

**WYPEŁNIA ZDAJĄCY**

**KOD**

--	--	--

**PESEL**

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

**Egzamin maturalny**

**Formuła 2023**

**INFORMATYKA**

**Poziom rozszerzony**


**Próbna Matura z Operonem 2025/2026**

DATA: **21 listopada 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. Rekurencja na tablicy

Dana jest funkcja rekurencyjna `tablica`, która ma trzy parametry:

`A[1..n,1..n]` – tablica dwuwymiarowa  $n \times n$  dodatnich liczb całkowitych  
`n` – dodatnia liczba całkowita będąca rozmiarem tablicy `A`  
`i` – dodatnia liczba całkowita

```
funkcja tablica(A, i, n)
    jeżeli i = n*n to
        zwróć 0
    w ← (i div n) + 1
    k ← (i mod n) + 1
    zwróć A[w,k] + tablica(A,i+1,n)
```

**Uwaga:** Operator `mod` oznacza resztę z dzielenia, natomiast `div` – część całkowitą z dzielenia.

1.1.

0–1–2

#### Zadanie 1.1. (0–2) 📄

Przeanalizuj funkcję `tablica` dla podanych argumentów `A`, `i` oraz `n`. Uzupełnij tabelę – wpisz do ostatniej kolumny zwracany wynik funkcji `tablica(A, i, n)`.

A	i	n	tablica(A, i, n)
2 1 5 3	0	2	11
3 4 5 1	0	2	
1 2 1 3 1 1 2 1 4	0	3	

1.2.

0–1

#### Zadanie 1.2. (0–1) 📄

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

Niech  $n$  będzie wymiarem tablicy dwuwymiarowej `A[1..n, 1..n]`. Liczba wywołań funkcji `tablica(A, 0, n)` jest równa:

1.	$n$	P	F
2.	$n \cdot \log_2 n$	P	F
3.	$n^2$	P	F

### Zadanie 1.3. (0–3)

W postaci pseudokodu lub w wybranym języku programowania napisz nierekurencyjną funkcję `tablica2`, która dla tablicy `A` dodatnich liczb całkowitych o rozmiarze  $n \times n$  zwróci ten sam wynik, co funkcja `tablica(A, 0, 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 tablica**.

## Specyfikacja

Dane:

$A[1..n, 1..n]$  – tablica dwuwymiarowa  $n \times n$  dodatnich liczb całkowitych

n – dodatnia liczba całkowita będąca rozmiarem tablicy A

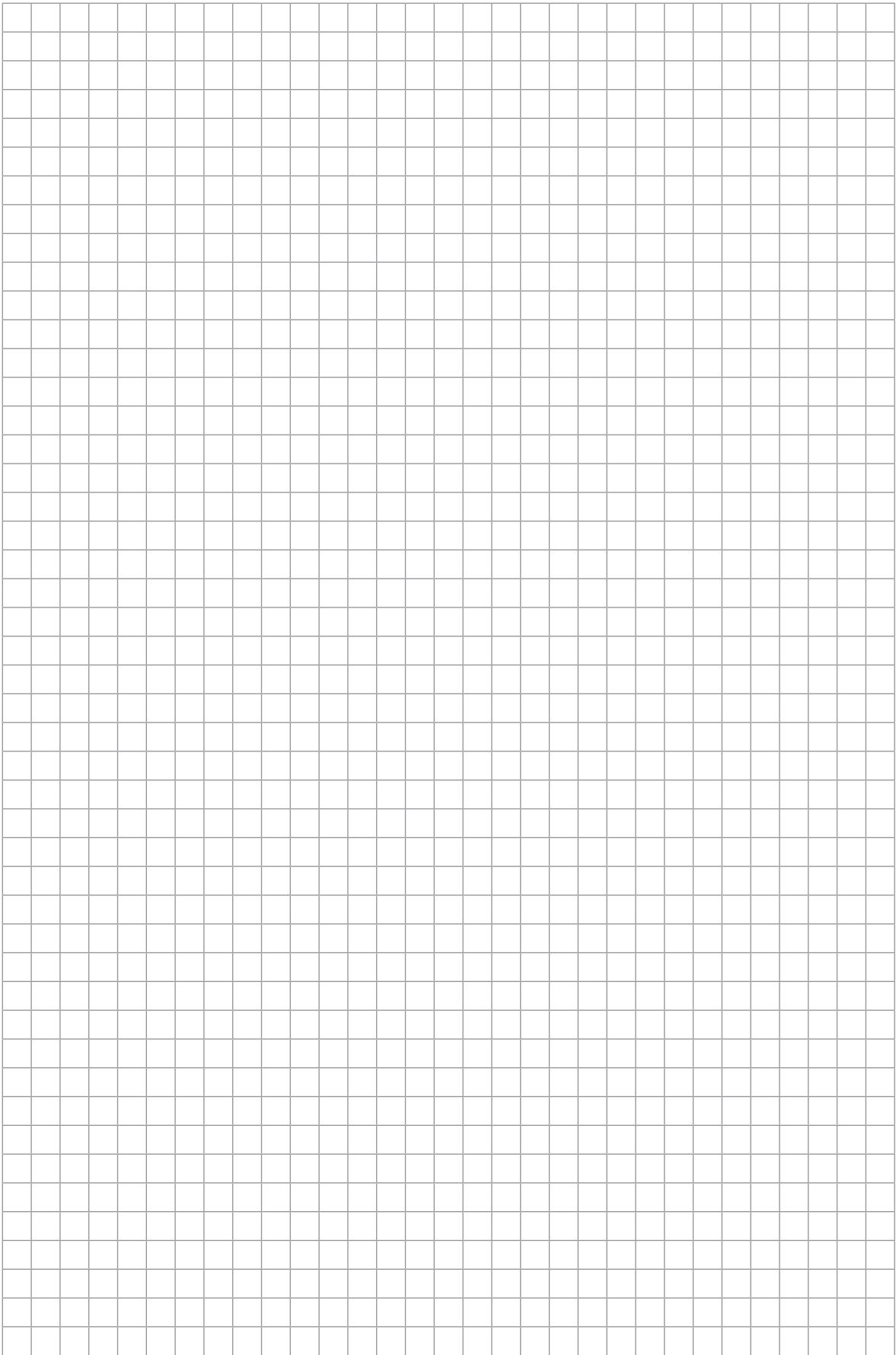
i – dodatnia liczba całkowita

**Wynik:**

dodatnia liczba całkowita

*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. Schemat Hornera

Dany jest wielomian  $A(x)$  stopnia  $n$  o współczynnikach  $a_i$  dla  $i \in \{0, 1, 2, \dots, n\}$ :

$$A(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

Wielomian  $A(x)$  możemy również zapisać w postaci ciągu współczynników  $a_i$ :

$$A(x) = (a_n, a_{n-1}, \dots, a_1, a_0)$$

Chcemy obliczyć  $A(x_0)$ . W tym celu wystarczy użyć tradycyjnego sposobu, czyli podstawić  $x_0$  do wielomianu  $A(x)$ :

$$A(x_0) = a_n x_0^n + a_{n-1} x_0^{n-1} + \dots + a_1 x_0 + a_0$$

Jak łatwo zauważyć, rachunki są nieefektywne, ponieważ liczba mnożeń i dodawań rośnie wraz ze stopniem wielomianu.

Aby obliczyć wartość wielomianu bardziej efektywnie, można przepisać go w postaci zagnieżdżonej:

$$A(x) = (\dots((a_n x + a_{n-1})x + a_{n-2})x + \dots + a_1)x + a_0$$

Liczba wykonywanych mnożeń dla tak zapisanego wielomianu jest zdecydowanie mniejsza.

**Schemat Hornera** to algorytm służący do szybkiego obliczania wartości wielomianu. Redukuje liczbę mnożeń i dodawań do minimum niezbędnego do uzyskania wyniku.

### Zadanie 2.1. (0–2)

Wielomian  $A(x)$  zapisany w tradycyjnej formie:

$$A(x) = 2x^3 - 4x^2 + 5x + 3$$

możemy zapisać w postaci tablicy  $A$  zawierającej współczynniki wielomianu  $A$ :

$$A = [2, -4, 5, 3]$$

Do wyznaczenia wartości wielomianu tradycyjnym sposobem należy wykonać 6 mnożeń i 3 dodawania.

$$A(x) = 2 \cdot x \cdot x \cdot x + (-4) \cdot x \cdot x + 5 \cdot x + 3$$

**Uzupełnij tabelę – podaj liczbę mnożeń i dodawań potrzebnych do wyznaczenia w tradycyjny sposób wartości wielomianu  $A$ , którego współczynniki podano w kolumnie  $A$ .**

n	A	Liczba mnożeń	Liczba dodawań
3	2, -4, 5, 3	6	3
5	2, -1, 4, 3, -7, 1		
n	$a_n, a_{n-1}, \dots, a_1, a_0$		

2.1.

0–1–2

2.2.

0–1–2

### Zadanie 2.2. (0–2)

Wielomian:

$$A(x) = 2x^3 - 4x^2 + 5x + 3$$

możemy za pomocą schematu Hornera zapisać w postaci:

$$A(x) = ((2 \cdot x - 4) \cdot x + 5) \cdot x + 3$$

Uzupełnij tabelę – podaj liczbę mnożeń i dodawań potrzebnych do wyznaczenia za pomocą schematu Hornera wartości wielomianu  $A$ , którego współczynniki podano w kolumnie **A**.

n	A	Liczba mnożeń	Liczba dodawań
3	2, -4, 5, 3	3	3
5	2, -1, 4, 3, -7, 1		
$n$	$a_n, a_{n-1}, \dots, a_1, a_0$		

2.3.

0–1–2

### Zadanie 2.3. (0–2)

Uzupełnij luki oznaczone wy kropkowaną linią w poniższym algorytmie schematu Hornera.

#### Specyfikacja

Dane:

$n$  – dodatnia liczba całkowita będąca stopniem wielomianu  $A$

$A$  – tablica liczb całkowitych – współczynników wielomianu  $A$ ,  $A = [a_n, a_{n-1}, \dots, a_1, a_0]$

$a$  – liczba całkowita, dla której należy policzyć wartość wielomianu  $A(x)$

Wynik:

wartość wielomianu  $A(a)$

#### Algorytm

funkcja horner( $n$ ,  $a$ ,  $A$ )

wynik  $\leftarrow$  .....

dla  $i = n-1, n-2, \dots, 1, 0$  wykonuj

wynik  $\leftarrow$  .....

zwróć wynik

### Informacja do zadań 2.4. i 2.5.

Plik wielomiany.txt zawiera 1000 wierszy, natomiast plik przyklad\_wielomiany.txt – 10 wielomianów. W każdym wierszu obu plików znajduje się jeden wielomian zapisany w sposób następujący:

- wszystkie liczby w wierszu są oddzielone pojedynczym odstępem,
- pierwsza liczba w wierszu oznacza stopień wielomianu,
- kolejne liczby to współczynniki wielomianu, zapisane w kolejności od  $a_n$  do  $a_0$ .

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

### Zadanie 2.4. (0–2)

Dla każdego wielomianu oblicz liczbę mnożeń potrzebnych do wyznaczenia wartości wielomianu  $A$  w sposób tradycyjny i za pomocą schematu Hornera. Podaj wszystkie wielomiany, dla których różnica między liczbą mnożeń w sposobie tradycyjnym a liczbą mnożeń w schemacie Hornera jest największa.

Dla danych zawartych w pliku `przyklad_wielomiany.txt` poprawną odpowiedzią jest:

6 40 -38 39 -47 -26 -36 2

2.4.

0–1–2

### Zadanie 2.5. (0–3)

Dla każdego wielomianu oblicz jego wartość dla  $x = 2$ . Podaj wszystkie wielomiany z pliku `wielomiany.txt`, dla których otrzymana wartość należy do ciągu Fibonacciego.

Dla danych zawartych w pliku `przyklad_wielomiany.txt` poprawną odpowiedzią jest:

3 1 0 0 5

5 1 1 1 1 1 -28

2.5.

0–1

2–3

Do oceny oddajesz:

- plik tekstowy `wyniki2.txt` – zawierający odpowiedzi do zadań 2.4. i 2.5. (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- plik (lub pliki) zawierający kod źródłowy (lub kody źródłowe) twojego programu o nazwie (uwaga: brak tego pliku lub plików jest równoznaczny z brakiem rozwiązania zadania):

.....

### Zadanie 3. Hasła

Hasło powinno łączyć w sobie cztery podstawowe cechy: **długość**, **złożoność**, **unikalność** i **losowość**. Im dłuższe jest hasło i im więcej różnych typów znaków zawiera, tym jest trudniejsze do odgadnięcia.

W pliku `hasla.txt` zapisano 1000 ciągów znaków składających się z: małych i dużych liter alfabetu łacińskiego (bez polskich znaków), cyfr i znaków specjalnych: \$ ! %. Każde hasło liczy od 8 do 16 znaków.

Plik `przyklad_hasla.txt` zawiera 25 przykładowych haseł. Odpowiedzi dla tego pliku są podane po każdym zadaniu.

### Zadanie 3.1. (0–2)

Podaj liczbę haseł w pliku `hasla.txt`, które składają się z co najmniej 12 znaków i każde z nich zawiera przynajmniej jeden z trzech znaków specjalnych: \$ ! %.

Dla pliku `przyklad_hasla.txt` poprawną odpowiedzią jest 8.

3.1.

0–1–2

3.2.

0–1  
2–3

### Zadanie 3.2. (0–3)

Niektóre hasła w pliku `hasla.txt` mają dodatkową regułę. Każdy znak po zamianie na odpowiedni kod ASCII jest liczbą pierwszą.

Wypisz wszystkie hasła spełniające tę regułę (każde w osobnej linii).

Przykładem hasła, które spełnia tę regułę, jest: `geag%5CkIqY`, które występuje w pliku `przyklad_hasla.txt`.

Litera	g	e	a	g	%	5	C	k	I	q	Y
Kod ASCII	103	101	97	103	37	53	67	107	73	113	89

3.3.

0–1  
2–3

### Zadanie 3.3. (0–3)

Podaj liczbę haseł w pliku `hasla.txt`, które nie zawierają żadnego znaku specjalnego, zawierają jeden znak specjalny, dwa, trzy, cztery itd. znaki specjalne (znaki te mogą się powtarzać). Dodatkowo wypisz wszystkie hasła, które zawierają dokładnie cztery znaki specjalne.

Wynik przedstaw w formacie:

*[liczba znaków specjalnych]: [liczba haseł w pliku hasla.txt]*

Przykładowa odpowiedź dla pliku `przyklad_hasla.txt`:

0: 13

1: 11

2: 1

`w80Tuw%1h$n`

**Do oceny oddajesz:**

- plik tekstowy `wyniki3.txt` – zawierający odpowiedź do poszczególnych zadań (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- plik (lub pliki) zawierający kod źródłowy (lub kody źródłowe) twojego programu o nazwie (brak tego pliku lub plików jest równoznaczny z brakiem rozwiązania zadania):

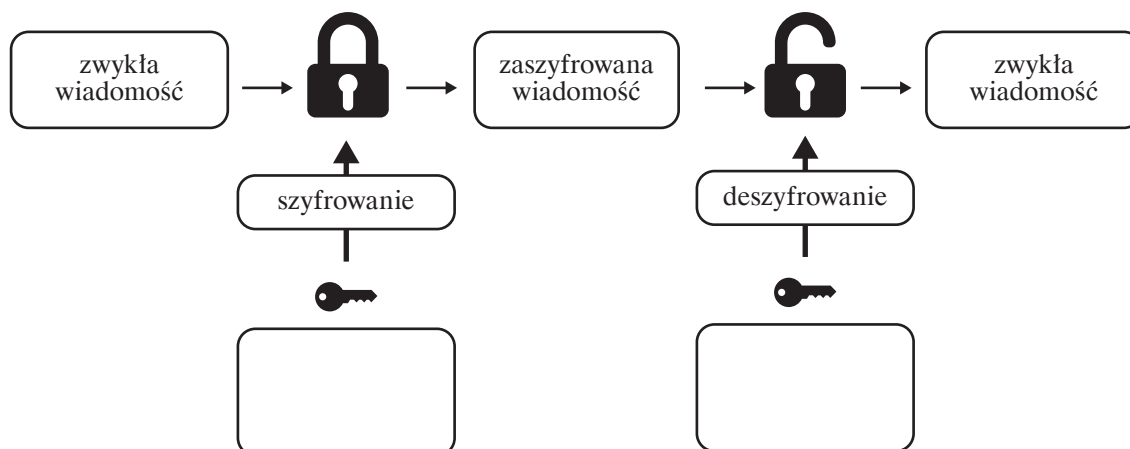
.....



### Zadanie 4. (0-1)

Na poniższym schemacie znajduje się algorytm szyfrowania RSA z dwoma pustymi polami tekstowymi.

**Uzupełnij puste pola wpisami: klucz prywatny i klucz publiczny odbiorcy wiadomości.**



### Zadanie 5. (0–2)

Poniżej sposobem pisemnym odjęto dwie liczby zapisane w systemie dwójkowym.

**Uzupełnij brakujące cyfry tak, aby działanie było wykonane poprawnie.**

$$\begin{array}{cccccccc}
 & 1 & 1 & \square & 0 & 1 & 1 & 0 & 1 \\
 - & & \square & 1 & 1 & 0 & \square & 1 & 1 \\
 \hline
 & 1 & 0 & \square & 0 & 1 & 1 & 0 & 
 \end{array}$$

*Miejsce na obliczenia (brudnopis)*

[illegible]

## Zadanie 6. Warownia

Gra Warownia przenosi gracza do czasów, gdy honor rycerza nie był tylko mitem, a od sprawności pozyskiwania surowców zależało zwycięstwo.

Gracz rozpoczął grę o godzinie 10:00, a skończył o 20:00. W grze co minutę do magazynu są transportowane trzy surowce: jedzenie, drewno i złoto. Pierwszy transport trafił do magazynu o godz. 10:01. Dzięki tym surowcom gracz mógł tworzyć obrońców tytułowej warowni: rycerzy i łuczników.

W pliku tekstowym `gra.txt` w kolejnych wierszach zapisano informacje o pozyskiwanych podczas gry surowcach: jedzeniu, drewnie i złocie, rozdzielone średnikiem.

### Przykład:

```
Jedzenie;Drewno;Zloto
40;28;10
74;14;1
```

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

6.1.

0–1–2

### Zadanie 6.1. (0–2)

Podaj najmniejszą ilość jedzenia jednorazowo przetransportowanego do magazynu oraz określ, ile razy odbył się taki transport. Podaj największą ilość jedzenia jednorazowo przetransportowanego do magazynu oraz określ, ile razy odbył się taki transport.

6.2.

0–1–2

### Zadanie 6.2. (0–2)

Znajdź najdłuższą serię kolejnych minut, w których ilość otrzymanego jedzenia była większa od ilości drewna, a ilość drewna była większa od ilości złota. Podaj długość serii i godzinę jej rozpoczęcia.

### Przykład:

10:03	11	26	8
10:04	74	29	3
10:05	39	15	3
10:06	36	26	5

Od godziny 10:03 w czterech kolejnych transportach ilość otrzymanego jedzenia była większa od ilości drewna, a ilość drewna większa od ilości złota.

### Zadanie 6.3. (0–3)

Wykonaj zestawienie z pierwszych trzech godzin gry. Dla każdej pełnej godziny oblicz sumę dostarczonego jedzenia, drewna i złota do magazynu warowni.

Na podstawie wykonanego zestawienia utwórz wykres kolumnowy. Pamiętaj o czytelnym opisie wykresu: dodaj opisy obu osi – „Godzina” dla osi X i „Liczba surowców” dla osi Y, tytuł oraz legendę zawierającą rodzaj surowca.

6.3.

0–1  
2–3

### Zadanie 6.4. (0–2)

Podczas gry okazało się, że warownia zarządzana przez gracza nie jest atakowana. Dlatego gracz chce przekazać część zgromadzonych surowców innemu członkowi swojej drużyny. Transport surowców odbywa się po każdej minucie, ale tylko wtedy, gdy w magazynie jest co najmniej 100 jednostek surowców (łącznie: jedzenia, drewna i złota). Wówczas wóz przewozi dokładnie 100 jednostek w kolejności: złoto, potem drewno, a na końcu jedzenie.

Podaj, ile razy transport się nie odbył oraz jaka była łączna ilość przekazanego jedzenia.

6.4.

0–1–2

**Przykład:**

O godzinie 10:20 nie było wystarczającej ilości surowców, by odbył się transport. W następnej minucie przekazano 14 jednostek złota, 46 jednostek drewna i 40 jednostek jedzenia.

### Zadanie 6.5. (0–3)

Zamiast przekazywania surowców (opisanego w zadaniu 6.4.) można rozegrać inny scenariusz gry: zatrudnić rycerzy i łuczników. Koszt rycerza to 200 jednostek jedzenia, 10 drewna i 40 złota, a koszt łucznika to 50 jednostek jedzenia, 40 drewna oraz 5 złota.

Jeżeli gracz ma odpowiednie ilości surowców, musi zatrudnić tylu rycerzy, na ilu go stać. Natomiast pozostałe środki musi przeznaczyć na zatrudnienie łuczników.

Podaj łączną liczbę zatrudnionych rycerzy i łuczników oraz liczbę minut, które upłynęły od godziny 10:00, w której po raz pierwszy zatrudniono dwóch łuczników.

6.5.

0–1  
2–3

**Do oceny oddajesz:**

- plik tekstowy wyniki6.txt – zawierający odpowiedzi do poszczególnych zadań (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- plik zawierający wykres do zadania 6.3. o nazwie: .....
- plik (lub pliki) zawierający kod źródłowy (lub kody źródłowe) twojego programu o nazwie (brak tego pliku lub plików jest równoznaczny z brakiem rozwiązania zadania):

.....

## Zadanie 7. Licytacje

Portal B2W to serwis internetowy umożliwiający sprzedaż i zakup różnych przedmiotów codziennego użytku. Zakup przedmiotów odbywa się w formie licytacji, w której wygrywa najwyższa oferta. Serwis działa od 1 sierpnia 2025 r.

W trzech plikach tekstowych o nazwach `uzytkownicy.txt`, `przedmioty.txt`, `oferty.txt` zapisano informacje zawierające dane o licytacjach na portalu B2W.

Plik o nazwie `uzytkownicy.txt` zawiera informacje o użytkownikach portalu B2W. Każdy wiersz tego pliku zawiera:

<code>uzytkownik_id</code>	– unikalny numer użytkownika portalu
<code>imie</code>	– imię użytkownika
<code>nazwisko</code>	– nazwisko użytkownika
<code>rola</code>	– rolę użytkownika: sprzedawca albo kupujący produkt

### Przykład:

```
uzytkownik_id;imie;nazwisko;rola
1;Sylwia;Stanczyk;kupujacy
2;Iwona;Chmielewska;kupujacy
3;Martyna;Stankiewicz;kupujacy
```

Plik o nazwie `przedmioty.txt` zawiera informacje o przedmiotach wystawionych na licytację. Każdy wiersz tego pliku zawiera:

<code>przedmiot_id</code>	– unikalny numer przedmiotu wystawionego na licytację
<code>nazwa</code>	– nazwę przedmiotu
<code>cena_wywolawcza</code>	– cenę wywoławczą przedmiotu
<code>wlasciciel_id</code>	– właściciela przedmiotu – użytkownika z rolą sprzedawca

### Przykład:

```
przedmiot_id;nazwa;cena_wywolawcza;wlasciciel_id
1;Laptop Dell Inspiron;2500;1
2;Smartfon Samsung Galaxy;1800;13
3;Imbryk do herbaty porcelanowy;120;24
```

Plik o nazwie `oferty.txt` zawiera informacje o ofertach złożonych przez kupujących na dany przedmiot, zarejestrowanych przez system B2W. Każdy wiersz tego pliku zawiera:

<code>oferta_id</code>	– numer oferty zakupu przedmiotu
<code>przedmiot_id</code>	– numer przedmiotu wystawionego na licytację
<code>uzytkownik_id</code>	– numer użytkownika portalu z rolą kupujący
<code>kwota</code>	– oferowaną kwotę za produkt
<code>data_oferty</code>	– datę wraz z godziną złożenia oferty zakupu przedmiotu

### Przykład:

```
oferta_id;przedmiot_id;uzytkownik_id;kwota;data_oferty
1;39;49;467;01.08.2025 12:05
2;40;32;167;01.08.2025 12:44
3;23;58;131;01.08.2025 12:44
```

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.

### Zadanie 7.1. (0–2)

Podaj imię i nazwisko kupującego, który złożył największą liczbę ofert zakupu przedmiotów. Podaj liczbę złożonych ofert.

7.1.

0–1–2

### Zadanie 7.2. (0–2)

Podaj nazwę trzech przedmiotów, dla których – w wyniku licytacji – różnica między największą kwotą wylicytowaną a ceną wywoławczą była największa.

7.2.

0–1–2

### Zadanie 7.3. (0–2)

Podaj imię i nazwisko każdego z użytkowników serwisu, który ma konto na portalu i rolę kupujący, ale nie licytował żadnego przedmiotu.

7.3.

0–1–2

### Zadanie 7.4. (0–2)

Podaj nazwę przedmiotu, który był najdłużej licytowany, oraz imię i nazwisko użytkownika, który wygrał tę licytację.

7.4.

0–1–2

Do oceny oddajesz:

- plik tekstowy `wyniki7.txt` – zawierający odpowiedzi do poszczególnych zadań (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- plik (lub pliki) zawierający kod źródłowy (kody źródłowe) twojego programu o nazwie (brak tego pliku lub plików jest równoznaczny z brakiem rozwiązania zadania):

.....

### Zadanie 7.5. (0–2)

Do tabel utworzonych na podstawie opisanych wcześniej plików dołączamy kolejną o nazwie `Grupy`, w której zapisano informacje o kategoriach przedmiotów, np.: elektronika, dom i ogród, sport.

7.5.

0–1–2

Tabela `Grupy` zawiera następujące pola:

`grupy_id` – unikatowy numer grupy  
`nazwa_grupy` – nazwa kategorii

Do tabeli `Przedmioty` dodano pole `grupy_id`.

Napisz w języku SQL zapytanie, w wyniku którego otrzymasz liczby przedmiotów w danej kategorii przedmiotów, posortowane malejąco.

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.

**BRUDNOPIS (*nie podlega ocenie*)**

ISBN 978-83-8197-613-8



9 788381 976138