CENTRALNA KOMISJA EGZAMINACYJNA

Arkusz zawiera informacje prawnie chronione do momentu rozpoczęcia egzaminu.

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 ***Formuła 2023***

INFORMATYKA

Poziom rozszerzony

WYPEŁNIA ZDAJĄCY

WYBRANE:

..... (system operacyjny)

..... (program użytkowy)

..... (środowisko programistyczne)

Symbol arkusza
MINP-R0-**100**-2606

DATA: **10 czerwca 2026 r.**

GODZINA ROZPOCZĘCIA: **9:00**

CZAS TRWANIA: **210 minut**

LICZBA PUNKTÓW DO UZYSKANIA: **50**

Przed rozpoczęciem pracy z arkuszem egzaminacyjnym

1. Sprawdź, czy nauczyciel przekazał Ci **właściwy arkusz egzaminacyjny**, tj. arkusz we **właściwej formule**, z **właściwego przedmiotu** na **właściwym poziomie**.
2. Jeżeli przekazano Ci **niewłaściwy** arkusz – natychmiast zgłoś to nauczycielowi. Nie rozrywaj banderol.
3. Jeżeli przekazano Ci **właściwy** arkusz – rozerwij banderole po otrzymaniu takiego polecenia od nauczyciela. Zapoznaj się z instrukcją na stronie 2.

Układ graficzny
© CKE 2022

Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 20 stron (zadania 1–8) i czy dołączony jest do niego nośnik danych – podpisany DANE. Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
2. Na pierwszej stronie arkusza oraz na karcie odpowiedzi wpisz swój numer PESEL i przyklej naklejkę z kodem.
3. Wpisz zadeklarowane (wybrane) przez Ciebie na egzamin: system operacyjny, program użytkowy oraz środowisko programistyczne.
4. Symbol  zamieszczony w nagłówku zadania oznacza, że zadanie nie wymaga użycia komputera i odpowiedź do zadania należy zapisać tylko w miejscu na to przeznaczonym w arkuszu egzaminacyjnym.
5. Jeśli rozwiązaniem zadania lub jego części jest program komputerowy, to zapisz go w zadeklarowanym (wybranym) języku programowania i umieść w katalogu (folderze) oznaczonym Twoim numerem PESEL wszystkie utworzone przez siebie pliki w wersji źródłowej.
6. Jeżeli rozwiązaniem zadania lub jego części jest algorytm (który trzeba zapisać w arkuszu) i wybrałeś(-łaś) jego zapis w postaci języka programowania, to użyj języka programowania, który wybrałeś(-łaś) na egzamin (Java, C++ lub Python).
7. Jeśli rozwiązaniem zadania lub jego części jest baza danych utworzona z wykorzystaniem MySQL lub MariaDB, to umieść w katalogu (folderze) oznaczonym Twoim numerem PESEL treści zapytań w języku SQL oraz (przed zakończeniem egzaminu) wyeksportowaną całą bazę danych w formacie *.sql.
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**.
   Pamiętaj, że zadania praktyczne niezawierające komputerowej realizacji rozwiązań zostaną ocenione na 0 punktów.
9. **Przed upływem czasu przeznaczonego na egzamin** zapisz w katalogu (folderze) oznaczonym Twoim numerem PESEL ostateczną wersję plików stanowiących rozwiązania zadań.
10. Pisz czytelnie. Używaj długopisu/pióra tylko z czarnym tuszem/atramentem.
11. Nie używaj korektora, a błędne zapisy wyraźnie przekreśl.
12. Nie wpisuj żadnych znaków w tabelkach przeznaczonych dla egzaminatora. Tabelki są umieszczone na marginesie przy każdym zadaniu.
13. Pamiętaj, że zapisy w brudnopisie nie będą oceniane.

Zadania egzaminacyjne są wydrukowane na następnych stronach.

Zadanie 1. Szyfrowanie

Algorytm Szyfr szyfruje jawny tekst o długości *n* znaków, który jest zapisany w tablicy *T*. Kluczem szyfrowania jest dodatnia liczba całkowita *k*. Szyfrogram jest zapisywany w tablicy *S*.

Specyfikacja:

Dane:

n – długość tekstu, dodatnia liczba całkowita
T[1..*n*] – tekst jawny, *T*[*i*] – *i*-ty znak w tekście
k – klucz, dodatnia liczba całkowita

Wynik:

S[1..*n*] – szyfrogram, *S*[*i*] – *i*-ty znak w szyfrogramie

Szyfr:

```
a ← n div k
c ← n – a
dla i = 1, 2, …, n wykonuj
    jeżeli i mod k = 0
        b ← i div k
        S [ n – a + b ] ← T [ i ]
    w przeciwnym razie
        S [ c ] ← T [ i ]
        c ← c – 1
```

Uwaga:

← oznacza przypisanie wartości
zapis *x* div *y* oznacza dzielenie całkowite liczby *x* przez liczbę *y*
zapis *x* mod *y* oznacza resztę z dzielenia liczby *x* przez liczbę *y*.

1.1.
0–1–2

Zadanie 1.1. (0–2)

Uzupełnij tabelę – zapisz wynik działania algorytmu (szyfrogram) dla podanych wartości tekstu jawnego i klucza.

Liczba znaków *n*	Tekst jawny	Klucz *k*	Szyfrogram
14	defragmentacja	3	
10	tropikalny	5	

Miejsce na obliczenia (brudnopis)

Zadanie 1.2. (0–1)

1.2.
0–1

Podaj wartość klucza *k*, dla której zawsze szyfrogram i tekst jawny są identyczne.

Odp. *k* = ……

Zadanie 1.3. (0–1)

1.3.
0–1

Podaj wartość klucza *k*, dla której, dla każdego *n* zawsze szyfrogram jest tekstem jawnym zapisanym od końca.
Np. tak jest dla tekstu *serwer* i szyfrogramu *rewres*.
Uwaga: podana wartość może zależeć od *n*.

Odp. ……

Miejsce na obliczenia (brudnopis)

Zadanie 2. Algorytm

Dane są: liczba $n > 1$ oraz tablica $T[1..n]$ zawierająca liczby całkowite. Dla liczby całkowitej $1 \le k < n$ powiemy, że tablica T jest ***k-podobna*** wtedy i tylko wtedy, gdy $T[1..k] = T[n - k + 1..n]$. Ponadto przyjmujemy, że każda tablica jest 0-*podobna* ($k = 0$).

Przykład:
Tablica $T = [1,2,1,1,2,1,1]$ jest 0-*podobna*,1-*podobna* i 4-*podobna*.
Tablica $T = [1,2,2]$ jest 0-*podobna*.
Tablica $T = [2,2,2]$ jest 0-*podobna*,1-*podobna* i 2-*podobna.*

Uwaga:
$T = [2,2,2]$ nie jest 3-podobna, bo k musi być mniejsze niż liczba elementów tablicy (analogicznie $T = [1,2,2]$ nie jest 3-podobna, a $T = [1,2,1,1,2,1,1]$ nie jest 7-podobna).

2.1.
0–1–2

Zadanie 2.1. (0–2)

Uzupełnij tabelę – dla podanych wartości tablicy T wypisz największe takie k, dla którego T jest k-podobna.

Długość tablicy – *n*	**Tablica *T***	**Największe takie *k*, dla którego *T* jest *k*-podobna**
4	[1,2,3,4]	
10	[3,5,3,5,3,5,3,5,3,5]	

2.2.
0–1–2–3–4

Zadanie 2.2. (0–4)

W pseudokodzie lub w zadeklarowanym języku programowania zapisz algorytm, który oblicza największe takie k, dla którego tablica T jest *k-podobna*.

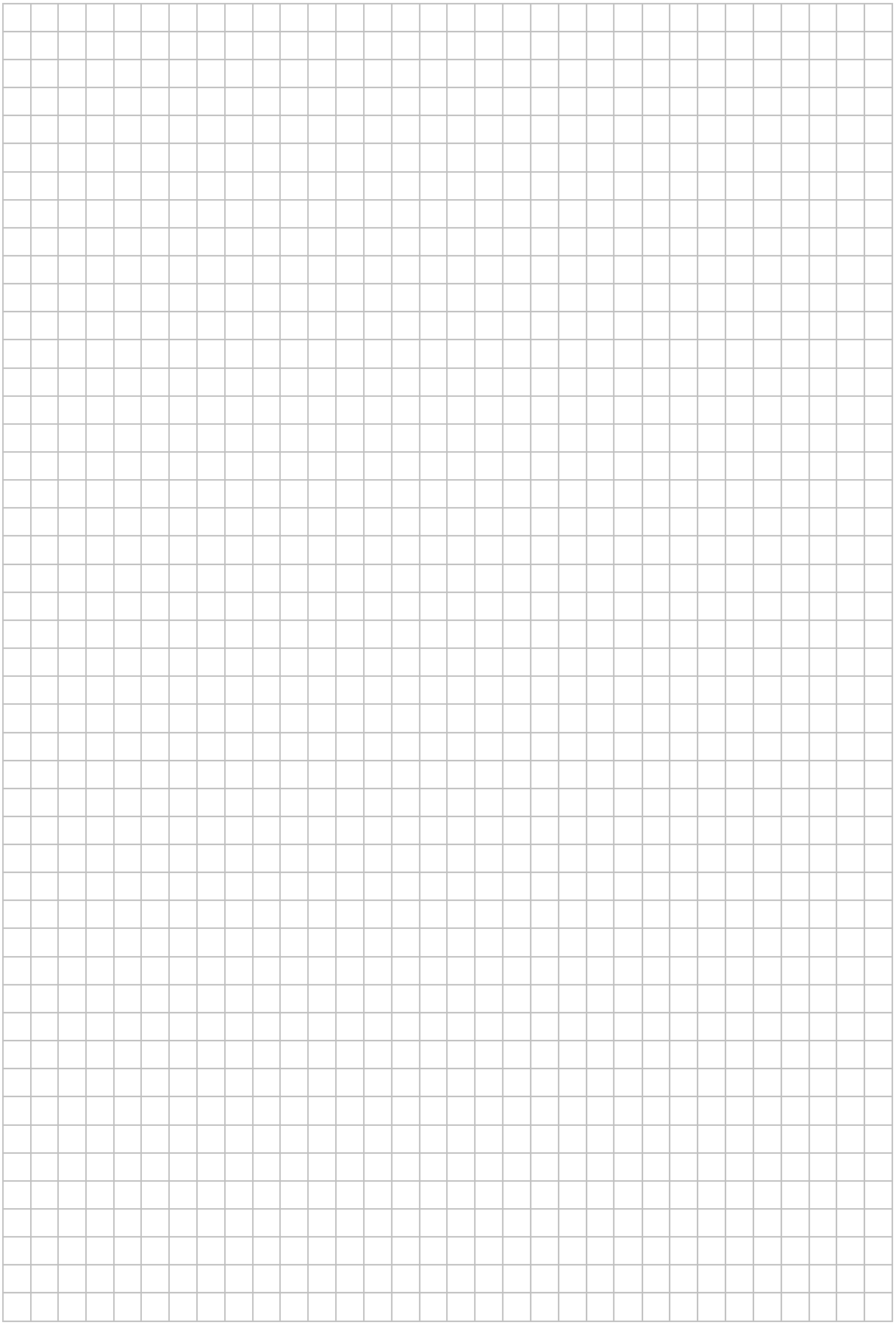
Uwaga: W zapisie algorytmu możesz korzystać tylko z operatorów arytmetycznych: dodawania, odejmowania, mnożenia, dzielenia, dzielenia całkowitego i reszty z dzielenia; operatorów logicznych, porównań liczb, odwoływania się **za pomocą indeksów** do **pojedynczych elementów** tablicy, instrukcji sterujących, instrukcji przypisania lub samodzielnie napisanych funkcji i procedur wykorzystujących powyższe operacje. Zabronione jest używanie funkcji wbudowanych oraz operatorów innych niż wymienione, dostępnych w językach programowania, w szczególności nie wolno używać operacji pobierania fragmentu tablicy (listy).

Specyfikacja:

Dane:
- n – liczba całkowita większa od 1
- $T[1..n]$ – tablica n liczb

Wynik:
- k – największe takie k, dla którego tablica T jest *k-podobna*

Miejsce na zapis algorytmu



Zadanie 3. Pary

Na potrzeby zadań 3.1.–3.3. definiujemy pojęcie *np-pary* następująco:

np-parą będziemy nazywać dwa następujące po sobie znaki napisu, takie że pozycja pierwszego znaku w napisie jest nieparzysta, a drugiego – parzysta, gdzie pozycje znaków w napisie **liczymy od 1**.

Pierwsza *np-para* to pierwsze dwa znaki napisu. Drugą *np-parę* tworzą znaki trzeci i czwarty napisu, trzecią – piąty i szósty itd. Na przykład w napisie MATURA są trzy *np-pary* znaków: MA, TU i RA.

W pliku `dane.txt` znajduje się jeden napis o parzystej długości złożony z wielkich liter alfabetu angielskiego.

Napisz program(y), który(-e) da(-dzą) odpowiedzi do podanych zadań. Odpowiedzi do zadań zapisz w pliku `wyniki3.txt`, a każdą poprzedź numerem odpowiedniego zadania.

Plik `dane_przyklad.txt` zawiera jeden napis o parzystej długości złożony z wielkich liter alfabetu angielskiego. Odpowiedzi dla danych z pliku `dane_przyklad.txt` są podane pod treściami zadań.

3.1.
0–1–2

Zadanie 3.1. (0–2)

Podaj, ile razy występuje sytuacja, że dwie występujące bezpośrednio po sobie *np-pary* znaków są takie same.

Przykład:
Dla napisu CABABC prawidłową odpowiedzią jest:
0
(brak *np-par* spełniających warunki zadania).

Dla napisu ABABAB prawidłową odpowiedzią jest:
2
Zauważ, że liczymy dwa razy ABAB. Nie liczymy par BABA ponieważ nie spełniają warunków zadania (nie są *np-parami*).

Dla danych z pliku `dane_przyklad.txt` (YKFORMYKIN**RMRM**AT**YKYK**ATYKATFOAT) prawidłową odpowiedzią jest
2

3.2.
0–1–2–3

Zadanie 3.2. (0–3)

Fragment napisu złożony z *np-par* znaków nazwiemy *parzyście pierwszym*, jeżeli suma kodów ASCII wszystkich znaków tego fragmentu jest liczbą pierwszą.

Wyznacz i podaj najdłuższy *parzyście pierwszy* fragment napisu z pliku `dane.txt`. Jest tylko jeden taki napis w pliku z danymi.

Przykład:
W napisie NEWSROOM napisem *parzyście pierwszym* jest np. fragment NEWS.

Znak	Kod ASCII
N	78
E	69
W	87
S	83

Suma kodów ASCII dla napisu NEWS: 78+69+87+83 = 317
317 jest liczbą pierwszą.
Najdłuższym *parzyście pierwszym* fragmentem napisu NEWSROOM jest WSROOM (suma kodów ASCII równa 487).

Dla pliku `dane_przyklad.txt` prawidłową odpowiedzią jest
FORMYKINRMRMATYKYKATYKATFO

Zadanie 3.3. (0–3)

3.3.
0–1–2–3

Dla każdej *np-pary* znaków z pliku `dane.txt` policz, ile razy łącznie występuje ona w pliku. Uporządkuj *np-pary* rosnąco według liczby ich wystąpień. Podaj napis, który powstanie po ustawieniu *np-par* zgodnie z tym porządkiem.

Uwaga: W napisie zapisanym w pliku `dane.txt` nie ma dwóch *np-par*, które mają taką samą liczbę wystąpień. Każda *np-para* występuje jednokrotnie w tekście wynikowym.

Przykład:
Dla napisu RAPARA mamy *np-pary* RA, PA i RA, liczba wystąpień to 1 × PA i 2 × RA.
Po uporządkowaniu otrzymujemy tekst PARA
Dla napisu ABCABABACABA mamy 1 × AB, 2 × CA i 3 × BA oraz wynik: ABCABA

Dla pliku `dane_przyklad.txt` poprawna odpowiedź to:
INFORMATYK

Do oceny oddajesz:

- plik tekstowy `wyniki3.txt` zawierający odpowiedzi do zadań 3.1.–3.3. (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- plik(-i) zawierający(-e) kody źródłowe Twoich programów o nazwach odpowiednio (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania):

  zadanie 3.1. .....

  zadanie 3.2. .....

  zadanie 3.3. .....

Zadanie 4. Miasta na mapie

W kartezjańskim układzie współrzędnych (x, y) oznaczono położenie n miast. Położenie każdego miasta jest określone przez punkt o współrzędnych całkowitoliczbowych. Dla każdego miasta podano jego populację w tysiącach mieszkańców.
Między miastami można poruszać się tylko wzdłuż dróg, które biegną w kierunku północ – południe albo wschód – zachód. Odległość między miastami położonymi w punktach o współrzędnych (x_1, y_1) i (x_2, y_2) będziemy liczyć jako

$$d = |x_1 - x_2| + |y_1 - y_2|$$

W pliku tekstowym o nazwie `mapa.txt` znajdują się dane 1000 miast. Każdy wiersz pliku zawiera informacje dotyczące jednego miasta oddzielone spacjami:

- wspX – współrzędna miasta na osi 0X, liczba całkowita z zakresu [-250, 250]
- wspY – współrzędna miasta na osi 0Y, liczba całkowita z zakresu [-250, 250]
- wielkość – liczba mieszkańców wyrażona w tysiącach, większa od 0 i nieprzekraczająca 50000.

Informacje o i-tym mieście (i = 1, 2, …, 1000) zapisane są w i-tym wierszu pliku.

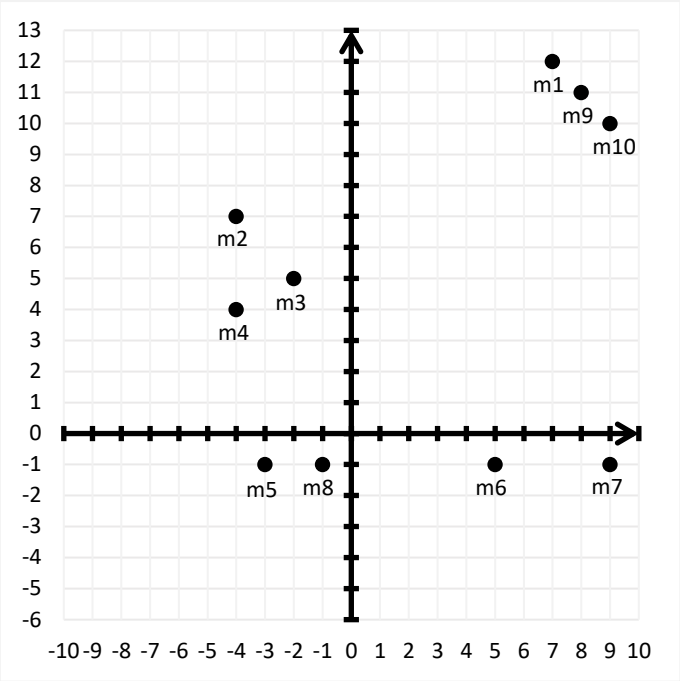
Przykład:

```
7 12 53
-4 7 315
17 77 12
42 -60 253
```

Z wykorzystaniem danych zawartych w pliku `mapa.txt` **napisz program(-y)**, który(-e) da(-dzą) odpowiedzi do zadań 4.1.–4.3. Odpowiedzi zapisz w pliku `wyniki4.txt`, a każdą z nich poprzedź numerem odpowiedniego zadania.

Do dyspozycji masz plik `mapa_przyklad.txt`, który zawiera przykładowy opis mapy składający się z 50 wierszy danych spełniających warunki zadania – możesz sprawdzać na nim działanie swojego programu. Odpowiedzi dla danych z pliku `mapa_przyklad.txt` podano pod treściami zadań.

Do Twojej dyspozycji jest także przykładowa mapa (Rysunek 1.), na której oznaczono położenie 10 miast (dane tych miast zapisano w Tabeli 1.). Odpowiedzi dla danych z tej mapy również umieszczono pod treściami zadań.



Rysunek 1. Przykładowa mapa (mX oznacza miasto numer X)

Nr miasta	wspX	wspY	Wielkość (liczba mieszkańców w tysiącach)
1	7	12	60
2	-4	7	10
3	-2	5	7
4	-4	4	15
5	-3	-1	6
6	5	-1	11
7	9	-1	12
8	-1	-1	20
9	8	11	130
10	9	10	220

Tabela 1. Dane miast przedstawionych na mapie na Rysunku 1.

Zadanie 4.1. (0–2)

4.1.
0–1–2

Dla danych zapisanych w pliku `mapa.txt` podaj numer miasta o największej liczbie mieszkańców. Podaj także liczbę mieszkańców tego miasta w tysiącach. W pliku jest tylko jedno takie miasto.

Dla danych przedstawionych na Rysunku 1. poprawna odpowiedź to
10 220
Dla danych z pliku `mapa_przyklad.txt` poprawna odpowiedź to
10 48967
(w obu przypadkach miasto nr 10 ma największą liczbę mieszkańców).

4.2.
0–1–2–3

Zadanie 4.2. (0–3)

Wyznacz i podaj najmniejszą odległość między dwoma miastami. Podaj także pary numerów miast (każdą parę w oddzielnym wierszu), między którymi jest taka odległość.

Uwaga:
Każdą parę miast wypisujemy tylko raz, np. 1 2 oraz 2 1 to ta sama para.
Pamiętaj, że odległość między miastami liczymy według wzoru podanego na początku zadania.

Dla danych przedstawionych na Rysunku 1. odpowiedź to
2
1 9
5 8
9 10

Dla danych z pliku `mapa_przyklad.txt` odpowiedź to
7
14 15
27 32
(odległość 7, pary miast: numer 14 i 15 oraz 27 i 32).

4.3.
0–1–2–3

Zadanie 4.3. (0–3)

Na potrzeby tego zadania *aglomeracją* będziemy nazywać **największy** obszar spełniający warunki:
a) kształt obszaru jest kwadratem, którego:
- południowo-zachodni (lewy dolny) róg ma współrzędne (0,0)
- północno-wschodni (prawy górny) róg ma współrzędne całkowitoliczbowe dodatnie
- boki są równoległe do osi układu współrzędnych

b) gęstość zaludnienia obszaru jest większa od 2.

Gęstość zaludnienia liczymy, dzieląc sumę ludności miast leżących wewnątrz kwadratu (nie liczymy tych na krawędziach) przez powierzchnię kwadratu.
Dla danych z pliku `mapa.txt` znajdź *aglomerację* oraz podaj:
- długość boku kwadratu tworzącego *aglomerację*
- sumaryczną liczbę mieszkańców *aglomeracji*.

Zauważ, że miasta leżące w *aglomeracji* mogą znajdować się jedynie w I ćwiartce układu współrzędnych. Dla danych opisanych w pliku `mapa.txt` istnieje taka *aglomeracja*.

Dla danych przedstawionych na mapie na Rysunku 1. poprawna odpowiedź to
14 410

Dla danych z pliku `mapa_przyklad.txt` poprawna odpowiedź to
94 17674

Do oceny oddajesz:

- plik tekstowy `wyniki4.txt` zawierający odpowiedzi do zadań 4.1.–4.3. (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- plik(-i) zawierający(-e) kody źródłowe Twoich programów o nazwach odpowiednio (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania):

  zadanie 4.1. .....

  zadanie 4.2. .....

  zadanie 4.3. .....

Zadanie 5. (0–2)

5.
0–1–2

Poniżej zapisano wyrażenie matematyczne zawierające dodatnie liczby całkowite zapisane w systemach: szóstkowym, dziesiętnym i trójkowym.
W miejsce kropek wpisz odpowiednie liczby, tak aby obie równości były prawdziwe

$$1251_6 + \ldots\ldots\ldots\ldots_6 = 500_{10} = \ldots\ldots\ldots\ldots_3 - 110102_3$$

Miejsce na obliczenia (brudnopis)

Zadanie 6. (0–1)

6.
0–1

Oceń prawdziwość podanych zdań. Zaznacz **P**, jeśli zdanie jest prawdziwe, albo **F** – jeśli jest fałszywe.

1.	Stosowanie uwierzytelniania dwuskładnikowego nie gwarantuje braku dostępu dla osób niepowołanych.	**P**	**F**
2.	Stosowanie uwierzytelniania dwuskładnikowego jest możliwe tylko na urządzeniach wyposażonych w kamerę i w mikrofon.	**P**	**F**

Zadanie 7. Recykling

W pliku `dostawy.txt` zapisano informacje na temat dostaw materiałów do magazynu przetwórni, która zajmuje się recyklingiem. Pierwszy wiersz pliku zawiera nazwy kolumn, a dane w wierszach są rozdzielone znakiem tabulacji. W kolejnych wierszach pliku zapisano numer dostawy, datę dostawy, nazwę dostawcy i liczbę kilogramów dostarczonego materiału.

Fragment pliku:

```
Nr_dostawy      Data            Dostawca  Liczba_kg
1               02.01.2023      B-12      634
2               02.01.2023      C-55      385
3               02.01.2023      C-92      526
4               02.01.2023      D-72      899
5               02.01.2023      C-40      655
```

Z wykorzystaniem dostępnych narzędzi informatycznych podaj odpowiedzi do podanych zadań. Odpowiedzi zapisz w pliku `wyniki7.txt`, a każdą z nich poprzedź numerem odpowiedniego zadania.

7.1. 0–1–2

Zadanie 7.1. (0–2)

Podaj datę, kiedy dostarczono łącznie najwięcej materiału do przetwórni oraz podaj liczbę dostarczonych kilogramów. Jest tylko jeden taki dzień.

7.2. 0–1–2

Zadanie 7.2. (0–2)

Podaj, ile razy zdarzyło się, że ten sam dostawca przywiózł materiał dokładnie dwa razy w tym samym dniu.

7.3. 0–1–2–3

Zadanie 7.3. (0–3)

Dla każdego miesiąca podaj łączną liczbę kilogramów materiału dostarczonego do recyklingu.
Do wykonanego zestawienia wykonaj wykres kolumnowy. Pamiętaj o opisie wykresu (tytuł wykresu, nazwy miesięcy na osi 0X).

Informacja do zadań 7.4.–7.5.

Przetwórnia jest uruchamiana jedynie w dniach dostaw i jest w stanie przetworzyć maksymalnie 3900 kg materiału dziennie. Materiał do recyklingu jest pobierany rano z magazynu. Jeżeli w magazynie jest mniej niż 3900 kg lub dokładnie 3900 kg materiału, to przetwórnia przetwarza całość. Jeżeli jest więcej niż 3900 kg, to przetwarzane jest 3900 kg, a pozostała część zostaje w magazynie. Na koniec dnia stan magazynu powiększa się o dostawy z tego dnia.
Rano 2.01.2023, przed rozpoczęciem przetwarzania i przed dostawami, w magazynie przetwórni znajdowało się 5000 kg materiału do recyklingu.

Zadanie 7.4. (0−2)

7.4.
0–1–2

Podaj, jaki był największy stan magazynu na koniec dnia (po przetworzeniu i po dostawie nowego materiału) oraz dzień (datę), kiedy to było. Jest jeden taki dzień.

Zadanie 7.5. (0−2)

7.5.
0–1–2

Podaj z dokładnością do 1 kilograma, jaką maksymalnie powinno się przyjąć dzienną wydajność przetwórni (ile kg dziennie maksymalnie może wytwórnia przetworzyć), żeby – przy takich samych dostawach i założeniach – pracowała każdego dnia pracy z pełną mocą (to jest przetwarzała liczbę kg równą swojej wydajności).

Do oceny oddajesz:

- plik tekstowy `wyniki7.txt` zawierający odpowiedzi do zadań 7.1.–7.5. (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- plik zawierający wykres do zadania 7.3. o nazwie:

  ……
- plik(pliki) zawierający(-e) komputerową realizację Twoich rozwiązań o nazwie(-ach) (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania):

  .....

Zadanie 8. Oceny

W trzech plikach tekstowych zapisano dane dotyczące oceniania grupy uczniów z pewnej szkoły z okresu od września 2023 do czerwca 2024. Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w plikach są rozdzielone znakiem tabulacji. Dane są zapisane bez polskich znaków.

Plik o nazwie `uczniowie.txt` zawiera dane uczniów. Są to:
`Id_Ucznia` – identyfikator ucznia
`Plec` – płeć ucznia (k lub m)
`Id_Klasy` – identyfikator klasy ucznia.

Przykład:

```
Id_Ucznia Plec Id_Klasy
u1        k    D
u2        k    D
u3        m    D
u4        k    C
```

W pliku `nauczyciele.txt` znajdują się dane nauczycieli uczących w szkole:
`Id_Naucz` – identyfikator nauczyciela
`Przedmiot` – nazwa przedmiotu, którego uczy nauczyciel.

Przykład:

```
Id_Naucz  Przedmiot
N01       matematyka
N02       matematyka
N03       jezyk_polski
N04       jezyk_polski
```

Plik `oceny.txt` zawiera zestawienie ocen:
`Id_Oceny` – identyfikator wystawionej oceny
`Id_Ucznia` – identyfikator ucznia
`Data` – data wystawienia oceny w formacie rrrr-mm-dd
`Ocena` – ocena uzyskana przez ucznia
`Id_Naucz` – identyfikator nauczyciela, który wystawił ocenę.

Przykład:

```
Id_Oceny  Id_Ucznia Data        Ocena     Id_Naucz
1         u50       2023-09-04  5         N01
2         u119      2023-09-04  1         N02
3         u31       2023-09-04  5         N02
4         u106      2023-09-04  3         N03
```

Z wykorzystaniem danych zawartych w wymienionych plikach oraz dostępnych narzędzi informatycznych, podaj odpowiedzi do zadań 8.1.–8.5. Odpowiedzi zapisz w pliku `wyniki8.txt`, a każdą z nich poprzedź numerem odpowiedniego zadania.

Zadanie 8.1. (0–2)

Oblicz średnią ocen z matematyki oddzielnie dla kobiet i dla mężczyzn. Wynik podaj z dokładnością do dwóch miejsc po przecinku.

8.1.
0–1–2

Zadanie 8.2. (0–2)

Podaj identyfikatory nauczycieli, którzy nie wystawili żadnej oceny niedostatecznej (1) w okresie od 2023-12-01 do 2023-12-31.

8.2.
0–1–2

Zadanie 8.3. (0–2)

Dla każdego miesiąca podaj: ocenę, która została wystawiona najmniejszą liczbę razy, oraz liczbę wystąpień tej oceny.

8.3.
0–1–2

Do oceny oddajesz:

- plik tekstowy `wyniki8.txt` – zawierający odpowiedzi do zadań 8.1.–8.3. (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- plik(i) zawierający(e) komputerową realizację Twoich obliczeń o nazwie(-ach) (uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania):

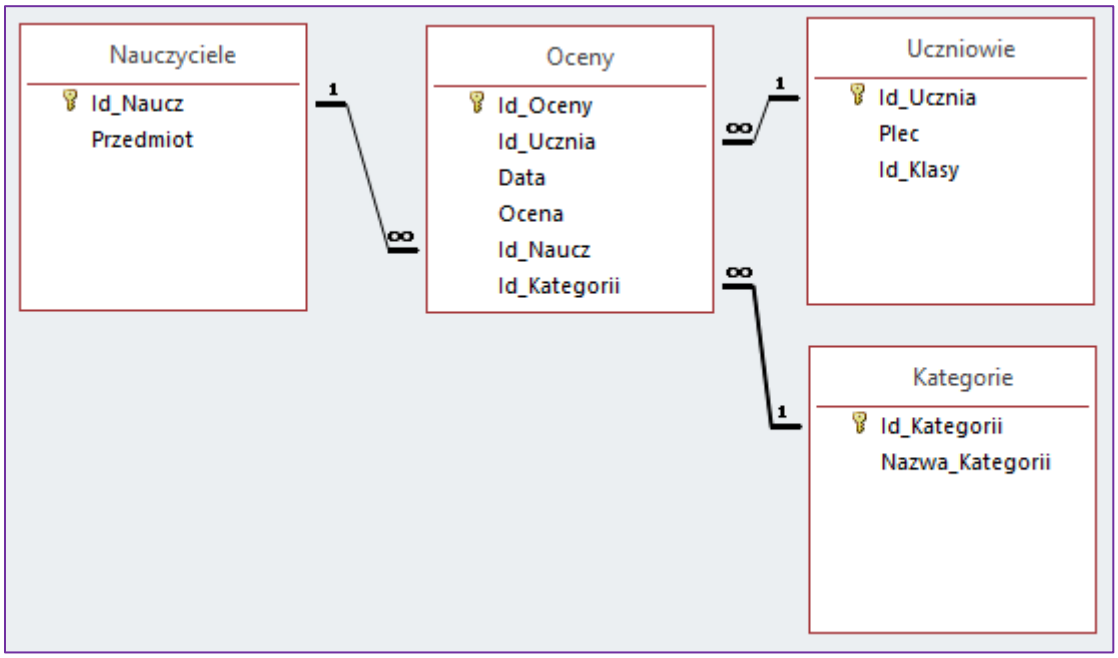
……..

Informacja do zadań 8.4.–8.5.

Do bazy danych wprowadzono dodatkową tabelę *Kategorie* określającą kategorię ocen (np. sprawdzian, kartkówka itp.):

`Id_Kategorii` – identyfikator kategorii
`Nazwa_Kategorii` – nazwa kategorii.

Do tabeli *Oceny* dodano pole `Id_Kategorii`, które określa kategorię wystawionej oceny.



8.4.
0–1–2

Zadanie 8.4. (0–2)

Napisz w języku SQL zapytanie, w wyniku którego uzyskasz zestawienie liczby wystawionych ocen dla każdej kategorii. W zestawieniu podaj nazwę kategorii oraz liczbę wystawionych ocen.

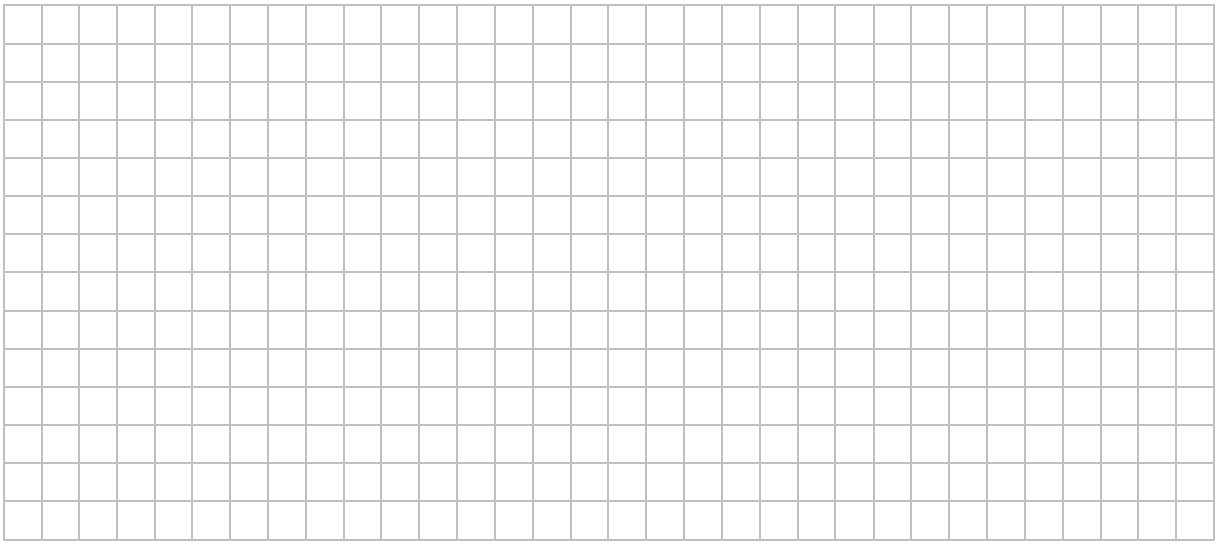
Miejsce na zapis zapytania

Zadanie 8.5. (0–2)

8.5.
0–1–2

Napisz w języku SQL zapytanie, w wyniku którego uzyskasz identyfikator ucznia, który otrzymał najwięcej ocen celujących (6) ze sprawdzianów (nazwa kategorii: `sprawdzian`).

Miejsce na zapis zapytania



BRUDNOPIS (*nie podlega ocenie*)

INFORMATYKA

Poziom rozszerzony

Formuła 2023

INFORMATYKA

Poziom rozszerzony

Formuła 2023

INFORMATYKA

Poziom rozszerzony

Formuła 2023