

**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**-2505

DATA: **14 maja 2025 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.





## Instrukcja dla zdającego

1. Sprawdź, czy arkusz egzaminacyjny zawiera 18 stron (zadania 1–7) i czy dołączony jest do niego nośnik danych – podpisany DANE. Ewentualny brak zgłoś przewodniczącemu zespołu nadzorującego egzamin.
2. Na pierwszej stronie 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.**

Dana jest rekurencyjna funkcja *przestaw*, której parametrem jest nieujemna liczba całkowita:

```


$$r \leftarrow n \bmod 100$$


$$a \leftarrow r \operatorname{div} 10$$


$$b \leftarrow r \bmod 10$$


$$n \leftarrow n \operatorname{div} 100$$

jeżeli  $n > 0$ 
    
$$w \leftarrow a + 10 * b + 100 * \text{przestaw}(n)$$

w przeciwnym razie
    jeżeli  $a > 0$ 
        
$$w \leftarrow a + 10 * b$$

    w przeciwnym razie
        
$$w \leftarrow b$$

wynikiem jest  $w$ 

```

Operator mod oznacza resztę z dzielenia, natomiast div – część całkowitą z dzielenia.

Uzupełnij tabelę – wpisz w drugiej kolumnie wynik funkcji  $przestaw(n)$  dla podanych wartości argumentu  $n$  oraz wpisz w trzeciej kolumnie liczbę wywołań funkcji  $przestaw$  łącznie z pierwszym wywołaniem z parametrem  $n$ .

$n$	Wynik działania funkcji <i>przestaw</i>	Liczba wywołań funkcji <i>przestaw</i>
316498	134689	3
43657688		
154005710		
998877665544321		

[illegible]



### Zadanie 1.3. (0–4)

W postaci pseudokodu lub w wybranym języku programowania napisz **nierekurencyjną** funkcję *przestaw2*, która dla danej nieujemnej liczby całkowitej  $n$  da taką samą wartość jak *przestaw*( $n$ ).

**Uwaga:** Twój algorytm może używać **wyłącznie zmiennych przechowujących liczby całkowite** oraz może operować **wyłącznie na liczbach całkowitych**. 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. **Zabronione** jest używanie funkcji wbudowanych oraz operatorów innych niż wymienione, **w tym – funkcji przestaw**.

### Specyfikacja:

*Dane*

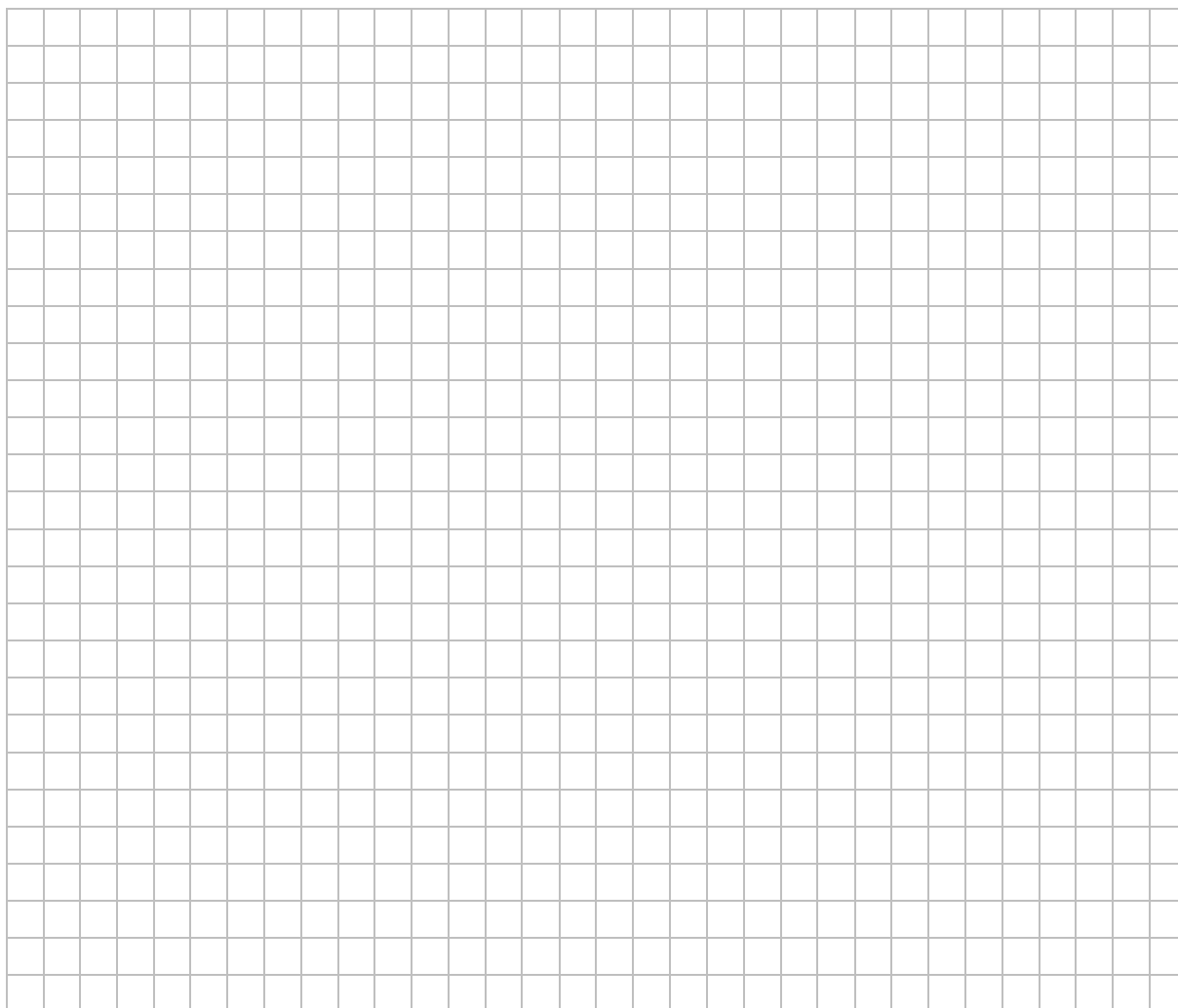
$n$  – nieujemna liczba całkowita

### Wynik

$w$  – nieujemna liczba całkowita, wynik działania taki sam jak po wykonaniu  $przestaw(n)$

*Miejsce na zapis algorytmu*

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 small squares across the entire surface. There are no margins, text, or other markings on the paper.



## Zadanie 2. Zapis symboliczny

W pliku `symbole.txt` zapisano 2000 napisów. Każdy z nich jest zapisany w osobnym wierszu i składa się z dokładnie 12 znaków spośród: `o`, `+`, `*`.

**Napisz program** (lub kilka programów) znajdujący(-ch) odpowiedzi do podanych zadań. Każdą odpowiedź zapisz w pliku `wyniki2.txt` i poprzedź ją numerem oznaczającym zadanie.

Do Twojej dyspozycji jest plik `symbole_przyklad.txt`, który zawiera 20 wierszy danych spełniających warunki zadania. Odpowiedzi dla pliku `symbole_przyklad.txt` są podane pod każdym zadaniem.

Pamiętaj, że Twój program musi ostatecznie zadziałać na pliku `symbole.txt` zawierającym 2000 napisów.

2.1.

0-1-2

### Zadanie 2.1. (0-2)

Podaj wszystkie takie napisy z pliku `symbole.txt`, które są palindromami (czytane od przodu i od tyłu są takie same). Wypisz je po jednym w wierszu, w kolejności takiej jak w pliku `symbole.txt`.

Odpowiedź dla pliku `symbole_przyklad.txt` to

`oooo+**+oooo`

(w tym pliku jest jeden palindrom)

2.2.

0-1-  
2-3-4

### Zadanie 2.2. (0-4)

W pliku `symbole.txt` szukamy „kwadratów” złożonych z dziewięciu sąsiadujących identycznych symboli:

<code>+ + +</code>		<code>o o o</code>		<code>* * *</code>
<code>+ + +</code>	lub	<code>o o o</code>	lub	<code>* * *</code>
<code>+ + +</code>		<code>o o o</code>		<code>* * *</code>

Podaj, ile takich kwadratów występuje w pliku `symbole.txt`. Jeżeli w pliku występuje jeden taki kwadrat, podaj numer wiersza i numer pozycji w wierszu (licząc od 1) jego **środkowego pola**. Jeżeli jest więcej takich kwadratów, podaj numer wiersza i numer pozycji w wierszu dla środkowego pola każdego z nich.

### Przykład:

Poniżej podano 6 wierszy przykładowych danych (po 12 znaków w każdym wierszu):

- `+ * * + o * o + + * o +`
- `+ + + o o o o * o * * *`
- `+ o * o o o o * * + + +`
- `* + * o o o o o o + + +`
- `o * * o + + + o + + + +`
- `o o o o + + * * + * + o`





Mamy tutaj trzy kwadraty złożone z 9 identycznych symboli: pierwszy ma środek w wierszu 3 na pozycji 5, drugi – w wierszu 3 na pozycji 6, a trzeci – w wierszu 4 na pozycji 11.

Odpowiedź dla pliku `symbole_przyklad.txt` to

1 6 3

(jeden kwadrat, który ma środkowe pole w wierszu 6, na pozycji 3).

### Informacja do zadań 2.3. i 2.4.

Każdy z napisów podanych w pliku `symbole.txt` będziemy traktować jako liczbę zapisaną w systemie trójkowym, w którym:

znak `o` odpowiada cyfrze 0

znak `+` odpowiada cyfrze 1

znak `*` odpowiada cyfrze 2.

### Zadanie 2.3. (0–2)

Podaj największą liczbę spośród liczb zapisanych w pliku `symbole.txt`. W odpowiedzi podaj tę liczbę w zapisie dziesiętnym oraz napis jej odpowiadający.

Odpowiedź dla pliku `symbole_przyklad.txt` to

519789 \*\*\*+o\*ooo++o

2.3.

0–1–2

### Zadanie 2.4. (0–3)

Oblicz sumę wszystkich liczb z pliku `symbole.txt`. Podaj jej wartość w zapisie dziesiętnym oraz w zapisie trójkowym z użyciem symboli: `o`, `+`, `*`.

Odpowiedź dla pliku `symbole_przyklad.txt` to

4841542 +oooo\*\*\*\*\*+oo+o+

2.4.

0–1–

2–3

### Do oceny oddajesz:

- plik `wyniki2.txt` – zawierający odpowiedzi do zadań 2.1.–2.4.  
(odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- pliki zawierające kody źródłowe Twojego(-ich) programu(-ów) o nazwach  
(uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania):

zadanie 2.1. ....

zadanie 2.2. ....

zadanie 2.3. ....

zadanie 2.4. ....

### Zadanie 3. Dron

Tor lotu pewnego drona składa się z prostych odcinków. Lot rozpoczyna się w punkcie  $(0, 0)$ , a kończy w punkcie  $(20000, 0)$ . Dron poza startem i lądowaniem jest zawsze na wysokości większej od zera.

Plik `dron.txt` zawiera 100 wierszy, w których zapisano dane dotyczące ruchu drona.

W każdym wierszu jest zapisana para liczb całkowitych rozdzielonych znakiem spacji.

Pierwsza liczba oznacza przemieszczenie drona (odległość) w poziomie od ostatniej pozycji – jest to zawsze liczba dodatnia. Druga liczba oznacza przemieszczenie w pionie od ostatniej pozycji. Jeśli druga liczba jest dodatnia, to dron wykonał ruch w górę, jeśli ujemna – w dół, a jeśli równa 0 – nie zmieniał wysokości.

#### Przykład 1.

Dla przykładowych danych:

3000 2000

2000 9000

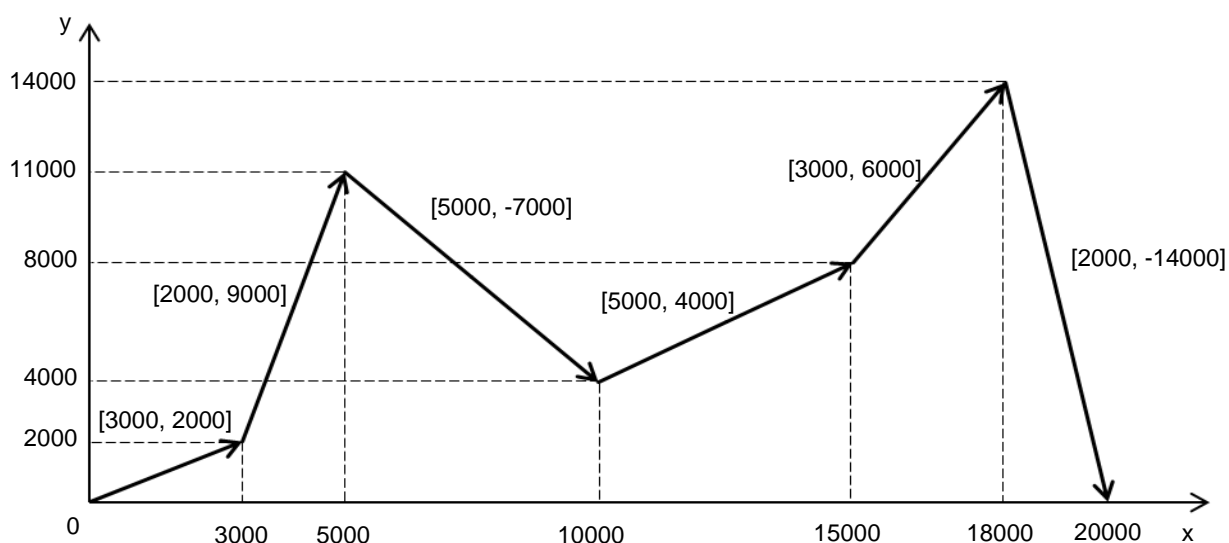
5000 -7000

5000 4000

3000 6000

2000 -14000

lot drona można zilustrować na wykresie:



gdzie:

$x$  – odległość w poziomie od punktu startowego

$y$  – wysokość (odległość w pionie od punktu startowego)

$[A, B]$  – umieszczone na wykresie pary liczb oznaczające przemieszczenia drona odpowiednio w poziomie i w pionie.

**Napisz program** (lub kilka programów), który(-e) znajdzie(-dą) odpowiedzi dla podanych zadań. Każdą odpowiedź zapisz w pliku `wyniki3.txt` i poprzedź ją numerem oznaczającym zadanie.

Do Twojej dyspozycji jest plik `dron_przyklad.txt` zawierający 10 wierszy danych w opisanej postaci. Odpowiedzi dla pliku `dron_przyklad.txt` są podane pod każdym zadaniem. Pamiętaj, że Twój program musi ostatecznie zadziałać na pliku `dron.txt` zawierającym 100 wierszy danych.

### Zadanie 3.1. (0–2)

Dla każdego przesunięcia  $[A, B]$  zapisanego w pliku `dron.txt` oblicz największy wspólny dzielnik (NWD) wartości bezwzględnych liczb  $A$  i  $B$ . Podaj **liczbę** par  $[A, B]$ , dla których największy wspólny dzielnik wartości bezwzględnych liczb  $A$  i  $B$  jest większy od 1.

**Uwaga:** przyjmujemy, że  $\text{NWD}(A, 0) = A$ .

Odpowiedź dla pliku `dron_przyklad.txt` to  
6

3.1.

0–1–2

### Zadanie 3.2. (0–4)

Rozważmy **wszystkie punkty**, w których dron znajdował się **po wykonaniu** kolejnych ruchów (przesunięć).

Dla danych z przykładu 1. będą to punkty: (3000, 2000), (5000, 11000), (10000, 4000), (15000, 8000), (18000, 14000) i (20000, 0).

- Podaj, ile spośród wszystkich rozważanych punktów znajduje się wewnątrz kwadratu o wierzchołkach (0, 0), (0, 5000), (5000, 5000), (5000, 0). Nie liczymy punktów leżących na krawędziach kwadratu.
- Spośród wszystkich rozważanych punktów znajdź i podaj trzy różne, takie, że jeden z nich jest środkiem odcinka o końcach w pozostałych dwóch. Jest tylko jedna taka trójka punktów.  
**Uwaga:** punkty należące do szukanej trójki nie muszą być trzema kolejnymi punktami, do których przemieszczał się dron.

Odpowiedź dla pliku `dron_przyklad.txt` to

- 2
- (14000, 3014), (16000, 2010), (18000, 1006)

3.2.

0–1–

2–3–4

### Do oceny oddajesz:

- plik `wyniki3.txt` – zawierający odpowiedzi do zadań 3.1.–3.2.  
(odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- pliki zawierające kody źródłowe Twojego(-ich) programu(-ów) o nazwach  
(uwaga: brak tych plików jest równoznaczny z brakiem rozwiązania zadania):

zadanie 3.1. ....

zadanie 3.2. ....

0-1



Program typu *keylogger* służy do

- A.** szyfrowania informacji do postaci uniemożliwiającej jej odczytanie bez zdefiniowanego klucza.
- B.** przechowywania danych logowania, w tym haseł, w bezpiecznym miejscu na dysku użytkownika.
- C.** generowania kodu, który umożliwia użytkownikowi bankowości elektronicznej wykonanie operacji.
- D.** przechwytywania i gromadzenia informacji o naciśniętych klawiszach.

0-1-2



Poniżej sposobem pisemnym dodano dwie liczby podane w zapisie binarnym. Uzupełnij brakujące cyfry tak, aby działanie było wykonane poprawnie.

$$\begin{array}{cccccccccc}
 1 & 1 & \boxed{\phantom{00}} & 0 & 1 & 0 & 1 & 1 & 0 & \boxed{\phantom{00}} & 1 \\
 + & 1 & 1 & 0 & 0 & \boxed{\phantom{00}} & 1 & 0 & 1 & 1 & 1 \\
 \hline
 1 & \boxed{\phantom{00}} & 0 & 1 & 1 & 0 & 0 & 1 & \boxed{\phantom{00}} & \boxed{\phantom{00}} & 1 & 0
 \end{array}$$

*Miejsce na obliczenia (brudnopis)*

A full-page view of a blank sheet of graph paper. The grid consists of small, uniform squares formed by thin gray lines. There are no margins, text, or other markings on the page.

## Zadanie 6. Martianeum

W 2033 roku na Marsie wylądowała automatyczna stacja wydobywcza wyposażona w transporter i w autonomiczny dron pobierający ładunki skał zawierających minerał niewystępujący na Ziemi – martianeum.

Stacja działa według następujących zasad:

- dron codziennie przywozi ładunek z pewnego obszaru Marsa
- stacja waży ładunek przywieziony przez drona i bada zawartość martianeum
- jeśli zawartość martianeum w przywiezionym ładunku wynosi co najmniej 1%, to stacja automatycznie wydobywa cały minerał z tego ładunku
- jeśli na koniec dnia (po wydobyciu martianeum) ilość minerału na stacji osiągnie co najmniej 100 kg, to transporter zabiera 100 kg na orbitę, skąd ładunek jest wysyłany na Ziemię, a transporter wraca do stacji (jeśli na stacji zgromadzone jest więcej niż 100 kg, to nadmiar pozostaje na stacji)
- początkowy stan magazynu na stacji – 0 kg martianeum.

W pliku tekstowym `martianeum.txt` w kolejnych wierszach zapisano dane z lat 2033–2038:

<code>data</code>	– data przywozu ładunku w formacie <code>rrrr-mm-dd</code>
<code>nazwa_obszaru</code>	– nazwa obszaru Marsa, z którego ładunek został pobrany
<code>masa [kg]</code>	– masa ładunku drona w kilogramach
<code>zawartość [%]</code>	– zawartość martianeum w próbce w % (nieujemna liczba z jednym miejscem po przecinku, np. 0,1 oznacza 0,1%)

Dane w pliku rozdzielono znakami tabulacji.

### Przykład:

<code>data</code>	<code>nazwa_obszaru</code>	<code>masa [kg]</code>	<code>zawartosc [%]</code>
2033-03-03	Cebrenia	27,8	0,2
2033-03-04	Amenthes	11,8	1,7
2033-03-05	Noachis	21,0	6,0
2033-03-06	Coprates	26,3	11,4
2033-03-07	Ismenius Lacus	28,8	0,0
2033-03-08	Mare Boreum	29,2	0,0

Z wykorzystaniem danych zawartych w pliku `martianeum.txt` oraz dostępnych narzędzi informatycznych wykonaj podane zadania. Wyniki zapisz w pliku tekstowym `wyniki6.txt`. Odpowiedź do każdego zadania poprzedź numerem tego zadania.

### Zadanie 6.1. (0–2)

Podaj łączną masę ładunków drona oraz łączną masę martianeum wydobytego przez stację.

6.1.

0–1–2

6.2.

0-1

**Zadanie 6.2. (0-1)**

Podaj nazwę obszaru, dla którego średnia masa przywiezionych ładunków jest najmniejsza.

6.3.

0-1-2

**Zadanie 6.3. (0-2)**

Czas pracy stacji dzielimy na kolejne 7-dniowe okresy. Pierwszy okres obejmuje dni od 03.03.2033 do 09.03.2033, drugi – od 10.03.2033 do 16.03.2033 itd.

Podaj największą łączną masę ładunków przywiezionych w ciągu kolejnych 7-dniowych okresów oraz podaj datę początku okresu, w którym przywieziono tę największą masę.

6.4.

0-1-  
2-3**Zadanie 6.4. (0-3)**

Wykonaj zestawienie, w którym dla każdego obszaru podasz, ile razy dron przewoził ładunek z tego obszaru w poszczególnych latach.

Na podstawie wykonanego zestawienia utwórz wykres skumulowany kolumnowy. Pamiętaj o czytelnym opisie wykresu: na osi X umieść nazwy obszarów, dodaj opisy osi – „nazwy obszarów” dla osi X i „liczba przewozów ładunku” dla osi Y, tytuł oraz legendę zawierającą kolejne lata.

6.5.

0-1-  
2-3**Zadanie 6.5. (0-3)**

Uwzględnij zasady działania stacji opisane na początku zadania i podaj:

- ile razy stacja wysłała ładunek na orbitę
- datę pierwszego transportu ładunku na orbitę
- datę ostatniego transportu ładunku na orbitę.

**Do oceny oddajesz:**

- plik tekstowy `wyniki6.txt` zawierający odpowiedzi do zadań 6.1.–6.5. Odpowiedź do każdego zadania powinna być poprzedzona jego numerem.
- plik zawierający wykres do zadania 6.4. o nazwie:

.....

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

.....



## Zadanie 7. Poszukiwanie wody na Marsie

W trzech plikach tekstowych o nazwach `laziki.txt`, `obszary.txt`, `pomiary.txt` zapisano informacje zawierające dane o poszukiwaniu wody na Marsie w latach 2050–2080. Łaziki zasilane energią słoneczną poruszają się po różnych obszarach Marsa i wykonują pomiary georadarowe, na podstawie których szacują ilość wody i głębokość, na której się ona znajduje. Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w wierszach rozdzielono znakami tabulacji.

Plik o nazwie `laziki.txt` zawiera informacje o różnych łazikach, które wykonywały pomiary. W każdym wierszu tego pliku znajdują się:

<code>nr_lazika</code>	– co najwyżej trzycyfrowy, unikatowy numer łazika
<code>nazwa_lazika</code>	– nazwa łazika (tekst do 50 znaków)
<code>rok_wyslania</code>	– rok startu z Ziemi
<code>wsp_ladowania</code>	– współrzędne lądowania na Marsie oddzielone znakiem przecinka i spacją

### Przykład:

<code>nr_lazika</code>	<code>nazwa_lazika</code>	<code>rok_wyslania</code>	<code>wsp_ladowania</code>
1	Mariner 3	2049	50.51N, 70.01E
2	Mariner 6	2050	11.90N, 119.49E
3	Mariner 7	2050	44.90S, 130.80W

Plik o nazwie `obszary.txt` zawiera informacje o obszarach na Marsie. W każdym wierszu tego pliku znajdują się:

<code>kod_obszaru</code>	– pięciodziesiętny, unikatowy kod obszaru
<code>nazwa_obszaru</code>	– nazwa obszaru (tekst do 50 znaków)

### Przykład:

<code>kod_obszaru</code>	<code>nazwa_obszaru</code>
MC-01	Mare Boreum
MC-02	Diacria
MC-03	Arcadia

Plik o nazwie `pomiary.txt` zawiera informacje o wynikach badań georadarowych wykonanych przez łaziki. W każdym wierszu tego pliku znajdują się:

<code>nr_lazika</code>	– co najwyżej trzycyfrowy numer łazika
<code>data_pomiaru</code>	– data wykonania pomiaru (w formacie rrrr-mm-dd)
<code>kod_obszaru</code>	– pięciodziesiętny kod obszaru, na którym został wykonany pomiar
<code>wspolrzedne</code>	– współrzędne wykonania pomiaru, oddzielone znakiem przecinka i spacją
<code>glebokosc</code>	– szacowana głębokość, na której znajduje się woda (w metrach)
<code>ilosc</code>	– szacowana ilość wody (w m <sup>3</sup> )

**Przykład:**

nr_ łazika	data_ pomiaru	kod_ obszaru	wspolrzedne	glebokosc	ilosc
17	2061-06-03	MC-13	13.17N, 77.80E	344	5622
17	2056-06-02	MC-14	14.93N, 106.00E	43	2054
47	2075-10-18	MC-05	45.57N, 3.30E	9	23366

Z wykorzystaniem danych zawartych w podanych plikach oraz dostępnych narzędzi informatycznych podaj odpowiedzi do zadań 7.1–7.4. 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 nazwę obszaru, na którym znaleziono łącznie we wszystkich pomiarach najwięcej m<sup>3</sup> wody na głębokości do 100 metrów włącznie. Jest jeden taki obszar.

**7.2.**

0–1–2

**Zadanie 7.2. (0–2)**

Podaj nazwę łazika, który wykonywał pomiary w najdłuższym okresie, licząc od pierwszego (najwcześniejszego) do ostatniego (najpóźniejszego) pomiaru. Podaj datę pierwszego i ostatniego pomiaru wykonanego przez ten łazik.

**7.3.**

0–1–2

**Zadanie 7.3. (0–2)**

Podaj nazwy obszarów na Marsie, na których żaden z łazików nie wykonał ani jednego pomiaru w tym samym roku, w którym został wysłany z Ziemi.

**7.4.**

0–1–2

**Zadanie 7.4. (0–2)**

Podaj nazwy łazików, które wylądowały na półkuli południowej, ale wykonywały pomiary na obu półkulach: północnej (N) i południowej (S).

**Do oceny oddajesz:**

- plik tekstowy `wyniki7.txt` – zawierający odpowiedzi do poszczególnych zadań (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):





### Zadanie 7.5. (0-2)

7.5.
0-1-2

kraj – kraj producenta

Napisz w języku SQL zapytanie, w wyniku którego otrzymasz listę nazw producentów, których łaziki badały obszar Marsa o nazwie Arcadia w roku 2060. Nazwy producentów nie mogą się powtarzać.

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.

## BRUDNOPIS (*nie podlega ocenie*)



# INFORMATYKA

Poziom rozszerzony

*Formuła 2023*



# INFORMATYKA

Poziom rozszerzony

*Formuła 2023*



# INFORMATYKA

Poziom rozszerzony

*Formuła 2023*

