

WYPEŁNIA ZDAJĄCY

KOD

--	--	--

PESEL

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

*miejsce
na naklejkę*

EGZAMIN MATURALNY Z INFORMATYKI
POZIOM ROZSZERZONY
CZĘŚĆ I



MIN-R1_1P-202

DATA: **11 maja 2020 r.**

GODZINA ROZPOCZĘCIA: **14:00**

CZAS PRACY: **60 minut**

LICZBA PUNKTÓW DO UZYSKANIA: **15**

WYPEŁNIA ZDAJĄCY

WYBRANE:

.....
(system operacyjny)

.....
(program użytkowy)

.....
(środowisko programistyczne)

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 10 stron. 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 zadeklarowane (wybrane) przez Ciebie na egzamin system operacyjny, program użytkowy oraz środowisko programistyczne.
7. Jeżeli rozwiązaniem zadania lub jego części jest algorytm, to zapisz go w notacji wybranej przez siebie: listy kroków, pseudokodu lub języka programowania, który wybierasz na egzamin.
8. Na tej stronie oraz na karcie odpowiedzi wpisz swój numer PESEL i przyklej naklejkę z kodem.
9. Nie wpisuj żadnych znaków w części przeznaczonej dla egzaminatora.

NOWA FORMUŁA

Zadanie 1. Podobne tablice

Niech n będzie dodatnią liczbą całkowitą, a $A[1..n]$ i $B[1..n]$ będą n -elementowymi tablicami liczb całkowitych.

Dla nieujemnej liczby całkowitej k , gdzie $k < n$, powiemy, że tablice A i B są **k -podobne**, gdy $A[1..k] = B[n-k+1..n]$ oraz $A[k+1..n] = B[1..n-k]$.

Liczbę k nazywamy świadectwem podobieństwa.

Uwaga: dla $k = 0$ przyjmujemy, że prawdziwe jest $A[1..0] = B[n+1..n]$.

Zadanie 1.1. (0–1)

Uzupełnij tabelę – wpisz w pustych kratkach odpowiednie wartości. W wierszu piątym i siódmym wpisz słowo *PRAWDA*, jeśli tablice A i B są k -podobne przy podanym k , albo *FAŁSZ* w przeciwnym przypadku. W wierszu szóstym wpisz takie k , dla którego tablice A i B są k -podobne.

Lp.	n	Tablica A	Tablica B	k	Odpowiedź
1.	3	[5, 7, 9]	[5, 7, 9]	0	PRAWDA
2.	5	[4, 7, 1, 4, 5]	[1, 4, 5, 4, 7]	2	PRAWDA
3.	5	[10, 9, 12, 10, 9]	[10, 10, 9, 9, 12]	3	FAŁSZ
4.	5	[3, 6, 5, 1, 8]	[5, 1, 8, 3, 6]	4	FAŁSZ
5.	5	[1, 2, 3, 4, 5]	[3, 4, 5, 1, 2]	2	
6.	9	[1,1,1,1,3,1,1,1,1]	[3,1,1,1,1,1,1,1,1]		PRAWDA
7.	6	[4, 2, 4, 4, 2, 6]	[4, 4, 2, 6, 4, 2]	1	

Zadanie 1.2. (0–3)

Zapisz w wybranej przez siebie notacji (w postaci pseudokodu, listy kroków lub w wybranym języku programowania) funkcję *czy_k_podobne*(n, A, B, k), gdzie A i B są n -elementowymi tablicami liczb całkowitych. Wynikiem funkcji jest *PRAWDA*, jeśli tablice A i B są k -podobne dla zadanego parametru k , natomiast *FAŁSZ* – w przeciwnym przypadku.

Uwaga: w zapisie możesz wykorzystać tylko operacje arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie, dzielenie całkowite, reszta z dzielenia), odwoływanie się do pojedynczych elementów tablicy, porównywanie liczb, instrukcje sterujące i przypisania do zmiennych lub samodzielnie napisane funkcje zawierające wyżej wymienione operacje.

Specyfikacja:

Dane:

- n – dodatnia liczba całkowita
- $A[1..n], B[1..n]$ – n -elementowe tablice liczb całkowitych
- k – nieujemna liczba całkowita mniejsza niż n

Wynik:

PRAWDA, jeśli tablice A i B są k -podobne dla podanego parametru k
FAŁSZ w przeciwnym przypadku.

Zadanie 1.3. (0–2)

Zapisz w wybranej przez siebie notacji funkcję `czy_podobne( $n, A, B$ )`, która dla danych tablic A i B daje odpowiedź *PRAWDA*, jeśli istnieje takie k , dla którego tablice A i B są k -podobne, natomiast *FALSZ* – w przeciwnym przypadku.

Uwaga: w zapisie możesz skorzystać jedynie z operacji wymienionych w zadaniu 1.2. oraz funkcji `czy_k_podobne(n, A, B, k)` opisaną w zadaniu 1.2.

Specyfikacja:

Dane:

n	– dodatnia liczba całkowita
$A[1..n], B[1..n]$	– n -elementowe tablice liczb całkowitych

Wynik:

PRAWDA, jeśli istnieje takie k ($0 \leq k < n$), dla którego tablice A i B są k -podobne
FAŁSZ w przeciwnym przypadku.

Algorytm:

This image shows a full page of blank graph paper. The grid consists of small, uniform squares formed by thin, light gray lines. There are no margins, text, or other markings on the page.

Zadanie 2.2. (0–3)

Uzupełnij tabelę – podaj długość ciągu liczbowego otrzymanego w wyniku wywołania procedury $\text{sym}(a, b)$ dla wskazanych argumentów a i b .

a	b	$\text{sym}(a, b)$
3	2	7
4	4	15
5	1	
6	6	
10	2020	

Miejsce na obliczenia:

This image shows a full page of blank graph paper. The grid consists of small, uniform squares formed by thin, light gray lines. There are no margins, text, or other markings on the page.

Zadanie 3. Test

Oceń prawdziwość podanych zdań. Zaznacz **P**, jeśli zdanie jest prawdziwe, albo **F** – jeśli jest fałszywe. W każdym zadaniu punkt uzyskasz tylko za komplet poprawnych odpowiedzi.

Zadanie 3.1. (0–1)

Czas można odczytywać na zegarach tradycyjnych i na zegarach binarnych. Poniżej zamieszczono przykładowy sposób zapisu godziny 12:46:39 na zegarze binarnym:

godzina	minuty	sekundy
☐	☐	☒
☐	☒	☒
☐	☒	☒
☒	☐	☒

- każda kolumna odpowiada jednej cyfrze zapisu dziesiętnego godziny przedstawionej w postaci binarnej
- czarny kwadrat (np. dioda zegara świeci) oznacza **1**
- biały kwadrat (np. dioda zegara nie świeci) oznacza **0**
- kwadraty w najniższym wierszu odpowiadają najmniej znaczącym cyfrom zapisu binarnego.

Zegar A

godzina	minuty	sekundy
☐	☐	☒
☐	☐	☒
☒	☒	☐
☐	☐	☒

Zegar B

godzina	minuty	sekundy
☐	☒	☐
☒	☒	☒
☐	☒	☒
☐	☒	☒

Zegar C

godzina	minuty	sekundy
☐	☐	☐
☐	☒	☒
☐	☐	☒
☒	☐	☐

Zegar D

godzina	minuty	sekundy
☐	☒	☐
☒	☒	☒
☐	☒	☒
☒	☒	☒

1.	Zegar A wskazuje godzinę 22:30:48.	P	F
2.	Zegar B wskazuje godzinę 07:58:35.	P	F
3.	Zegar C wskazuje godzinę 10:44:24.	P	F
4.	Zegar D wskazuje godzinę 17:48:54.	P	F

Wypełnia egzaminator	Nr zadania	2.2.	3.1.
	Maks. liczba pkt.	3	1
	Uzyskana liczba pkt.		

Zadanie 3.2. (0–1)

W poniższym algorytmie n jest nieujemną liczbą całkowitą, mod to operator reszty z dzielenia, div to operator dzielenia całkowitego.

$w \leftarrow 0$

dopóki $n \neq 0$ **wykonuj**

$w \leftarrow w + (n \text{ mod } 10)$

$n \leftarrow n \text{ div } 10$

1.	Po wykonaniu algorytmu dla $n = 45778$ zmienna w przyjmuje wartość 30.	P	F
2.	Po wykonaniu algorytmu dla liczby n wartością zmiennej w jest suma cyfr liczby n w zapisie dziesiętnym.	P	F
3.	Podczas wykonywania algorytmu dla $n = 1234$ w kolejnych iteracjach pętli <i>dopóki</i> , zmienna w przyjmuje wartości 1, 3, 6, 10.	P	F
4.	Po wykonaniu algorytmu dla $n = 11111$ zmienna w przyjmuje wartość 5.	P	F

Zadanie 3.3. (0–1)

Dana jest konfiguracja interfejsu sieciowego komputerów A i B.

Dla komputera A:

Adres IPv4: 192.168.10.65

Maska sieci: 255.255.255.0

Dla komputera B:

Adres IPv4: 192.168.10.128

Maska sieci: 255.255.255.0

1.	Komputer A i komputer B są w tej samej sieci.	P	F
2.	Adresem sieci dla komputera A jest adres 192.168.10.0.	P	F
3.	Dla maski 255.255.255.0 są dostępne 254 adresy hostów.	P	F
4.	Adres rozgłoszeniowy sieci, do której należy komputer B, to 192.168.255.255.	P	F

Zadanie 3.4. (0–1)Dana jest tabela *pracownicy*

nr_p	nazwisko	imie	staz	pensja
736	Smitko	Alan	10	2000
7499	Nowak	Kazimierz	15	3000
7521	Więcek	Mariusz	11	3500
7566	Jonas	Kamil	12	2500
7654	Martin	Leon	20	2300
7698	Bracki	Bartosz	15	1530
7782	Celerek	Agnieszka	12	1680
7788	Skotnik	Natalia	21	2000
7839	King	Mirosław	22	1500

oraz zapytanie SQL:

```
SELECT *
FROM pracownicy
WHERE pensja < 2000
ORDER BY staz DESC;
```

1.	Wynik powyższego zapytania to 3.					P	F
2.	Wynikiem powyższego zapytania jest zestawienie:					P	F
	nr_p	nazwisko	imie	staz	pensja		
	7839	King	Mirosław	22	1500		
	7698	Bracki	Bartosz	15	1530		
	7782	Celerek	Agnieszka	12	1680		
3.	Wynikiem zapytania będą wiersze z tabeli pracownicy, zawierające wszystkie dane z tej tabeli dotyczące pracowników, dla których wartość z kolumny pensja jest mniejsza niż 2000, posortowane nierosnąco według parametru staz.					P	F
4.	Wynikiem powyższego zapytania jest zestawienie:					P	F
	nr_p	staz	pensja				
	7782	12	1680				
	7698	15	1530				
	7839	22	1500				

Wypełnia egzaminator	Nr zadania	3.2.	3.3.	3.4.
	Maks. liczba pkt.	1	1	1
	Uzyskana liczba pkt.			

BRUDNOPIS (*nie podlega ocenie*)

