
PRÓBNY EGZAMIN MATURALNY Z INFORMATYKI STYCZEŃ 2022

Arkusz I

Czas pracy: **60 minut**

Liczba punktów do uzyskania: **15**

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 9 stron (zadania 1 – 3). Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
2. Rozwiązania i odpowiedzi zamieść w miejscu na to przeznaczonym.
3. Pisz czytelnie. Używaj długopisu/pióra tylko z czarnym tuszem/atramentem.
4. Nie używaj korektora, a błędne zapisy wyraźnie przekreśl.
5. Pamiętaj, że zapisy w brudnopisie nie podlegają ocenie.
6. Wpisz poniżej zadeklarowane (wybrane) przez Ciebie na egzamin środowisko komputerowe, kompilator języka programowania oraz program użytkowy.
7. Jeżeli rozwiązaniem zadania lub jego części jest algorytm, to zapisz go w wybranej przez siebie notacji: listy kroków, pseudokodu lub języka programowania, który wybrałeś/eś na egzamin.

Dane uzupełnia uczeń:

WYBRANE:

.....

(środowisko)

.....

(kompilator)

.....

(program użytkowy)

PESEL:

--	--	--	--	--	--	--	--	--	--	--	--

Klasa:

--	--	--

Zadanie 1. Test (0-5)

Oceń, czy poniższe zdania są prawdziwe. Zaznacz P, jeśli zdanie jest prawdziwe, albo F, jeśli zdanie jest fałszywe. W każdym zadaniu uzyskasz punkt, jeśli poprawnie odpowiesz na wszystkie jego części.

Zadanie 1.1. (0–1)

Sortując pewnym algorytmem ciąg: 7 4 5 4 2 3 5 1 9
po dwóch iteracjach tego algorytmu trzy pierwsze i trzy ostatnie wyrazy miały postać:
4 4 2 X X X 5 7 9

Jaki zastosowano algorytm?

przez wstawianie	P	F
przez wybór	P	F
algorytm bąbelkowy	P	F
przez scalanie	P	F

Zadanie 1.2. (0–1)

Jaś i Joasia używają algorytmu szyfrowania kluczem asymetrycznym. Jaś zaszyfrował poprawnie wiadomość dla Joasi, a Joasia ją odszyfrowała. Oznacza to, że:

Joasia odszyfrowała wiadomość swoim kluczem prywatnym.	P	F
Joasia odszyfrowała wiadomość swoim kluczem publicznym.	P	F
Jaś zaszyfrował wiadomość kluczem prywatnym Joasi	P	F
Jaś zaszyfrował wiadomość kluczem publicznym Joasi	P	F

Zadanie 1.3. (0–1)

W turnieju tenisowym rozgrywanym metodą pucharową startuje n zawodników. Każdy mecz jest rozgrywany jedną nową piłką. Ile piłek potrzeba do wyłonienia zwycięzcy?

n	P	F
$n - 1$	P	F
$n \log n$	P	F
n^2	P	F

Zadanie 1.4. (0–1)

Jaką postać w systemie binarnym ma dziesiętna liczba 5.5?

101.101	P	F
1.1	P	F
1.101	P	F
101.1	P	F

Zadanie 1.5. (0–1)

W relacyjnej bazie danych istnieje tabela `zamowienie`, która zawiera następujące dane:

id	Towar	sztuk	cena_jednostkowa
1	Procesor i5-9600k	1	900
2	Płyta główna Z390	1	1400
3	Karta graficzna RTX3070	2	5500
4	Dysk NVMe 1TB	1	500
5	Pamięć RAM DDR4 16GB	4	400

Wynikiem zapytania SELECT SUM(cena_jednostkowa) FROM zamowienie WHERE sztuk = 2; jest 5500	P	F
Wynikiem zapytania SELECT SUM(sztuk) FROM zamowienie WHERE sztuk = 1; jest 3	P	F
Wynikiem zapytania SELECT SUM(sztuk*cena_jednostkowa) FROM zamowienie; jest wartość całego zamówienia, tj. suma iloczynów liczby sztuk i ceny jednostkowej.	P	F
Wynikiem zapytania SELECT cena_jednostkowa FROM zamowienie WHERE id = 3; jest cena jednostkowa karty graficznej z zamówienia.	P	F

Wypełnia egzaminator	Numer zadania	1.1	1.2	1.3	1.4	1.5	Suma
	Maksymalna liczba punktów	1	1	1	1	1	5
	Uzyskana liczba punktów						

Zadanie 2. Liczby dwupierwsze (0-5)

Niech $n \geq 10$ będzie liczbą naturalną. Powiemy, że n jest *liczbą dwupierwszą*, o ile liczba powstała poprzez wykreślenie cyfr na pozycjach parzystych (najmniej znacząca cyfra ma numer 1) oraz liczba powstała poprzez wykreślenie cyfr na pozycjach nieparzystych są liczbami pierwszymi.

Na przykład, liczba 1213 jest liczbą *dwupierwszą*. Po wykreśleniu cyfr na pozycjach parzystych dostajemy liczbę 23 (która jest pierwsza), natomiast po wykreśleniu cyfr na pozycjach nieparzystych dostajemy liczbę 11 (która jest pierwsza).

Liczby *dwupierwsze* można łatwo generować – mając dane dwie liczby pierwsze o liczbie cyfr różniące się o co najwyżej jeden, wystarczy na zmianę zapisywać cyfry jednej i drugiej. Nietrudno się zatem przekonać, że najmniejszą liczbą *dwupierwszą* jest 22.

Uwaga. W algorytmach do zadań 2.2 i 2.3 może używać wyłącznie zmiennych przechowujących liczby całkowite (w tym tablic i list) oraz może operować wyłącznie na liczbach całkowitych. W zapisie algorytmu możesz korzystać tylko z instrukcji sterujących, operatorów arytmetycznych: dodawania, odejmowania, mnożenia, dzielenia całkowitego, reszty z dzielenia; operatorów logicznych, porównań i instrukcji przypisywania lub samodzielnie napisanych funkcji wykorzystujących powyższe operacji. Zabronione jest używanie funkcji wbudowanych w dostępnych językach programowania.

Zadanie 2.1. (0–1)

Dla podanych dwóch liczb pierwszych podaj **największą** liczbę *dwupierwszą*, jaką można z nich wygenerować. Dla liczb pierwszych p i q będziemy oznaczać przez \overline{pq} największą liczbę *dwupierwszą* generowaną przez p i q .

p	q	\overline{pq}
61	67	
83	19	
7	71	

Zadanie 2.2. (0–2)

Opracuj algorytm generujący wszystkie trzycyfrowe liczby *dwupierwsze* i zapisz go w wybranej przez siebie notacji (schemat blokowy, pseudokod, lista kroków, język programowania). Algorytm może wypisać je w dowolnej kolejności.

Specyfikacja:

Dane: algorytm nie przyjmuje żadnych danych wejściowych

Wynik: Wszystkie trzycyfrowe liczby *dwupierwsze*

Zadanie 3. Flip-sort (0-5)

Dana jest tablica **tab** n liczb całkowitych indeksowana od jedynek. (W niektórych językach programowania zamiast tablicy używa się listy.) Dla tej tablicy definiujemy operację **flip(a, b)**, która jako parametry przyjmuje dwie liczby naturalne określające dwa indeksy w tablicy **tab**, przy czym zakładamy, że: $a \leq b$ oraz $1 \leq a, b \leq n$. Operacja **flip** odwraca kolejność elementów w tablicy od indeksu **a** do indeksu **b**.

Przykład: dla tablicy **tab**=[3,2,1,4] po wykonaniu operacji **flip(1,4)** otrzymamy **tab**=[4,1,2,3].

Zadanie 3.1. (0–1)

Uzupełnij poniższą tabelkę. Dla każdej wartości początkowej tablicy **tab** i operacji **flip(a,b)** podaj zawartość tablicy **tab** po wykonaniu tej operacji.

Tab	flip(a,b)	Tablica po wykonaniu operacji flip
[5,8,4,2,9,1,3]	flip(1,6)	[1,9,2,4,8,5,3]
[1,4,8,3,8,2,9]	flip(2,6)	[1,2,8,3,8,4,9]
[1,2,7,6,3,9,7]	flip(3,5)	
[1,2,3,7,9,4,5]	flip(4,6)	

Zadanie 3.2. (0–1)

Operację **flip** możemy stosować wielokrotnie na tablicy, aby w efekcie uzyskać tablicę posortowaną niemalejąco. W praktyce otrzymujemy algorytm sortowania podobny do algorytmu sortowania przez wybieranie. Uzupełnij poniższą tabelkę podając sekwencję operacji **flip** jakie należy wykonać, aby posortować zadaną tablicę **niemalejąco**.

Tablica tab	Sekwencja operacji flip
[2, 3, 1, 7, 5]	flip(1,3), flip(2,3), flip(4,5)
[4, 7, 1, 2, 3]	

Zadanie 3.3. (0–1)

Uzupełnij poniższe zdanie.

Dla dowolnej n elementowej tablicy liczb całkowitych wystarczy wykonać co najwyżej

.....

operacji flip, aby posortować tablicę niemalejąco.

BRUDNOPIS (*nie podlega ocenie*)