przychód	ile już kupił
2	03.01.2022	Jonagold	Z 128-29-15-591	470	cennik;2	470
3	03.01.2022	Jonagold	Z 192-09-72-275	410		410
4	03.01.2022	Jonagored	Z 140-36-11-559	242		242
5	03.01.2022	Jonagored	Z 053-79-35-388	533		1865,5
6	03.01.2022	Jonagold	Z 159-34-45-151	543		1846,2
7	03.01.2022	Alwa	Z 159-34-45-151	341		988,9
8	03.01.2022	Ligol	Z 093-96-93-428	284		965,6
9	03.01.2022	Idared	Z 102-48-01-310	159		540,6
10	03.01.2022	Gala	Z 159-34-45-151	609		2070,6
11	03.01.2022	Idared	Z 047-26-54-835	464		1577,6
12	03.01.2022	Szampion	Z 050-38-86-889	570		1368
13	03.01.2022	Gloster	Z 170-89-76-803	222		754,8
14	03.01.2022	Gala	Z 192-09-72-275	720		2448
15	03.01.2022	Jonagold	Z 080-77-49-649	283		962,2
16	04.01.2022	Szampion	Z 170-26-38-135	204		489,6
17	04.01.2022	Alwa	Z 140-36-11-559	368		1067,2
18	04.01.2022	Gloster	Z 102-48-01-310	110		374
19	04.01.2022	Szampion	Z 089-90-67-935	534		1281,6
20	04.01.2022	Cortland	Z 128-91-02-348	438		1401,6
21	04.01.2022	Gloster	Z 080-77-49-649	34		115,6

odmiana	cena
Alwa	2,9
Antonowka	3,2
Cortland	3,2
Delikates	3,2
Gala	3,4
Gloster	3,4
Idared	3,4
Jonagold	3,4
Jonagored	3,5
Kosztela	2,5
Ligol	3,4
Lobo	2,5
Melba	2,7
Papierowka	3,2
Reneta	3,5
Szampion	2,4

	A	B	C	D	E	F
1						
2						
3	Etykiety wierszy	Suma z przychod				
4	Alwa	104124,5			Reneta	389921
5	Antonowka	79593,6			Gala	302389,2
6	Cortland	112400			Jonagored	256949
7	Delikates	99340,8			Jonagold	197978,6
8	Gala	302389,2			Papierowka	164332,8
9	Gloster	107504,6			Melba	150130,8
10	Idared	100837,2			Ligol	120312,4
11	Jonagold	197978,6			Szampion	112922,4
12	Jonagored	256949			Cortland	112400
13	Kosztela	59962,5			Gloster	107504,6
14	Ligol	120312,4			Alwa	104124,5
15	Lobo	63970			Idared	100837,2
16	Melba	150130,8			Delikates	99340,8
17	Papierowka	164332,8			Antonowka	79593,6
18	Reneta	389921			Lobo	63970
19	Szampion	112922,4			Kosztela	59962,5
20	Suma końcowa	2422669,4				
21						

Przykład 18.

Fragment rozwiązania zadania 7.2. sposobem drugim przedstawiono na przykładzie 19.

M20		=MAX(K4:K19)													
	A	B	C	D	E	F	G	H	I	J	K	L	M		
1															
2															
3	Etykiety wierszy	Suma z Kilo													
4	Alwa	35905						Alwa	2,9		104124,5				
5	Antonowka	24873						Antonowka	3,2		79593,6				
6	Cortland	35125						Cortland	3,2		112400				
7	Delikates	31044						Delikates	3,2		99340,8				
8	Gala	88938						Gala	3,4		302389,2				
9	Gloster	31619						Gloster	3,4		107504,6				
10	Idared	29658						Idared	3,4		100837,2				
11	Jonagold	58229						Jonagold	3,4		197978,6				
12	Jonagored	73414						Jonagored	3,5		256949				
13	Kosztela	23985						Kosztela	2,5		59962,5				
14	Ligol	35386						Ligol	3,4		120312,4				
15	Lobo	25588						Lobo	2,5		63970				
16	Melba	55604						Melba	2,7		150130,8				
17	Papierowka	51354						Papierowka	3,2		164332,8				
18	Reneta	111406						Reneta	3,5		389921				
19	Szampion	47051						Szampion	2,4		112922,4				
20	Suma końcowa	759179									2422669,4			389921	
21											Reneta				
22															

Przykład 19.

Problem pod lupą

Algorytmiczne podejście do problemu w arkuszu kalkulacyjnym

Zadanie 7.4. okazało się drugim najtrudniejszym zadaniem w arkuszu egzaminacyjnym (poziom wykonania – 11%) zaraz po najtrudniejszym zadaniu programistycznym. Zadanie to nie jest typowym zadaniem polegającym na wykonaniu w arkuszu kalkulacyjnym analizy statystycznej danych, którą można prosto zrealizować za pomocą wbudowanych funkcji. Uzyskanie odpowiedzi, na pytania postawione w treści zadania, wymaga od zdającego podejścia algorytmicznego do rozwiązywania problemu: modelowania sytuacji problemowej, zaplanowania i przeprowadzenia wszystkich etapów prowadzących do otrzymania poprawnego wyniku. Zadania tego typu wymagające np. przeprowadzenia symulacji pojawiają się praktycznie w każdym arkuszu maturalnym z ostatnich lat. Zwykle są umieszczone jako ostatnie zadanie w wiązce zadań sprawdzających umiejętność posługiwania się narzędziami i funkcjami arkusza kalkulacyjnego. Wymagają one skupienia, dokładnej analizy treści zadania i zrozumienia postawionego w zadaniu problemu do rozwiązania.

Poniżej na rysunku 4. przytaczamy wstęp do wiązki zadań 7.1.–7.4. oraz na rysunku 5. treść zadania 7.4.

Zadanie 7. Hurtownia

Pewna hurtownia sprzedaje jabłka. W pliku `jablka.txt` znajduje się 2500 wierszy z danymi dotyczącymi sprzedaży jabłek od 3 stycznia 2022 roku do 31 grudnia 2022 roku. W każdym wierszu podane są dane opisujące jedną transakcję sprzedaży, oddzielone pojedynczymi znakami tabulacji:

- data sprzedaży zapisana w formacie `rrrr-mm-dd`
- nazwa odmiany jabłka
- kod, który określa, czy odmiana jest: L – letnia, J – jesienna czy Z – zimowa
- numer NIP klienta (13-znakowy tekst)
- liczba kilogramów sprzedanego towaru.

Fragment pliku `jablka.txt`:

```
2022-01-03      Jonagold      Z      128-29-15-591  470
2022-01-03      Jonagold      Z      192-09-72-275  410
2022-01-03      Jonagored     Z      140-36-11-559  242
```

Cena sprzedaży jednego kilograma jabłek zależy od odmiany jabłek. W pliku `cennik.txt` jest podana cena w złotych 1 kg jabłek każdej odmiany.

Fragment pliku `cennik.txt`:

```
Alwa           2,9
Antonowka     3,2
Cortland       3,2
```

Rysunek 4.

Zadanie 7.4. (0–3)

Hurtownia ma system premiowania klientów hurtowych. Klient otrzymuje przy zakupie rabat, którego wysokość zależy od łącznej ilości jabłek zakupionych do tej pory przez tego klienta, wliczając w to jabłka zakupione w bieżącej transakcji.

Wysokość rabatu za każdy kilogram w bieżącej transakcji wynosi:

- 5 gr, jeśli klient dotychczas zakupił co najmniej 15 000 kg, ale mniej niż 20 000 kg
- 10 gr, jeśli klient dotychczas zakupił co najmniej 20 000 kg.

Podaj, w ilu transakcjach hurtownia udzieliła rabatu, oraz podaj łączną wartość tych rabatów w złotych.

Przykład:

Założmy, że przed 1.04.2022 klient z NIP 128-29-15-591 zakupił łącznie 10 000 kg jabłek, klient 192-09-72-275 nabył 14 900 kg, a klient 140-36-11-559 – kupił 19 900 kg. Wtedy dla przykładowych danych 1.04.2022:

2022-04-01	Jonagold	Z	128-29-15-591	470
2022-04-01	Jonagold	Z	192-09-72-275	410
2022-04-01	Jonagored	Z	140-36-11-559	242

- pierwszy klient, po dokonaniu transakcji 1.04.2022, będzie miał już 10 470 kg zakupionych jabłek i dla tej transakcji jeszcze nie należy się rabat
- drugi klient, po dokonaniu transakcji 1.04.2022, będzie miał już 15 310 kg zakupionych jabłek, czyli podczas tej transakcji otrzyma 5 gr rabatu na każdy kilogram kupowanych tego dnia jabłek ($5 \text{ gr} * 410 = 20,50 \text{ zł}$ rabatu)
- trzeci klient, po dokonaniu transakcji 1.04.2022, będzie miał już 20 142 kg zakupionych jabłek i podczas tej transakcji otrzyma 10 gr rabatu za każdy kilogram ($10 \text{ gr} * 242 = 24,20 \text{ zł}$ rabatu).

Rysunek 5.

Pierwszym krokiem zdającego było przeanalizowanie treści zadania oraz przestudiowanie podanego przykładu, który powinien ułatwić zrozumienie zagadnienia.

Aby rozwiązać zadanie należało obliczyć dla każdego klienta, w każdej transakcji wartość udzielonego rabatu lub stwierdzić jego brak. W tym celu w pierwszej kolejności należało zliczyć przy każdej transakcji ile kilogramów jabłek dany klient zakupił do danego dnia, łącznie z ilością jabłek zakupioną tego dnia. Najprostszym sposobem wykonania tego kroku było zastosowanie funkcji *suma.jeżeli* w dodatkowej kolumnie – wielu zdających zapomniało o tej przydatnej funkcji i realizowało ten krok sortując dane według numeru klienta. W przypadku tego zadania, o ile zachowano kolejność transakcji według dat, takie sortowanie było poprawne – jednak w innych zadaniach takie postępowanie może zaburzyć wyniki, np. gdy zależą one od kolejności danych zależnych od wielu kolumn. W kolejnym kroku, znając już liczby kilogramów jabłek dotychczas zakupionych przez klientów, należało obliczyć kwotę rabatu w danej transakcji i ostatecznie zliczyć liczbę wszystkich udzielonych rabatów i obliczyć ich sumę.

Przykłady od 20. do 25. przedstawiają kolejne kroki poprawnego rozwiązania zadania, w którym obliczono ilość dotychczas zakupionych jabłek przez klientów za pomocą funkcji *suma.jeżeli*.

SUMA									
=SUMA.JEŻELI(\$D\$2:D3;D3;\$E\$2:E3)									
	A	B	C	D	E	F	G	H	I
1	Data	Nazwa	Kod	NIP	kg	ile zakupione	czy rabat	kwota rabatu	rabat
2	01.03.2022	Jonagold	Z	128-29-15-591	470	470	0	0	0
3	01.03.2022	Jonagold	Z	192-09-72-275	410	=SUMA.JEŻELI(\$D\$2:D3;D3;\$E\$2:E3)			0
4	01.03.2022	Jonagored	Z	140-36-11-559	242	242	0	0	0
5	01.03.2022	Jonagored	Z	053-79-35-388	533	533	0	0	0
6	01.03.2022	Jonagold	Z	159-34-45-151	543	543	0	0	0
7	01.03.2022	Alwa	Z	159-34-45-151	341	884	0	0	0
8	01.03.2022	Ligol	Z	093-96-93-428	284	284	0	0	0
9	01.03.2022	Idared	Z	102-48-01-310	159	159	0	0	0

Przykład 20.

Po ustaleniu dla każdego klienta osobno łącznej liczby kilogramów sprzedanych jabłek w poprzednich i bieżącej transakcji należy zdecydować, czy w danej transakcji należy się klientowi rabat i jaki to będzie rabat. W analizowanym przykładzie zdający zapisał oddzielnie informacje w kolumnach: czy rabat się należy, kwota rabatu (0 groszy, 5 groszy czy 10 groszy) oraz „rabat” (czyli wartość rabatu w danej transakcji). Ponieważ w odpowiedzi do zadania należało podać ile razy udzielono rabatu oraz łączną sumę rabatów, nie zachodziła konieczność przedstawiania tych informacji aż w trzech kolumnach – wystarczyłaby ostatnia, której wartości po zsumowaniu oraz liczba komórek zawierających liczby większe od zera stanowią rozwiązanie zadania.

Na kolejnych trzech przykładach przedstawiono zastosowane funkcje do:

– sprawdzenia, czy należy się rabat (1 – tak, 0 – nie):

SUMA									
= JEŻELI(F2>=15000;1;0)									
	A	B	C	D	E	F	G	H	I
1	Data	Nazwa	Kod	NIP	kg	ile zakupione	czy rabat	kwota rabatu	rabat
2	01.03.2022	Jonagold	Z	128-29-15-591	470	470	= JEŻELI(F2>=15000;1;0)		0
3	01.03.2022	Jonagold	Z	192-09-72-275	410	410	0	0	0
4	01.03.2022	Jonagored	Z	140-36-11-559	242	242	0	0	0

Przykład 21.

– określenia wysokości rabatu za każdy kilogram w bieżącej transakcji (5 groszy lub 10 groszy):

SUMA												
= JEŻELI(ORAZ(F2>=15000; F2<20000);5;JEŻELI(F2>=20000;10;0))												
	A	B	C	D	E	F	G	H	I	J	K	L
1	Data	Nazwa	Kod	NIP	kg	ile zakupione	czy rabat	kwota rabatu	rabat			ile rabatow udzie
2	01.03.2022	Jonagold	Z	128-29-15-591	470	470	0	= JEŻELI(ORAZ(F2>=15000; F2<20000);5;JEŻELI(F2>=20000;10;0))				
3	01.03.2022	Jonagold	Z	192-09-72-275	410	410	0	0	0			JEŻELI(test_logiczny; [warto

Przykład 22.

– obliczenia wartości udzielonego rabatu w zależności od ilości zakupionych jabłek w danej transakcji:

SUMA									
A	B	C	D	E	F	G	H	I	J
1	Data	Nazwa	Kod	NIP	kg	ile zakupione	czy rabat	kwota rabatu	rabat
2	01.03.2022	Jonagold	Z	128-29-15-591	470	470	0	0	=E2*G2*H2
3	01.03.2022	Jonagold	Z	192-09-72-275	410	410	0	0	0

Przykład 23.

– obliczenia liczby transakcji, w których udzielono rabatów (sumowanie wartości w kolumnie „czy rabat”):

=SUMA(G2:G2501)									
D	E	F	G	H	I	J	K	L	
NIP	kg	ile zakupione	czy rabat	kwota rabatu	rabat				ile rabatow udzielono
128-29-15-591	470	470	0	0	0				=SUMA(G2:G2501)
192-09-72-275	410	410	0	0	0				SUMA(liczba1; [liczba2]; ...)
140-36-11-559	242	242	0	0	0				

Przykład 24.

– obliczenia łącznej wartości udzielonych rabatów (sumowanie wartości w kolumnie „rabat”):

=SUMA(I2:I2501)*0,01												
C	D	E	F	G	H	I	J	K	L	M	N	O
od	NIP	kg	ile zakupione	czy rabat	kwota rabatu	rabat			ile rabatow udzielono		suma rabatu	
	128-29-15-591	470	470	0	0	0			180		=SUMA(I2:I2501)*0,01	
	192-09-72-275	410	410	0	0	0					SUMA(liczba1; [liczba2]; ...)	
	140-36-11-559	242	242	0	0	0						
	053-79-35-388	533	533	0	0	0						
	150-24-45-153	543	543	0	0	0						

Przykład 25.

Przykłady od 26. do 30. przedstawiają kolejne kroki rozwiązania zadania, w którym zdający nie korzysta z funkcji *suma.jeżeli*.

Po zastosowaniu sortowania transakcji według numeru NIP, a następnie według daty, w kolumnie F określono, czy następuje zmiana klienta, a w kolumnie G zliczono liczby kilogramów dotychczas zakupionych jabłek.

SUMA							
=JEŻELI(F3>0;E3;G2+E3)							
A	B	C	D	E	F	G	
1	data	odmiana	kod	nip	kilogramy	czy ten sam klient	łączna ilosc
2	05.01.2022	Ligol	Z	014-02-05-290	110	1	110
3	10.01.2022	Jonagold	Z	014-02-05-290	695	0	=JEŻELI(F3>0;E3;G2+E3)
4	10.01.2022	Szampion	Z	014-02-05-290	232	0	1037
5	19.01.2022	Gloster	Z	014-02-05-290	11	0	1048
6	22.01.2022	Jonagored	Z	014-02-05-290	590	0	1638

Przykład 26.

W kolumnie H obliczono wartość udzielonego rabatu w groszach:

=JEŻELI(ORAZ(G2>=15000;G2<20000);E2*5;JEŻELI(G2>=20000;E2*10;0))									
C	D	E	F	G	H	I	J	K	L
kod	nip	kilogramy	czy ten sam klient	łączna ilosc	rabat (w groszach) czy rabat	suma rabatu			
Z	014-02-05-290	110	1	110	=JEŻELI(ORAZ(G2>=15000;G2<20000);E2*5;JEŻELI(G2>=20000;E2*10;0))				
Z	014-02-05-290	695	0	805	0	0			

Przykład 27.

W celu zliczenia liczby udzielonych rabatów, w kolumnie I zawarto informacje czy rabat został udzielony (1 – tak, 0 – nie):

=JEŻELI(H2>0;1;0)

C	D	E	F	G	H	I
cod	nip	kilogramy	czy ten sam klient	łączna ilość	rabat (w gorszach)	czy rabat
	014-02-05-290	110	1	110	0	=JEŻELI(H2>0;1;0)
	014-02-05-290	695	0	805	0	0
	014-02-05-290	232	0	1037	0	0
	014-02-05-290	11	0	1048	0	0

Przykład 28.

Po obliczeniu sumy wartości z kolumny I otrzymano liczbę udzielonych rabatów:

=SUMA(I2:I2501)

C	D	E	F	G	H	I	J	K
cod	nip	kilogramy	czy ten sam klient	łączna ilość	rabat (w gorszach)	czy rabat	suma rabat	
	014-02-05-290	110	1	110	0	0	3 089,10 zł	
	014-02-05-290	695	0	805	0	0		
	014-02-05-290	232	0	1037	0	0	ile razy rabat	
	014-02-05-290	11	0	1048	0	0	=SUMA(I2:I2501)	

Przykład 29.

Natomiast iloraz sumy wartości w kolumnie H przez 100, to łączna wartość rabatów wyrażona w złotych:

=SUMA(H2:H2501)/100

C	D	E	F	G	H	I	J	K
cod	nip	kilogramy	czy ten sam klient	łączna ilość	rabat (w gorszach)	czy rabat	suma rabat	
	014-02-05-290	110	1	110	0	0	=SUMA(H2:H2501)/100	

Przykład 30.

Zdający często nie podejmowali próby rozwiązania zadania 7.4. Możliwym powodem może być zbyt małe oswojenie się z zadaniami tego typu u zdających.

Wnioski i rekomendacje

1. W tym roku po raz pierwszy egzamin w nowej formule zdawali absolwenci techników. Uzyskany przez nich średni wynik – podobnie jak w poprzednich latach na egzaminach w formule 2015 – jest niższy niż średni wynik absolwentów liceów.
2. Poziom wykonania poszczególnych zadań wskazuje, że w arkuszu nie było zadań określanych jako łatwe lub bardzo łatwe dla zdających. Największe problemy, podobnie jak w poprzednich latach, sprawiły zadania wymagające zapisu algorytmu lub programu oraz zadania symulacyjne w arkuszu kalkulacyjnym. Zmiana formuły egzaminu nie wpłynęła na zmiany poziomów trudności pomiędzy rodzajami zadań.
3. Wydaje się możliwe, że część zadań, zwłaszcza wymagających większej uwagi i wysiłku, zostało pominiętych z powodu nieodpowiedniego rozplanowania czasu. Obecna formuła arkusza maturalnego z informatyki, w której nie oddziela się części teoretycznej od praktycznej, daje większe możliwości rozplanowania czasu poświęcanego na różne zadania i dobrze by było przyszłym absolwentom zwrócić na to uwagę.
4. Zadania wymagające zapisu algorytmu w arkuszu egzaminacyjnym nadal są jednymi z trudniejszych dla zdających. W tych zadaniach należy zwrócić uwagę na kilka aspektów:
 - jeżeli zdający decyduje się na zapis w języku programowania, ma obowiązek użyć języka zadeklarowanego w deklaracji maturalnej
 - zapisując algorytm, należy dokładnie przestrzegać warunków zadania, w tym uwag do niego dołączonych, które wykluczają np. użycie funkcji wbudowanych języka programowania, czy konkretnych instrukcji.
5. Arkusz maturalny z informatyki w większości składa się z zadań praktycznych wymagających użycia komputera. Przyszli zdający powinni więc kształcić umiejętność rozwiązywania zadań praktycznych, w szczególności poprzez:
 - ćwiczenia polegające na rozwiązywaniu typowych zadań praktycznych, tak by nie tracić czasu na rozwiązywanie zadań niewymagających dużej inwencji
 - częste podejmowanie prób rozwiązywania zadań, w których należy przeprowadzić symulację
 - dokładną analizę treści zadań oraz dobór odpowiednich narzędzi do ich rozwiązywania
 - omawianie popełnianych błędów i wskazywanie przyczyny tych błędów.
6. Odpowiedzi do zadań praktycznych zapisywane w pliku z wynikami muszą być odzwierciedleniem komputerowej realizacji obliczeń. Przygotowując uczniów do egzaminu, należy:
 - zwracać uwagę na konieczność czytelnego wskazania komputerowej realizacji zadania, np.: wpisywanie nazw plików w przeznaczonym do tego miejscu, umieszczanie rozwiązania każdego zadania w odrębnej, odpowiednio nazwanej zakładce arkusza kalkulacyjnego, zapisywanie kwerend pod odpowiednimi nazwami itp.
 - przeanalizować zapisy dotyczące formy, w jakiej należy przekazać rozwiązania zadań, np. zapisy w arkuszach z lat ubiegłych, tj. *Instrukcja dla zdającego*, sekcja *Do oceny oddajesz*.