

WYPEŁNIA ZDAJĄCY

KOD

--	--	--

PESEL

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

Egzamin maturalny

Formuła 2023

INFORMATYKA

Poziom rozszerzony


Próbna Matura z Operonem 2024/2025

DATA: **marzec 2025 r.**

CZAS TRWANIA: **210 minut**

LICZBA PUNKTÓW DO UZYSKANIA: **50**

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 15 stron i czy dołączony jest do niego nośnik danych. Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
2. Na pierwszej stronie arkusza wpisz swój numer PESEL i kod.
3. Wpisz zadeklarowane (wybrane) przez ciebie na egzamin system operacyjny, program użytkowy oraz język programowania i środowisko programistyczne.
4. Pisz czytelnie. Używaj długopisu/pióra tylko z czarnym tuszem/atramentem.
5. Nie używaj korektora, a błędne zapisy wyraźnie przekreśl.
6. Pamiętaj, że zapisy w brudnopisie nie będą oceniane.
7. Symbol  zamieszczony w nagłówku zadania oznacza, że zadanie nie wymaga użycia komputera i odpowiedź do niego należy zapisać tylko w miejscu na to przeznaczonym w arkuszu egzaminacyjnym.
8. Pliki oddawane do oceny nazwij dokładnie tak, jak polecono w treści zadań, lub zapisz je pod nazwami (wraz z rozszerzeniem zgodnym z zadeklarowanym oprogramowaniem), jakie podajesz w arkuszu egzaminacyjnym. Pliki o innych nazwach nie będą sprawdzane przez egzaminatora.
9. Nie wpisuj żadnych znaków w części przeznaczonej dla egzaminatora.

WYPEŁNIA ZDAJĄCY

WYBRANE:

.....
(system operacyjny)

.....
(program użytkowy)

.....
(środowisko programistyczne)

Zadanie 1. Poprawne nawiasowanie

Poprawne wyrażenie nawiasowe to takie, w którym każdy nawias otwierający rozpoczyna nawiasowanie, a każdy odpowiadający mu nawias zamykający zamyka nawiasowanie w odpowiedniej kolejności. Wewnątrz nawiasowania mogą znajdować się kolejne, zagnieżdżone nawiasowania. Do zadania używamy czterech rodzajów nawiasów { }, [], (), < >.

Przykład:

Poprawne nawiasowanie

{[((<>)(<>))]}({})

Błędne nawiasowanie

{({})}

W pliku tekstowym `nawiasy1.txt` umieszczono 20 linii, a w pliku `nawiasy2.txt` umieszczono 10 linii z danymi nawiasowania, po jednym zestawie w każdej linii. Pojedynczy zestaw (linia) zawiera od 10 do 1000 nawiasów.

Napisz program (lub programy), który znajdzie odpowiedzi do poniższych zadań. Odpowiedzi zapisz w pliku `wyniki1.txt`, a każdą z nich poprzedź numerem odpowiedniego zadania.

1.1.

0–1
2–3

Zadanie 1.1. (0–3)

W pliku `nawiasy1.txt` wyznacz liczbę nawiasów otwierających każdego rodzaju. Jako rozwiązanie podaj symbol nawiasu i liczbę jego wystąpień.

1.2.

0–1
2–3

Zadanie 1.2. (0–3)

W pliku `nawiasy1.txt` znajdź ciągi, które w całości przedstawiają poprawne nawiasowanie. Jako rozwiązanie podaj numer wiersza, w którym jest umieszczony ciąg. Zakładamy, że wiersze w pliku są numerowane od wartości 1. Dla przykładu pierwszy wiersz zawiera poprawne nawiasowanie.

1.3.

0–1–2
3–4

Zadanie 1.3. (0–4)

Głębokością nawiasowania nazywamy najdłuższą sekwencję otwartych nawiasów. Na przykład: dla ciągu `{[((<{}>)]}` głębokość nawiasowania wynosi 6, dla ciągu `(<{}>){(<()>)}` największa głębokość nawiasowania wynosi 4.

W pliku `nawiasy2.txt` umieszczono tylko poprawne zestawy danych. Dla każdego zestawu danych wyznacz największą głębokość nawiasowania. Jako wynik podaj numer wiersza i liczbę określającą głębokość nawiasowania.

Do oceny oddajesz:

- plik `wyniki1.txt`, zawierający odpowiedzi do zadań 1.1.–1.3.
- plik (lub pliki) zawierający kody źródłowe twojego programu (programów) o nazwie (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania):

Zadanie 1.1.

Zadanie 1.2.

Zadanie 1.3.

0-1
2-3

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który zaszyfruje ciąg znaków algorytmem Janka.

Dane:

P – PESEL – jedenastocyfrowa liczba z zerami wiodącymi

B – ciąg znaków będący zaszyfrowanym tekstem

This image shows a full page of blank graph paper. The grid consists of thin, light gray horizontal and vertical lines that intersect to form a uniform pattern of small squares across the entire surface. There are no margins, text, or other markings on the paper.

Zadanie 3.1. (0–3)

Uwaga: 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. Dopuszczalne jest użycie sprawdzenia długości (wielkości) łańcucha tekstowego oraz rzutowania typów pomiędzy tekstem a liczbami. Zabronione jest używanie wszelkich innych niż wymienione operatorów oraz funkcji wbudowanych.

This image shows a full page of blank graph paper. The grid consists of thin, light gray horizontal and vertical lines that intersect to form a uniform pattern of small squares across the entire surface. There are no margins, text, or other markings present.

3.2.

0-1
2-3

Zadanie 3.2. (0–3)

Przedstawiony poniżej algorytm ma za zadanie wyznaczyć i wypisać wszystkie liczby palindromiczne, które są kwadratami liczb naturalnych z zakresu od 1 do 10000.

Uzupełnij luki w taki sposób, aby algorytm działał poprawnie.

Specyfikacja:

czy_palindrom – funkcja sprawdzająca, czy podana liczba jest palindromem

X – sprawdzana liczba

P – zmienna pomocnicza

W – wynik odwracania liczby

A – liczba podnoszona do kwadratu

```
funkcja czy_palindrom (X)
    P=X
    W=0
    dopóki.....
        W=W*10 +.....
        P=.....
    Jeżeli W==X
        Zwróć prawda
    W przeciwnym wypadku
        Zwróć fałsz

dla A=1,2,3... 10000 wykonaj
    jeżeli czy_palindrom(A*A)
        wypisz A*A
```

3.3.

0-1
2-3

Zadanie 3.3. (0–3)

Uwaga: W tym zadaniu wszelkie dane wejściowe, wyniki końcowe i obliczenia pośrednie należy wykonać w systemie ósemkowym.

Plik `osemkowy.txt` zawiera 100 dwudziestocyfrowych liczb zapisanych w systemie ósemkowym. Napisz program, który dla każdej liczby wyznaczy najmniejszą wartość, jaką należy do tej liczby dodać, aby powstała liczba palindromiczna. Podaj minimalną i maksymalną liczbę wśród wyznaczonych wartości.

Do oceny oddajesz:

- plik `wyniki3.txt`, zawierający odpowiedzi do zadania 3.3.
- plik (lub pliki) zawierający kody źródłowe twojego programu (programów) o nazwie (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania):

.....
.....

Zadanie 4. Regaty

Aby obliczyć prędkość jachtu, należy podzielić odległość, jaką przeplynał jacht, przez czas, którego potrzebował na jej przebycie. Prędkość jest zwykle wyrażana w węzłach (1 węzeł to 1 mila morska na godzinę).

GPS co dwie godziny dokonuje pomiaru przeplniętego dystansu, zaczyna od godziny 2.00, a kończy o godzinie 00.00.

W pliku tekstowym `regaty.txt` znajduje się zapis dzienny regat: data oraz pomiary odległości w milach morskich. Wszystkie dane są rozdzielone znakiem spacji.

Przykład fragmentu pliku:

```
Data m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12
06.05.2024 3 11 11 43 49 38 43 40 46 26 30 20
07.05.2024 2 8 14 43 39 47 43 38 48 27 27 20
08.05.2024 10 3 8 45 49 45 41 45 43 28 21 20
09.05.2024 8 9 14 44 39 38 42 40 46 29 33 18
10.05.2024 3 12 10 39 39 43 46 39 41 18 27 14
11.05.2024 1 2 8 39 43 49 41 38 45 18 31 19
```

Wykorzystaj dane zawarte w pliku i dostępne narzędzia informatyczne oraz wykonaj zadania. Odpowiedzi zapisz w kolejnych wierszach pliku tekstowego `wyniki4.txt`. Odpowiedź do każdego zadania poprzedź numerem tego zadania.

Zadanie 4.1. (0–2)

Wyznacz średnią prędkość, z jaką płynął jacht w trakcie regat. Prędkość podaj w kilometrach na godzinę, z dokładnością do dwóch miejsc po przecinku (1 węzeł = 1,852 km/h).

4.1.

0–1–2

Zadanie 4.2. (0–2)

Rejs rozpoczyna się w poniedziałek, a kończy w niedzielę.

Wyznacz średnie prędkości (w węzłach) w kolejnych tygodniach rejsu. Na podstawie wykonanego zestawienia utwórz wykres kolumnowy ilustrujący średnie prędkości jachtu w tygodniach. Pamiętaj o czytelnym opisie wykresu.

4.2.

0–1–2

Zadanie 4.3. (0–2)

Wykonaj zestawienie średniej przeplniętej odległości w każdym dniu tygodnia (osobno dla poniedziałków, wtorków itd.), w kolejnych przedziałach pomiarowych (od m1 do m12).

4.3.

0–1–2

Zadanie 4.4. (0–2)

Jeżeli średnia prędkość na danym odcinku jest większa od średniej prędkości na poprzednim odcinku lub jej równa, mówimy o wzroście prędkości (jacht nie zwalnia). Jeżeli jest mniejsza – mówimy o jej spadku (zwalnianiu).

Wyznacz, ile razy jacht przyspieszał i ile razy zwalniał.

Pierwszy odcinek pomiarowy każdego dnia jest odniesieniem dla kolejnej wartości (nie podajemy analizie okresu pomiędzy m12 a m1 dnia kolejnego), dlatego przyjmujemy go jako wartość neutralną i nie jest on ani przyspieszeniem, ani zwalnianiem.

4.4.

0–1–2

4.5.

0–1–2

Zadanie 4.5. (0–2)

Dobrym dniem regatowym jest dzień, w którym jacht nie zwolnił w czasie sześciu lub więcej kolejnych, następujących bezpośrednio po sobie odcinków pomiarowych, czyli jego średnia prędkość na danym odcinku była taka sama lub większa w stosunku do odcinka poprzedniego.

Wyznacz daty, w których wystąpił dobry dzień regatowy.

Do oceny oddajesz:

- plik `wyniki4.txt`, zawierający odpowiedzi do zadań 4.1.–4.5.
- plik z wykresem do zadania 4.2. o nazwie
- pliki z komputerową realizacją twoich rozwiązań o nazwach (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązywania zadania):

.....
.....

Zadanie 5. Czysze

Wspólnota mieszkaniowa Skowronek planuje przeprowadzić na koniec roku weryfikację stanu opłat. Przekazała do analizy cztery pliki, które zawierają wyciąg informacji z bazy danych wspólnoty.

Plik `lokale.txt` zawiera informacje o lokalach i mieszkańcach. W każdym wierszu zapisano: `nr_lokalu` – wartość tekstowa składająca się z dwóch cyfr oznaczających numer bloku, znaku „/” i trzycyfrowego numeru mieszkania, np.: 03/024

`l_mieszkancow` – liczba osób mieszkających w lokalu, liczba całkowita

`id_wlasciciela` – identyfikator właściciela, krótki tekst zbudowany z litery i cyfr

Lokal ma zawsze tylko jednego właściciela, ale właściciel może mieć kilka lokali.

Przykład:

<code>nr_lokalu</code>	<code>l_mieszkancow</code>	<code>id_wlasciciela</code>
01/001	4	11
01/002	6	12
01/003	3	13
01/004	2	14
01/005	2	15

Plik `wlasciciele.txt` zawiera informacje o właścicielach. W każdym wierszu zapisano:

`id_wlasciciela` – identyfikator właściciela, krótki tekst zbudowany z litery i cyfr

`imie` – imię właściciela, ciąg znaków w kodowaniu UTF-8

`nazwisko` – nazwisko właściciela, ciąg znaków w kodowaniu UTF-8

Przykład:

<code>id_wlasciciela</code>	<code>imie</code>	<code>nazwisko</code>
11	Gracjan	Kołodziej
12	Przemysław	Głowacka
13	Mateusz	Jasiński
14	Bogumił	Szewczyk
15	Miłosz	Kaczmarczyk

Plik `czynsz.txt` zawiera informacje o wysokościach składowych czynszu w każdym miesiącu, dla każdego lokalu. W każdym wierszu zapisano:

`nr_lokalu` – wartość tekstowa składająca się z dwóch cyfr oznaczających numer bloku, znaku „/” i trzycyfrowego numeru mieszkania, np.: 03/024 (uwaga: to nie jest ani liczba, ani data)

`miesiac` – liczba określająca kolejny miesiąc roku, liczba całkowita

`czynsz` – podstawowa wysokość czynszu, liczba rzeczywista

`smieci` – opłata za wywóz śmieci, liczba rzeczywista

`gaz` – opłata za gaz, liczba rzeczywista

`woda` – opłata za wodę, liczba rzeczywista

Przykład:

<code>nr_lokalu</code>	<code>miesiac</code>	<code>czynsz</code>	<code>smieci</code>	<code>gaz</code>	<code>woda</code>
01/001	1	725,66	60	8,5	110,5
01/001	2	725,66	60	8,5	110,5
01/001	3	725,66	60	8,5	115,5
01/001	4	725,66	80	8,5	115,5
01/001	5	748,54	80	8,5	123,3

Plik `wplaty.txt` zawiera informacje o kwotach wpłat dokonywanych przez mieszkańców. W każdym wierszu zapisano:

`nr_lokalu` – wartość tekstowa składająca się z dwóch cyfr oznaczających numer bloku, znaku „/” i trzycyfrowego numeru mieszkania, np.: 03/024 (uwaga: to nie jest ani liczba, ani data)

`miesiac` – liczba określająca kolejny miesiąc roku, liczba całkowita

`kwota` – podstawowa wysokość czynszu, liczba rzeczywista

Przykład:

<code>nr_lokalu</code>	<code>miesiac</code>	<code>kwota</code>
01/001	1	904,66
01/001	2	904,66
01/001	3	909,66
01/001	4	929,66
01/001	5	960,34

We wszystkich plikach dane w wierszach są rozdzielone znakami tabulacji, a pierwszy wiersz w każdym pliku jest wierszem nagłówkowym.

Z wykorzystaniem danych zawartych w plikach i dostępnych narzędzi informatycznych wykonaj zadania. Odpowiedzi zapisz w kolejnych wierszach pliku tekstowego `wyniki5.txt`. Odpowiedź do każdego zadania poprzedź numerem tego zadania.

5.1.

0–1–2

Zadanie 5.1. (0–2)

Na opłaty za zajmowany lokal składają się wszystkie składniki podane w bazie.

Podaj imię i nazwisko właściciela, który ma największe zaległości w opłatach za cały rok, oraz kwotę długu. Pamiętaj, że można być właścicielem kilku lokali.

5.2.

0–1–2

Zadanie 5.2. (0–2)

Niezależnie od tego, czy opłaty na koniec roku zostały wyrównane, przewodniczący wspólnoty chce wiedzieć, czy w którymś z miesięcy nie uiszczono opłaty za lokal.

Podaj numery lokali w każdym z miesięcy, w którym nie była uiszczona opłata za lokal.

5.3.

0–1
2–3

Zadanie 5.3. (0–3)

Wiedząc, że 3% kwoty należnego czynszu to podatek ekologiczny, wyznacz numer lokalu, w którym w trakcie roku naliczono największy podatek w przeliczeniu na jednego mieszkańca. Podaj numer lokalu oraz kwotę podatku przeliczoną na jednego mieszkańca.

Do oceny oddajesz:

- plik `wyniki5.txt`, zawierający odpowiedzi do zadań 5.1.–5.3.
- plik (lub pliki) z komputerową realizacją twoich rozwiązań o nazwie (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania):

.....
.....

Informacja do zadań 5.4. i 5.5.

Założmy, że do opisanych wcześniej tabel bazy danych dodano jeszcze jedną – `garaze`, w której zapisano dokładne informacje o posiadanych garażach.

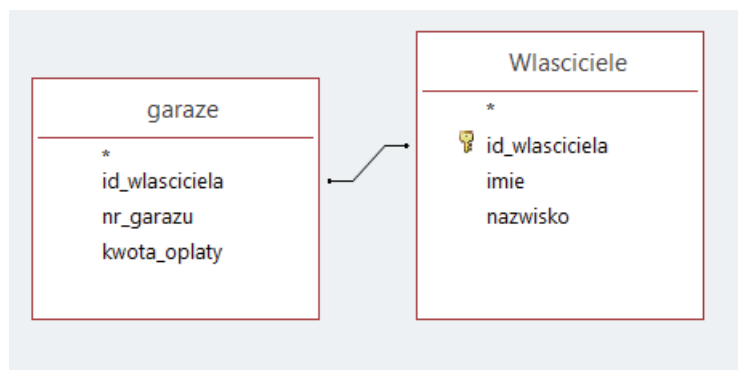
Tabela `garaze` składa się z pól:

`id_wlasciciela` – identyfikator właściciela, krótki tekst

`nr_garazu` – numer garażu, liczba całkowita

`kwota_oplaty` – roczna opłata za garaż, liczba rzeczywista

Relacje w powstałej bazie będą wyglądały, jak na poniższej ilustracji.



Zadanie 5.4. (0–1)

Napisz zapytanie SQL, którego wynikiem będzie zestawienie, w którym podasz imię i nazwisko właściciela mającego największą liczbę garaży.

Miejsce na rozwiązanie:

This image shows a full page of blank graph paper. The grid consists of thin, light gray horizontal and vertical lines that intersect to form a uniform pattern of small squares across the entire surface. There are no margins, text, or other markings on the paper.

Zadanie 5.5. (0–2)

Napisz zapytanie SQL, którego wynikiem będą imiona i nazwiska właścicieli mających dokładnie dwa garaże, a dane te będą posortowane według opłaty za użytkowanie garaży (od najniższej do najwyższej).

Miejsce na rozwiązanie:

[illegible]

1.	Brama sieciowa to maszyna podłączona do sieci komputerowej, za pośrednictwem której komputery z sieci lokalnej komunikują się z komputerami w innych sieciach.	P	F
2.	Model TCP/IP jest zbudowany na czterech warstwach: aplikacji, transportowej, internetu, fizycznej.	P	F
3.	P2P to mała sieć, w której komputery łączą się poprzez serwer lokalny, skonfigurowany przez lokalnego administratora.	P	F

Działanie na liczbach	Wynik w zapisie czwórkowym	Wynik w zapisie szesnastkowym
$D2C,4A1_{(16)} + 412,671_{(8)}$		
$317,26_{(10)} - 2121,1131_{(4)}$		

[illegible]

Kolor obiektu został wyrażony jako #BA55D3.

1.	Zapis dziesiętny składowej niebieskiej to 211.	P	F
2.	Zapis dziesiętny składowej zielonej to 87.	P	F
3.	Zapis dziesiętny koloru to 186,85,211.	P	F

W jakim czasie prześlemy dane, które zajmują na dysku 1,8 GB pojemności, a dysponujemy przepustowością 18Mb/s? Odpowiedź podaj w minutach.

Miejsce na rozwiązanie:

A full-page sheet of white graph paper featuring a uniform grid of thin, light gray horizontal and vertical lines. The grid consists of small squares covering the entire area of the page.

10.

0-1

Uzupełnij zdanie tak, aby było prawdziwe.

Strona 14

BRUDNOPIS (*nie podlega ocenie*)

ISBN 978-83-8197-532-2



9 788381 975322