KLUCZ ODPOWIEDZI · DLA NAUCZYCIELA

Sprawdzian z systemów liczbowych

Grupa A

1. (1 pkt) **C**

2. (1 pkt) **D**

3. (3 pkt) **1. F; 2. P; 3. F** — po 1 pkt za zdanie

4. (4 pkt) **1–D, 2–C, 3–B, 4–A** — po 1 pkt za parę

5. (4 pkt) **1–A, 2–C, 3–D, 4–B** — po 1 pkt za parę

6. (2 pkt) **(1) ASCII; (2) 65** — po 1 pkt za lukę

7. (3 pkt) Przykładowa odpowiedź: Zamiana na dziesiętne: $10110011_2 = 128 + 32 + 16 + 2 + 1 = 179$, $01001100_2 = 64 + 8 + 4 = 76$. Suma: $179 + 76 = 255 = 11111111_2$. Weryfikacja: $11111111_2 = 2^8 - 1 = 255$.
Punktacja: 1. Przeliczenie lub dodawanie kolumnowe z przeniesieniami (1 pkt). 2. Wynik dwójkowy: 11111111_2 (1 pkt). 3. Wynik dziesiętny: 255 (1 pkt).

8. (3 pkt) Przykładowa odpowiedź: Liczby zmiennoprzecinkowe IEEE 754 zapisują wartość jako iloczyn mantysy i potęgi dwójki, a mantysa ma ograniczoną, stałą liczbę bitów (np. 23 bity dla pojedynczej precyzji). Wiele ułamków dziesiętnych, np. $0,1$, nie da się zapisać dokładnie w systemie binarnym, więc są zaokrąglane do najbliższej reprezentowalnej wartości. Jeśli dodamy bardzo małą liczbę (np. 10^{-10}) do bardzo dużej (np. 10^{10}), różnica rzędów wielkości sprawia, że mała liczba nie mieści się w precyzji mantysy wyniku i zostaje zaokrąglona do zera względem dużej liczby — wynik pozostaje praktycznie niezmieniony.
Punktacja: Poprawne wyjaśnienie skończonej liczby bitów mantysy jako źródła utraty precyzji — 1pkt. Odniesienie do ułamków dziesiętnych niereprezentowalnych dokładnie binarnie (np. $0,1$) — 1pkt. Trafny przykład dodania bardzo małej liczby do bardzo dużej bez zmiany wyniku — 1pkt. Maks. 3 pkt.

9. (3 pkt) Przykładowa odpowiedź: W kodzie uzupełnień do dwóch (U2) najstarszy bit ma wagę ujemną — dla liczby n-bitowej wynosi -2^{n-1}, a pozostałe bity mają zwykłe, dodatnie wagi potęg dwójki. Aby zapisać liczbę ujemną, bierze się zapis binarny jej wartości bezwzględnej, neguje się wszystkie bity, a następnie dodaje się 1. Dla liczby 5 na 8 bitach: $5 = 00000101_2$. Negacja bitów daje 11111010. Dodanie 1 daje 11111011. Zatem -5 w kodzie U2 na 8 bitach to 11111011_2.
Punktacja: 1 pkt za wyjaśnienie idei U2 (najstarszy bit ma ujemną wagę, lub opis przez negację bitową i dodanie 1). 1 pkt za poprawny opis algorytmu zamiany liczby dodatniej na jej reprezentację ujemną w U2. 1 pkt za poprawny wynik dla -5 na 8 bitach. Maks. 3 pkt.

10. (3 pkt) Przykładowa odpowiedź: W systemie U2 liczbę ujemną otrzymuje się, zapisując najpierw odpowiadającą jej liczbę dodatnią w postaci binarnej, następnie negując wszystkie bity (zamieniając 0 na 1 i odwrotnie), a na koniec dodając 1 do wyniku. Dla -5 na 8 bitach: $5_{10} = 00000101_2$. Negacja bitowa: 11111010_2. Dodanie 1: $11111010_2 + 1 = 11111011_2$. Zatem -5 w systemie U2 na 8 bitach to 11111011_2.
Punktacja: Poprawne wyjaśnienie idei U2 — negacja bitowa liczby dodatniej i dodanie 1 (1 pkt) + poprawny zapis liczby 5 na 8 bitach jako punkt wyjścia (1 pkt) + poprawny końcowy wynik dla -5 wraz z pokazaniem obliczenia (1 pkt). Maksymalnie 3 pkt.

Grupa B

1. (1 pkt) **A**

2. (1 pkt) **A**

3. (3 pkt) **1. P; 2. P; 3. F** — po 1 pkt za zdanie

4. (4 pkt) **1–C, 2–D, 3–A, 4–B** — po 1 pkt za parę

5. (4 pkt) **1–A, 2–C, 3–D, 4–B** — po 1 pkt za parę

6. (2 pkt) **(1) 64; (2) AA** — po 1 pkt za lukę

7. (3 pkt) Przykładowa odpowiedź: 205 : 16 = 12, reszta 13. 12 : 16 = 0, reszta 12. Odczytując reszty od dołu do góry: 12, 13, czyli w systemie szesnastkowym cyfry C (12) i D (13). Wynik: 205 = CD_{16}.
Punktacja: Poprawnie wykonane kolejne dzielenia przez 16 z zapisem reszt (1 pkt) + poprawne odczytanie cyfr wyniku od dołu, w tym zamiana 12→C i 13→D (1 pkt) + poprawny wynik końcowy: CD_{16} (1 pkt). Razem 3 pkt.

8. (3 pkt) Przykładowa odpowiedź: Kod U2 pozwala reprezentować ujemne liczby całkowite: (1) zapisujemy wartość bezwzględną binarnie: $13_{10} = 00001101_2$; (2) negujemy wszystkie bity (NOT): 11110010_2; (3) dodajemy 1: $11110010_2 + 1 = 11110011_2$. Wynik: $-13_{10} = 11110011_{U2}$. Zakres 8-bitowego U2: od -128 do $+127$.
Punktacja: Definicja U2 (metoda reprezentacji ujemnych liczb całkowitych w binarnym systemie komputerowym) — 1 pkt. Poprawne obliczenie: $13_{10} = 00001101_2$; negacja bitowa → 11110010_2; +1 → 11110011_2 — 1 pkt. Poprawny wynik końcowy $-13_{10} = 11110011_2$ (U2) — 1 pkt.

9. (3 pkt) Przykładowa odpowiedź: Nadmiar (overflow) w kodzie U2 na n bitach występuje, gdy wynik dodawania dwóch liczb wykracza poza zakres reprezentowalny na tych n bitach (dla 4 bitów w U2 zakres to $[-8, 7]$). Nadmiar sygnalizuje sytuacja, w której dodajemy dwie liczby o tym samym znaku (obie dodatnie lub obie ujemne), a otrzymany wynik ma znak przeciwny — to znak, że „nie zmieścił się" w dostępnej liczbie bitów. Przykład: na 4 bitach liczba 5 to `0101`, a liczba 4 to `0100`. Ich suma binarna `0101+0100=1001`, co w U2 na 4 bitach odczytuje się jako -7, mimo że matematycznie $5 + 4 = 9$ — wystąpił nadmiar, bo wynik 9 przekracza maksymalną wartość 7 możliwą do zapisania na 4 bitach w U2.
Punktacja: Poprawne wyjaśnienie, że nadmiar występuje, gdy matematyczny wynik dodawania nie mieści się w zakresie reprezentowalnym przez daną liczbę bitów w U2 (1 pkt) + wskazanie, że sygnałem nadmiaru jest dodanie dwóch liczb tego samego znaku i otrzymanie wyniku o przeciwnym znaku (1 pkt) + poprawny przykład liczbowy (1 pkt). 3 pkt.

10. (3 pkt) Przykładowa odpowiedź: W reprezentacji znak-moduł najstarszy bit określa znak liczby (0 — dodatnia, 1 — ujemna), a pozostałe bity zapisują wartość bezwzględną wprost w systemie binarnym. W kodzie U2 nie wyodrębnia się osobnego bitu znaku — najstarszy bit ma po prostu ujemną wagę (np. dla 8 bitów wagę -128), dzięki czemu dodawanie liczb dodatnich i ujemnych realizuje ten sam obwód sumatora bez rozróżniania przypadków. Wadą znak-moduł jest istnienie dwóch różnych zapisów zera ($+0$ i -0), czego U2 nie ma — tam zero ma tylko jedną reprezentację.
Punktacja: Poprawny opis znak-moduł (najstarszy bit = znak, pozostałe bity = wartość bezwzględna) — 1pkt. Poprawny opis U2 (najstarszy bit ma ujemną wagę, brak osobnego bitu znaku) — 1pkt. Trafna wada znak-moduł (np. dwie reprezentacje zera, bardziej złożone dodawanie) — 1pkt. Maks. 3 pkt.

Skala ocen (orientacyjna, 27 pkt w każdej grupie)

bardzo dobry (5)	25–27 pkt
dobry (4)	21–24 pkt
dostateczny (3)	14–20 pkt
dopuszczający (2)	9–13 pkt
niedostateczny (1)	0–8 pkt

Progi: 90%, 75%, 50%, 30%. Progi w Twojej szkole mogą być inne. W zadaniach otwartych uznaj każdą odpowiedź poprawną merytorycznie, także sformułowaną inaczej niż przykład.

Ten sprawdzian online, z wyjaśnieniem każdej odpowiedzi: **matury-online.pl/sprawdzian/informatyka/systemy-liczbowe**. Nauczyciele i korepetytorzy ułożą własną kartę z banku ponad 20 tys. zadań: **matury-online.pl/dla-korepetytorow**.