

WYPEŁNIA ZDAJĄCY

KOD

--	--	--

PESEL

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

Miejsce na naklejkę.

Sprawdź, czy kod na naklejce to
M-100.

Jeżeli tak – przyklej naklejkę.
Jeżeli nie – zgłoś to nauczycielowi.

**EGZAMIN MATURALNY
INFORMATYKA
– POZIOM ROZSZERZONY**

TERMIN: **12 kwietnia 2023 r.**

CZAS PRACY: **210 minut**

LICZBA PUNKTÓW DO UZYSKANIA: **50**

WYPEŁNIA ZDAJĄCY


WYBRANE:

.....
(system operacyjny)

.....
(program użytkowy)

.....
(język programowania i środowisko programistyczne)

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 17 stron i czy dołączony jest do niego nośnik danych – podpisany DANE_PR. Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
2. Wpisz zadeklarowane (wybrane) przez Ciebie na egzamin system operacyjny, program użytkowy oraz język programowania i środowisko programistyczne.
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 będą oceniane.
6. 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.
7. 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.**
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.

MINP-R0-**100**-1204

Zadanie 1. Statki jednoosobowe

Statki jednoosobowe są odmianą gry w statki przeznaczoną dla jednego gracza. W grze tej dana jest tablica o rozmiarze 10x10 reprezentująca ułożenie statków:

	1	2	3	4	5	6	7	8	9	10
1	X	X						X		
2								X		
3		X	X	X	X			X		
4								X		
5				X				X		
6				X				X		X
7				X						X
8										X
9						X	X			X
10		X	X	X						

Każdy statek jest reprezentowany przez ciąg znaków 'X' ustawionych w tym samym wierszu albo w tej samej kolumnie. Pomiędzy znakami reprezentującymi dany statek nie może być pustych pól. Statki nie mogą stykać się bokami ani rogami. Na planszy zawsze znajdują się tylko statki o następujących wielkościach:

- 2 statki o długości 2 pól
- 2 statki o długości 3 pól
- 2 statki o długości 4 pól
- 1 statek o długości 6 pól

Gracz na początku rozgrywki ma przed sobą pustą planszę 10x10. W kolejnych krokach podaje współrzędne pola tabeli, w którym przewiduje, że znajduje się fragment pewnego statku. Współrzędne są podawane w postaci pary (x, y), gdzie x oznacza współrzędną poziomą, zaś y współrzędną pionową.

Przykład

Na powyższym rysunku statek o długości 6 zajmuje pola o współrzędnych: (8, 1), (8, 2), (8, 3), (8, 4), (8, 5), (8, 6).

Jeżeli współrzędne podane w danym kroku reprezentują pole, na którym znajduje się fragment statku, gracz otrzymuje odpowiednią informację zwrotną. Jeżeli w danym kroku zostało wybrane ostatnie nieodkryte pole danego statku to taka informacja również trafia do gracza. Celem gracza jest odkrycie wszystkich statków w jak najmniejszej liczbie kroków.

W pliku `statki.txt` znajduje się reprezentacja planszy z rozkładem statków zgodnym z warunkami gry (nie jest to układ przedstawiony na rysunku we wstępie do zadania). Fragment statku jest reprezentowany przez znak 'X' zaś puste pole przez znak '.'.

Ponadto w pliku `statki_ruchy.txt` znajduje się ciąg 40 par współrzędnych (x, y) podawanych przez gracza w kolejnych krokach. Zakładamy, że gracz nie wskazał dwa razy tego samego pola. Dane w każdym wierszu są oddzielone pojedynczym znakiem odstępu. Ostatni krok odkrywa ostatni nieodkryty do tej pory fragment statku. Zatem po wykonaniu tego kroku wszystkie statki znajdujące się na planszy są w pełni odkryte.

Napisz **program(-y)** który(-e) znajdzie(-dą) odpowiedzi do poniższych zadań. Odpowiedzi zapisz w pliku `wyniki1.txt`, a każdą z nich poprzedź numerem odpowiedniego zadania.

Zadanie 1.1. (0-2)

Ustal współrzędne pól, na których znajduje się statek o długości 6. Zapisz każdą parę w osobnym wierszu pliku `wyniki1.txt`.

Dla planszy opisanej we wstępie do zadania odpowiedzią jest

8 1

8 2

8 3

8 4

8 5

8 6

Uwaga

Można założyć, że w pliku `statki.txt` statek o długości 6 znajduje się w pewnym wierszu.

Zadanie 1.2. (0-3)

Dla każdego statku z planszy zapisanej w pliku `statki.txt` podaj współrzędne pola początkowego i współrzędne pola końcowego. Zakładamy, że w przypadku statków położonych na planszy pionowo pole początkowe ma niższą wartość współrzędnej pionowej niż pole końcowe. Analogicznie w przypadku statków położonych na planszy poziomo pole początkowe ma niższą wartość współrzędnej poziomej niż pole końcowe.

Zapisz znalezione pary współrzędnych w kolejnych wierszach pliku `wyniki.txt`, podając najpierw współrzędne pola początkowego (oddzielone pojedynczym znakiem odstępu), a następnie, po pojedynczym znaku odstępu współrzędne pola końcowego (również oddzielone pojedynczym znakiem odstępu).

Uwaga

Kolejność opisywanych statków nie ma znaczenia

Dla planszy opisanej we wstępie do zadania odpowiedzią jest

1 1 2 1

8 1 8 6

2 3 5 3

4 5 4 7

10 6 10 9

6 9 7 9

2 10 4 10

Zadanie 1.3. (0-3)

Ustal długość najdłuższego ciągu kolejnych ruchów gracza, w którym każdy ruch skutkuje odkryciem pola pewnego statku. Zapisz znalezioną długość w pliku wyniki1.txt.

Do oceny oddajesz:

- plik wyniki1.txt, zawierający odpowiedzi do zadań 1.1.–1.3.
- plik(-i) zawierający(-e) kody źródłowe Twojego(-ich) programu(-ów) o nazwie(nazwach): (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania)

.....

.....

Zadanie 2. Analiza algorytmu

Dany jest algorytm w postaci pseudokodu.

Specyfikacja:

Dane wejściowe:

n – liczba całkowita dodatnia

k – liczba całkowita dodatnia nie większa niż n

Dane wyjściowe:

wartość zmiennej *ile*, będąca efektem wywołania *Podział*(n , k)

Algorytm:

Podział(x , y)

jeśli $y > 1$ oraz $x > 0$

dla $i \leftarrow 1, 2, \dots, x - y + 1$

Podział($x - i$, $y - 1$)

w przeciwnym razie

$ile \leftarrow ile + 1$

$ile \leftarrow 0$

Podział(n , k)

Uwaga

Wartość zmiennej *ile* jest zapamiętywana pomiędzy poszczególnymi wywołaniami rekurencyjnymi funkcji *Podział*.

Zadanie 2.1. (0-2)

Podaj wartości zmiennej *ile* będące efektem wywołania funkcji *Podział* z następującymi parametrami:

n	k	<i>ile</i>
10	2	
6	3	
5	4	

Uwaga

Przed każdym spośród podanych wywołań ustawiamy wartość zmiennej *ile* na 0.

Zadanie 2.2. (0-1)

Wyjaśnij jaka jest interpretacja wartości zmiennej *ile* będącej efektem wywołania $\text{Podział}(n, k)$.

Zadanie 2.3. (0-1)

Podaj przykład parametrów wejściowych n i k , dla których wartość zmiennej *ile* będącej efektem wywołania *Podział*(n , k) będzie równa 165.

wartość n :

wartość k :

Zadanie 2.4. (0-3)

Zapisz w wybranej przez siebie notacji (w postaci pseudokodu lub w wybranym języku programowania) nierekurencyjny algorytm, który oblicza wartość zmiennej *ile*, odpowiadającą wywołaniu *Podział(n, 3)*.

Uwaga

W zapisie algorytmu możesz wykorzystać tylko operacje arytmetyczne: dodawanie, odejmowanie, mnożenie, dzielenie, dzielenie całkowite, resztę z dzielenia, oraz porównywanie liczb; instrukcje sterujące i przypisania do zmiennych lub samodzielnie napisane funkcje zawierające wyżej wymienione operacje. W szczególności zabronione jest wywoływanie funkcji *Podział*.

Specyfikacja:

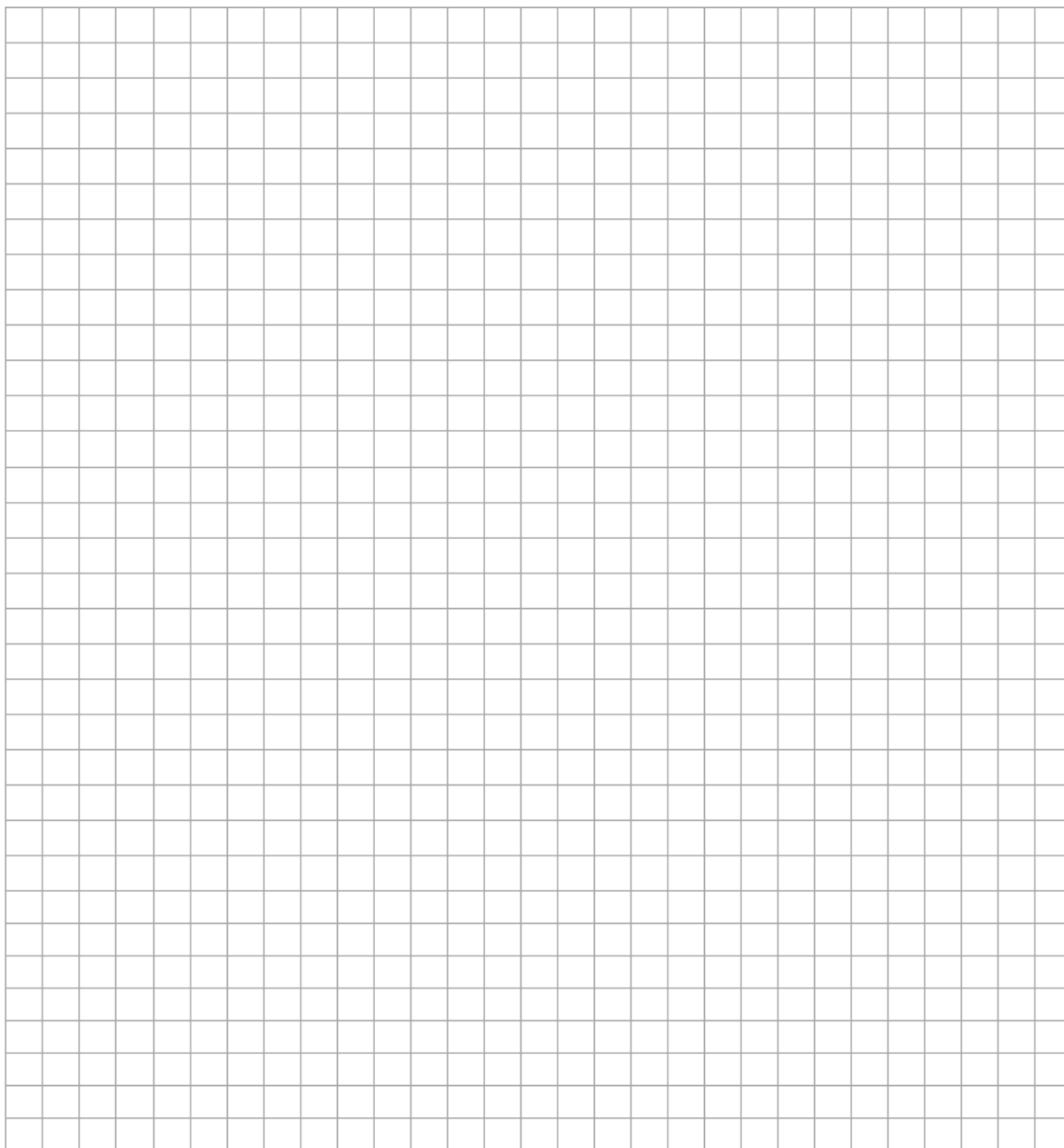
Dane wejściowe:

n – liczba całkowita dodatnia

Dane wyjściowe:

wartość zmiennej *ile*, odpowiadająca wywołaniu *Podział(n, 3)*

Algorytm:



Zadanie 3. Anagramy

Zadanie 3.1. (0-2)

Uzupełnij luki oznaczone poziomymi kreskami w poniższym algorytmie sprawdzającym, czy dwa słowa są anagramami. Algorytm został zapisany w postaci funkcji.

Specyfikacja:

Dane wejściowe:

$s1$, $s2$ – słowa, dla których sprawdzamy, czy są anagramami; tablice znaków (duże litery alfabetu angielskiego)

$n1$, $n2$ – liczby znaków w słowach, kolejno $s1$ i $s2$; liczby całkowite dodatnie

Dane wyjściowe:

informacja, czy $s1$ i $s2$ są anagramami (PRAWDA – jeśli są lub FAŁSZ – jeśli nie są)

Algorytm:

funkcja *czy_anagramy*($s1$, $s2$, $n1$, $n2$)

jeśli $n1 \neq n2$

zwróć _____

w przeciwnym razie

$T1[1..26] \leftarrow [0, 0, \dots, 0]$

$T2[1..26] \leftarrow [0, 0, \dots, 0]$

dla $i \leftarrow 1, 2, \dots, \underline{\hspace{2cm}}$

$T1[w(s1[i])] \leftarrow \underline{\hspace{2cm}}$

$T2[w(s2[i])] \leftarrow \underline{\hspace{2cm}}$

dla $i \leftarrow 1, 2, \dots, 26$

jeśli $T1[i] \neq T2[i]$

zwróć _____

zwróć PRAWDA

Uwaga

- Funkcja *w* konwertuje znak na jego reprezentację liczbową – dla znaku 'A' zwraca 1, dla znaku 'B' zwraca 2 itd. W przypadku znaku 'Z' funkcja *w* zwraca 26.
- Zapis $X[1..26] \leftarrow [0, 0, \dots, 0]$ oznacza, że tablicę *X* wypełniamy 26 zerami

liczba anagramów

10

Napisz **program**, który ustali w ilu wierszach pliku `pary_slow.txt` słowo pierwsze jest anagramem słowa drugiego. Jako odpowiedź podaj liczbę tych wierszy. Wynik zapisz w pliku `wyniki3_3.txt`.

10

Do oceny oddajesz:

- Odpowiedzi do zadań 3.1. i 3.2. zapisane w odpowiednich miejscach
- Pliki: wyniki3_3.txt i wyniki3_4.txt zawierające odpowiedzi do zadań 3.3. i 3.4.
- plik(-i) zawierający(-e) kody źródłowe Twojego(-ich) programu(-ów) o nazwie(nazwach):
(uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania)

.....

Zadanie 4. Pociąg relacji Przemyśl Główny – Świnoujście Port

W pliku rozklad_jazdy.txt zawarto informację o trasie pokonywanej przez pociąg relacji Przemyśl Główny – Świnoujście Port. W kolejnych wierszach pliku znajdują się następujące informacje: nazwa stacji (stacja), godzina przyjazdu pociągu (przyjazd), godzina odjazdu pociągu (odjazd) oraz odległość przebyta przez pociąg do tej pory (przebyta odleglosc w km). W przypadku stacji Przemyśl Główny (PRZEMYSŁ GLOWNY) przebyta odległość jest równa 0, ponieważ na tej stacji pociąg rozpoczyna swój bieg. Dane w każdym wierszu są oddzielone pojedynczym znakiem tabulacji.

Wykorzystując dane z pliku rozklad_jazdy.txt oraz dostępne narzędzia informatyczne podaj odpowiedzi do poniższych zadań. Uzyskane wyniki zapisz w pliku wyniki4.txt.

Zadanie 4.1. (0-2)

Podaj, między którymi następującymi po sobie stacjami pociąg rozwijał największą średnią prędkość.

Uwaga

Obliczoną prędkość wyrażamy w kilometrach na minutę i podajemy z dokładnością do dwóch miejsc po przecinku.

Przykład

Dla danych:

stacja	przyjazd	odjazd	przebyta odleglosc w km
PRZEWORSK	10:55:00	10:56:00	49,9
LANCUT	11:09:00	11:10:00	70

średnia prędkość na trasie między stacjami Przetworsk (PRZETWORSK) a Łańcut (LANCUT) jest obliczana w następujący sposób:

$$\text{długość trasy} = 70 - 49,9 = 20,1 \text{ km}$$

$$\text{czas jazdy} = 13 \text{ min}$$

$$\text{średnia prędkość} = \frac{20,1}{13} = 1,55 \frac{\text{km}}{\text{min}}$$

Zadanie 4.2. (0-3)

Dla każdego czasu postoju krótszego niż 10 minut podaj liczbę stacji, na których postój trwał tyle minut. Uzyskane dane przedstaw na wykresie kolumnowym. Zadbaj o czytelność wykresu.

Uwaga

- Za czas postoju na stacji przyjmujemy różnicę między czasem odjazdu a czasem przyjazdu na tę stację.
- Pomijamy postój na stacjach Przemyśl Główny (PRZEMYSŁ GLOWNY) i Świnoujście Port (SWINOUJSCIE PORT)

Przykład

Jeżeli pociąg przyjechał o godzinie 12:34:00 i odjechał o godzinie 12:37:00 to czas postoju jest równy 3 min. W szczególności gdy pociąg przyjechał i odjechał o tej samej godzinie, to za czas postoju przyjmujemy 0 min.

Zadanie 4.3. (0-2)

Podaj długość najdłuższego ciągu stacji, w którym dla każdych trzech sąsiednich stacji A, B, C odległość między stacjami B i A jest mniejsza od odległości między stacjami C i B.

Przykład

Dla danych:

stacja	przyjazd	odjazd	przebyta odleglosc w km
SEDZISZOW MALOPOLSKI	11:38:00	11:39:00	112,5
ROPCZYCE	11:44:00	11:45:00	120,6
DEBICA	11:53:00	11:54:00	133,7
TARNOW	12:10:00	12:12:00	166,9
BRZESKO OKOCIM	12:25:00	12:26:00	193,4

ciąg o własności opisanej powyżej składa się z następujących stacji:

SEDZISZOW MALOPOLSKI, ROPCZYCE, DEBICA, TARNOW i ma długość 4.

Informacja do zadań 4.4. i 4.5.

W rzeczywistości na punktualność pociągu wpływa wiele czynników. W efekcie pociąg może przybyć na stację (jak również odjechać z niej) o innej godzinie niż zostało to podane w rozkładzie jazdy.

Dokonaj symulacji rzeczywistego przebiegu trasy pociągu przyjmując następujące założenia:

- Pociąg odjechał punktualnie ze stacji Przemysł Główny (PRZEMYSŁ GLOWNY)
- Na każdy jeden pełen kilometr trasy między dwoma sąsiednimi stacjami przypada 0,5 min opóźnienia

Przykład

Dla danych:

stacja	przyjazd	odjazd	przebyta odleglosc w km
RADYMNO	10:35:00	10:36:00	21,2
JAROSLAW	10:45:00	10:46:00	35
PRZEWORSK	10:55:00	10:56:00	49,9

między stacjami Radymno i Jarosław odległość jest równa $35 - 21,2 = 13,8$ km. Zatem opóźnienie między tymi stacjami wzrośnie o $13 \cdot 0,5 = 6,5$ min. Oznacza to, że pokonanie odcinka z Radymna do Jarosława zajmie o 6,5 min dłużej niż wynikałoby to z rozkładu jazdy.

- Na stacjach, na których postój trwa ponad 2 min opóźnienie wzrasta o 1 min na każde pełne 50 km przejechanych do tej pory.

Przykład

Dla danych:

stacja	przyjazd	odjazd	przebyta odleglosc w km
SZCZECIN DABIE	21:44:00	21:50:00	864

na stacji Szczecin Dąbie (SZCZECIN DABIE) pociąg miał zaplanowany postój trwający 6 min. Ponieważ liczba pełnych 50 km zawierających się w przebytej do tej pory odległości jest równa $864 \div 50 = 17,28$, to opóźnienie pociągu na tej stacji wzrośnie o $17 \cdot 1 = 17$ min. Oznacza to, że postój na tej stacji będzie trwał o 17 min dłużej niż zaplanowano w rozkładzie.

Uwaga

Pociąg na stację Gogolin (GOGOLIN) przyjechał z 209 min opóźnieniem. Wykorzystaj tę informację aby sprawdzić poprawność swoich obliczeń.

Zadanie 4.4. (0-3)

- Podaj, o której godzinie (z dokładnością do 0,5 min) pociąg dotrze do stacji Świnoujście Port?
- Na którą stację, jako pierwszą, pociąg przyjechał z opóźnieniem przekraczającym 180 min?

Zadanie 4.5. (0-2)

Jak długo jechałby pociąg ze stacji Przemyśl Główny (PRZEMYSŁ GLOWNY) do stacji Świnoujście Port (SWINOUJSCIE PORT) jeżeli postój na każdej stacji (poza pierwszą i ostatnią) trwałby dokładnie 1 min? Wynik podaj z dokładnością do 0,5 min.

Do oceny oddajesz:

- Plik tekstowy wyniki4.txt zawierający odpowiedzi do poszczególnych zadań (odpowiedź do każdego zadania powinna być poprzedzona jego numerem).
- Plik zawierający wykres do zadania 4.2. o nazwie

.....

- Plik (pliki) zawierający (zawierające) komputerową realizację twoich obliczeń o nazwie(nazwach):

.....

Zadanie 5. Dzień otwarty

14.04.2023 w pewnej szkole średniej odbył się dzień otwarty. W trakcie tego wydarzenia uczniowie prezentowali przybyłym gościom, czym zajmują się na lekcjach oraz w jaki sposób rozwijają swoje pasje po zajęciach. Prezentacje miały formę pokazów o różnych czasach trwania. Każdy pokaz był związany z określonym profilem kształcenia (od A do G) lub dotyczył informacji ogólnych o szkole. W plikach `sale.txt`, `uczniowie.txt`, `pokazy.txt` oraz `uczniowie_pokazy.txt` zebrano informacje na temat tego wydarzenia.

W pliku `sale.txt` znajduje się informacja o salach, w których odbywały się pokazy: identyfikator sali (`id_sali`), profil, którego dotyczą pokazy w danej sali (`profil`).

Przykład

<code>id_sali</code>	<code>profil</code>
1	ogolny
2	ogolny

W pliku `uczniowie.txt` zawarto informacje na temat uczniów przygotowujących pokazy: identyfikator ucznia (`id_ucznia`), imię ucznia (`imie`) i nazwisko ucznia (`nazwisko`).

Przykład

<code>id_ucznia</code>	<code>imie</code>	<code>nazwisko</code>
1	Leonardo	Kwiatkowski
2	Kajetan	Urbanski

W pliku `pokazy.txt` znajdują się informacje o pokazach prezentowanych przez uczniów: identyfikator pokazu (`id_pokazu`), temat pokazu (`temat_pokazu`), identyfikator sali, w której odbywa się pokaz (`id_sali`), godzina rozpoczęcia pokazu (`godz_rozp`) oraz godzina zakończenia pokazu (`godz_zak`).

Przykład

<code>id_pokazu</code>	<code>temat_pokazu</code>	<code>id_sali</code>	<code>godz_rozp</code>	<code>godz_zak</code>
1	w zdrowym ciele zdrowy duch	1	10:00	12:00
2	turniej w piłkarzyki	2	09:00	13:00

W pliku `uczniowie_pokazy.txt` zebrano informacje na temat przyporządkowania uczniów szkoły do poszczególnych pokazów: identyfikator pokazu (`id_pokazu`) oraz identyfikator ucznia, który ten pokaz przygotowuje (`id_ucznia`).

Przykład

<code>id_pokazu</code>	<code>id_ucznia</code>
39	125
3	117

Wykorzystując dane zapisane w plikach `sale.txt`, `uczniowie.txt`, `pokazy.txt`, `uczniowie_pokazy.txt` oraz dostępne narzędzia informatyczne wykonaj zadania 5.1. – 5.3. Odpowiedzi do zadań zapisz w pliku `wyniki5.txt` poprzedzając każdą z nich numerem odpowiedniego zadania.

Zadanie 5.1. (0-2)

Podaj identyfikator sali (`id_sali`), w której odbyła się największa liczba pokazów.

Zadanie 5.2. (0-2)

Podaj identyfikatory sal (`id_sali`), w których w godzinach 11:00 – 12:00 nie odbywał się żaden pokaz.

Zadanie 5.3. (0-3)

Podaj identyfikator pokazu (`id_pokazu`) oraz jego temat (`temat_pokazu`), który był obsługiwany przez największą liczbę uczniów. Podaj również imiona (`imie`) i nazwiska (`nazwisko`) uczniów, którzy obsługiwali ten pokaz.

Do oceny oddajesz:

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

.....

.....

Informacja do zadań 5.4. i 5.5.

Po zakończeniu dnia otwartego uczniowie przygotowali statystykę dotyczącą liczby gości uczestniczących w poszczególnych pokazach. Dane zawarli w tabeli *pokazy goście*. Tabela *pokazy goście* zawiera następujące pola:

id pokazu (identyfikator pokazu)

I gości (liczba gości uczestniczących w pokazie)

Zadanie 5.4. (0-2)

Napisz zapytanie w języku SQL, za pomocą którego podasz, który pokaz cieszył się najmniejszym zainteresowaniem, tzn. było na nim obecnych najmniej gości.

[illegible]

Zadanie 5.5. (0-2)

Napisz zapytanie w języku SQL, za pomocą którego podasz dla każdego profilu kształcenia (od A do G) liczbę osób obecnych na pokazach związanych z tym profilem.

Uwaga

Pomijamy profil ogólny (ogólny)

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 6. (0-1) Wyszukiwanie elementów w zbiorze przez połowienie

Oceń, czy poniższe zdania są prawdziwe. Zaznacz P, jeśli zdanie jest prawdziwe, lub F – jeśli jest fałszywe.

Wyszukiwanie elementów w zbiorze przez połowienie

1.	wymaga wstępnego posortowania przeszukiwanego zbioru.	P	F
2.	ma złożoność liniową.	P	F
3.	ma złożoność logarytmiczną.	P	F
4.	jest przykładem algorytmu typu dziel i rządź.	P	F

BRUDNOPIS (*nie podlega ocenie*)