Zadanie 3. Liczby

W pliku **liczby.txt** znajduje się 1000 różnych liczb naturalnych z przedziału [2, 1 000 000], każda w osobnym wierszu.

Pierwsze wiersze pliku liczby.txt:

```
826504
315496
958243
715395
257376
```

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku **wyniki3.txt**. Każdą odpowiedź poprzedź numerem oznaczającym zadanie.

Do Twojej dyspozycji jest plik **liczby_przyklad.txt** w identycznym formacie. Odpowiedzi dla danych z tego pliku podano pod treściami zadań – możesz na nich sprawdzać działanie swojego programu.

Uwaga: Pamiętaj, że Twój program musi ostatecznie działać dla pliku **liczby.txt**.

Zadanie 3.1. (0–2)

3.1.
0–1–2

Podaj, ile liczb z pliku **liczby.txt** jest liczbami pierwszymi, oraz największą z tych liczb pierwszych.

Postać odpowiedzi: Dwie liczby w jednym wierszu: liczba liczb pierwszych i największa z nich.

Dla pliku **liczby_przyklad.txt** poprawna odpowiedź to:

```
2 39791
```

[kod zadania: inf-1-3a]

Zadanie 3.2. (0–4)

3.2.
0–1–2–3–4

Dla każdej liczby z pliku wyznacz liczbę jej różnych dzielników pierwszych (np. 360 = 2 · 2 · 2 · 3 · 3 · 5 ma trzy różne dzielniki pierwsze: 2, 3 i 5). Podaj największą z wyznaczonych wartości, a po niej – wszystkie liczby z pliku, które mają tyle różnych dzielników pierwszych, w kolejności ich występowania w pliku.

Postać odpowiedzi: Jeden wiersz: największa liczba różnych dzielników pierwszych, a po niej – liczby z pliku, oddzielone spacjami.

Dla pliku **liczby_przyklad.txt** poprawna odpowiedź to:

```
6 84630 72930 46410
```

[kod zadania: inf-1-3b]

Zadanie 3.3. (0–4)

3.3.
0–1–2–3–4

Dwie liczby są względnie pierwsze, jeżeli ich największy wspólny dzielnik jest równy 1. Znajdź najdłuższy spójny fragment ciągu liczb zapisanych w pliku (ciąg liczb z kolejnych wierszy), w którym każde dwie sąsiednie liczby są względnie pierwsze. Podaj długość tego fragmentu (liczbę jego elementów) oraz jego pierwszą liczbę. Jest jeden taki najdłuższy fragment.

Postać odpowiedzi: Dwie liczby w jednym wierszu: długość fragmentu i jego pierwsza liczba.

Zadanie 3. Hasła

W pliku **hasla.txt** zapisano 800 haseł użytkowników pewnego serwisu, każde w osobnym wierszu. Hasła składają się z małych i wielkich liter alfabetu łacińskiego (a–z, A–Z), cyfr (0–9) oraz znaków specjalnych ze zbioru **! @ # $ % & *** i nie zawierają odstępów. Wielkość liter ma znaczenie (np. Kot1 i kot1 to dwa różne hasła). W pliku **slownik.txt** zapisano 20 popularnych słów złożonych z małych liter a–z, każde w osobnym wierszu.

Pierwsze wiersze pliku hasla.txt:

```
Kasztan681
kawA75
3DLacq&1BD0goH#g
admin8081
cNJIlVeQ4Tb
```

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku **wyniki3.txt**. Każdą odpowiedź poprzedź numerem oznaczającym zadanie.

Do Twojej dyspozycji są pliki **hasla_przyklad.txt** i **slownik_przyklad.txt** w identycznym formacie. Odpowiedzi dla danych z tego pliku podano pod treściami zadań – możesz na nich sprawdzać działanie swojego programu.

Uwaga: Pamiętaj, że Twój program musi ostatecznie działać dla pliku **hasla.txt** (i **slownik.txt**).

Zadanie 3.1. (0–2)

3.1.
0–1–2

Hasło jest *silne*, jeżeli ma co najmniej 12 znaków i zawiera co najmniej jedną małą literę, co najmniej jedną wielką literę, co najmniej jedną cyfrę i co najmniej jeden znak specjalny. Podaj liczbę silnych haseł w pliku **hasla.txt** (hasło występujące w pliku kilka razy liczymy tyle razy, ile razy występuje).

Postać odpowiedzi: Jedna liczba całkowita.

Przykład: hasło **Ab3$efgh1234** jest silne, a hasło **abcd$efgh1234** nie jest silne (nie zawiera wielkiej litery).

Dla pliku **hasla_przyklad.txt i slownik_przyklad.txt** poprawna odpowiedź to:

```
9
```

[kod zadania: inf-2-3a]

Zadanie 3.2. (0–4)

3.2.
0–1–2–3–4

Niektóre hasła występują w pliku więcej niż jeden raz. Podaj, ile jest **różnych** haseł, które występują w pliku więcej niż jeden raz, oraz hasło występujące w pliku najwięcej razy wraz z liczbą jego wystąpień. Jest jedno takie hasło.

Postać odpowiedzi: Jeden wiersz: liczba różnych powtarzających się haseł, najczęstsze hasło, liczba jego wystąpień.

Dla pliku **hasla_przyklad.txt i slownik_przyklad.txt** poprawna odpowiedź to:

```
2 Hw#@eQu1CbBtEAe 5
```

[kod zadania: inf-2-3b]

Zadanie 3. Wyrażenia w ONP

W pliku **onp.txt** zapisano 500 wyrażeń w odwrotnej notacji polskiej, każde w osobnym wierszu. Tokeny – liczby całkowite od 1 do 20 oraz operatory **+**, **-** (odejmowanie) i ***** (mnożenie) – są oddzielone pojedynczymi spacjami. Wiersze numerujemy od 1.

Wyrażenie arytmetyczne w odwrotnej notacji polskiej (ONP) zapisujemy tak, że operator stoi po swoich dwóch argumentach, np. wyrażenie (3 + 4) · 2 ma w ONP postać **3 4 + 2 ***. Wartość wyrażenia w ONP oblicza się za pomocą stosu: tokeny czytamy od lewej do prawej; liczbę kładziemy na stos, a po napotkaniu operatora zdejmujemy ze stosu dwie liczby – najpierw *b*, potem *a* – i kładziemy na stos wynik działania *a* op *b* (np. dla **9 4 -** wynik to 5). Wyrażenie jest *poprawne*, jeżeli przy każdym operatorze na stosie są co najmniej dwie liczby, a po przeczytaniu wszystkich tokenów na stosie zostaje dokładnie jedna liczba – wartość wyrażenia. Na przykład wyrażenia **3 +** oraz **1 2** nie są poprawne.

token	3	4	+	2	*
stos po tokenie	3	3 4	7	7 2	14

Pierwsze wiersze pliku onp.txt:

```
6 10 8 + * 4 9 - * 12 - +
19 15 20 12 * - +
18 11 2 + 8 + -
12 1 - 13 - 10 7 + - 11 -
1
```

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku **wyniki3.txt**. Każdą odpowiedź poprzedź numerem oznaczającym zadanie.

Do Twojej dyspozycji jest plik **onp_przyklad.txt** w identycznym formacie. Odpowiedzi dla danych z tego pliku podano pod treściami zadań – możesz na nich sprawdzać działanie swojego programu.

Uwaga: Pamiętaj, że Twój program musi ostatecznie działać dla pliku **onp.txt**.

Zadanie 3.1. (0–3)

3.1.
0–1–2–3

Podaj liczbę poprawnych wyrażeń w pliku **onp.txt**.

Postać odpowiedzi: Jedna liczba całkowita.

Dla pliku **onp_przyklad.txt** poprawna odpowiedź to:

```
11
```

[kod zadania: inf-3-3a]

Zadanie 3.2. (0–3)

3.2.
0–1–2–3

Dla każdego poprawnego wyrażenia wyznacz największą liczbę elementów, które jednocześnie znajdowały się na stosie podczas obliczania jego wartości (np. dla wyrażenia 3 4 + 2 * jest to 2, a dla 1 2 3 * + jest to 3). Podaj największą z wyznaczonych liczb oraz liczbę poprawnych wyrażeń, dla których ją osiągnięto.

Postać odpowiedzi: Dwie liczby w jednym wierszu.

Zadanie 4. Łańcuchy DNA

W pliku **dna.txt** zapisano 300 par łańcuchów DNA, każdą parę w osobnym wierszu; łańcuchy w parze są oddzielone spacją. Łańcuch jest napisem złożonym z liter A, C, G, T, o długości od 1 do 50. Wiersze numerujemy od 1.

Pierwsze wiersze pliku dna.txt:

```
CTCGCGGCGAAAGCGGCCCAAGTGATAGAG TTATCAACGAGATCCTATAAGACCCCAGTC
TGACCTGGATGAGGGGA GCGTGAGGCTCATCAGGG
TCCTCGTGCG CGAGTATTCT
GACCCAATA TCCTCTCGTGTCGTACTG
TATGAGATGTAGGGTTACG TGTCCCACGGAGGCGGTGACG
```

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku **wyniki4.txt**. Każdą odpowiedź poprzedź numerem oznaczającym zadanie.

Do Twojej dyspozycji jest plik **dna_przyklad.txt** w identycznym formacie. Odpowiedzi dla danych z tego pliku podano pod treściami zadań – możesz na nich sprawdzać działanie swojego programu.

Uwaga: Pamiętaj, że Twój program musi ostatecznie działać dla pliku **dna.txt**.

Zadanie 4.1. (0–2)

4.1.
0–1–2

Podaj liczbę wierszy, w których pierwszy łańcuch występuje w drugim jako spójny fragment (np. ACG występuje w TTACGA; łańcuch jest też fragmentem samego siebie).

Postać odpowiedzi: Jedna liczba całkowita.

Dla pliku **dna_przyklad.txt** poprawna odpowiedź to:

```
1
```

[kod zadania: inf-4-4a]

Zadanie 4.2. (0–3)

4.2.
0–1–2–3

Łańcuch komplementarny odwrócony powstaje z danego łańcucha przez zapisanie go od końca i zastąpienie każdej litery literą komplementarną: A ↔ T, C ↔ G (np. dla AACG jest to CGTT). Podaj liczbę wierszy, w których drugi łańcuch jest łańcuchem komplementarnym odwróconym pierwszego łańcucha.

Postać odpowiedzi: Jedna liczba całkowita.

Dla pliku **dna_przyklad.txt** poprawna odpowiedź to:

```
1
```

[kod zadania: inf-4-4b]

Zadanie 4.3. (0–4)

4.3.
0–1–2–3–4

Podciąg łańcucha powstaje przez usunięcie z niego dowolnych liter (być może żadnej) bez zmiany kolejności pozostałych liter, np. ACT jest podciągiem łańcucha AGCGT. Dla każdego wiersza wyznacz długość najdłuższego wspólnego podciągu obu łańcuchów. Podaj największą z tych długości oraz numer wiersza, w którym ją osiągnięto. Jest jeden taki wiersz.

Postać odpowiedzi: Dwie liczby w jednym wierszu: długość i numer wiersza.

Zadanie 7. Rower miejski

W pliku **rowery.txt** zapisano dane o 2400 wypożyczeniach rowerów miejskich w okresie od 1 kwietnia do 30 września 2026 r. Pierwszy wiersz pliku jest wierszem nagłówkowym, a dane w wierszach rozdzielono znakami tabulacji. W każdym wierszu zapisano:

- ***IdKlienta*** – numer klienta (liczba całkowita)
- ***Data*** – data wypożyczenia (rrrr-mm-dd)
- ***Godzina*** – godzina wypożyczenia (gg:mm)
- ***StacjaStart*** – nazwa stacji, na której wypożyczono rower
- ***StacjaKoniec*** – nazwa stacji, na której zwrócono rower
- ***CzasMin*** – czas wypożyczenia w minutach (liczba całkowita)

Przykład:

IdKlienta	Data	Godzina	StacjaStart	StacjaKoniec	CzasMin
34	2026-04-01	08:07	Rynek	Uniwersytet	26
73	2026-04-01	09:16	Dworzec	Stadion	12
73	2026-04-01	15:09	Park	Dworzec	24
73	2026-04-01	17:24	Dworzec	Dworzec	133

Z wykorzystaniem danych zawartych w pliku oraz dostępnych narzędzi informatycznych wykonaj podane zadania. Odpowiedzi zapisz w pliku **wyniki7.txt**. Odpowiedź do każdego zadania poprzedź numerem tego zadania.

Zadanie 7.1. (0–2)

7.1.
0–1–2

Podaj nazwę stacji, na której rozpoczęło się najwięcej wypożyczeń, i nazwę stacji, na której zakończyło się najwięcej wypożyczeń, wraz z odpowiednimi liczbami wypożyczeń.

Zadanie 7.2. (0–3)

7.2.
0–1–2–3

Wykonaj zestawienie liczby wypożyczeń w poszczególnych miesiącach. Na podstawie zestawienia utwórz wykres kolumnowy. Pamiętaj o czytelnym opisie wykresu: tytuł wykresu, opisy osi, nazwy miesięcy na osi X.

Zadanie 7.3. (0–2)

7.3.
0–1–2

Opłata za wypożyczenie jest naliczana według zasady: pierwsze 20 minut jest bezpłatne, a za każde rozpoczęte kolejne 15 minut płaci się 2 zł. Na przykład wypożyczenie trwające 20 minut jest bezpłatne, 21 minut i 35 minut – kosztuje 2 zł, 36 minut – 4 zł. Podaj łączną kwotę opłat za wszystkie wypożyczenia oraz liczbę wypożyczeń bezpłatnych.

Zadanie 7.4. (0–3)

7.4.
0–1–2–3

Operator rozważa wprowadzenie abonamentu miesięcznego w cenie 25 zł, z którym wszystkie wypożyczenia w danym miesiącu kalendarzowym są bezpłatne. Dla każdego miesiąca podaj liczbę klientów, dla których suma opłat za wypożyczenia w tym miesiącu (naliczonych jak w zadaniu 7.3.) była **większa niż** 25 zł. Podaj też łączną kwotę, jaką zaoszczędziliby ci klienci w całym okresie,

Zadanie 4. Obraz

Plik **obraz.txt** zawiera obraz w 256 odcieniach szarości o wymiarach 100 wierszy na 120 kolumn. W każdym wierszu pliku zapisano jasności kolejnych pikseli jednego wiersza obrazu – liczby całkowite od 0 do 255 oddzielone pojedynczymi spacjami. Wiersze i kolumny obrazu numerujemy od 1.

Pierwsze wiersze pliku obraz.txt:

```
34 34 34 34 193 193 193 193 240 240 240 240 33 33 141 39 39 39 39 39 39 91 91 150 32 32 32 32 1
112 147 147 147 223 126 126 126 126 126 126 68 68 68 68 54 54 54 54 54 54 115 115 112 112 164 1
1 1 1 1 254 246 246 21 42 42 74 74 74 76 231 231 82 47 47 47 47 47 47 172 172 172 172 172 172 2
237 237 237 237 237 237 69 103 103 22 22 22 22 107 167 167 123 123 123 123 136 136 136 136 137
157 157 157 154 154 154 221 221 221 155 155 155 118 118 118 129 64 64 4 183 183 132 132 132 13
```

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku **wyniki4.txt**. Każdą odpowiedź poprzedź numerem oznaczającym zadanie.

Do Twojej dyspozycji jest plik **obraz_przyklad.txt** w identycznym formacie. Odpowiedzi dla danych z tego pliku podano pod treściami zadań – możesz na nich sprawdzać działanie swojego programu.

Uwaga: Pamiętaj, że Twój program musi ostatecznie działać dla pliku **obraz.txt**.

Zadanie 4.1. (0–2)

4.1.
0–1–2

Podaj numer kolumny obrazu, w której suma jasności pikseli jest największa, oraz tę sumę. Jest jedna taka kolumna.

Postać odpowiedzi: Dwie liczby w jednym wierszu: numer kolumny i suma jasności.

Dla pliku **obraz_przyklad.txt** poprawna odpowiedź to:

```
9 1375
```

[kod zadania: inf-2-4a]

Zadanie 4.2. (0–4)

4.2.
0–1–2–3–4

Obraz kompresujemy bezstratnie metodą RLE, osobno dla każdego wiersza: każdy maksymalny ciąg kolejnych pikseli wiersza o tej samej jasności zapisujemy jako jedną parę (jasność, długość ciągu). Na przykład wiersz 7 7 7 0 0 7 zapisujemy za pomocą trzech par: (7, 3) (0, 2) (7, 1). Podaj łączną liczbę par w zapisie całego obrazu oraz numer wiersza, którego zapis wymaga najmniejszej liczby par (jeżeli jest kilka takich wierszy – podaj najmniejszy numer).

Postać odpowiedzi: Dwie liczby w jednym wierszu: łączna liczba par i numer wiersza.

Dla pliku **obraz_przyklad.txt** poprawna odpowiedź to:

```
38 1
```

[kod zadania: inf-2-4b]

Zadanie 4.3. (0–4)

4.3.
0–1–2–3–4

Znajdź największy kwadratowy fragment obrazu, w którym wszystkie piksele mają tę samą jasność. Podaj długość boku tego kwadratu (w pikselach), jasność jego pikseli oraz numer wiersza i numer kolumny jego lewego górnego piksela. Jest jeden taki kwadrat.

Postać odpowiedzi: Cztery liczby w jednym wierszu: bok, jasność, numer wiersza, numer kolumny.

Zadanie 4. Kolejka w urzędzie

Urząd otwiera się o godzinie 8:00. W pliku **klienci.txt** zapisano dane 600 klientów w kolejności ich przychodzenia. W każdym wierszu są dwie liczby całkowite oddzielone spacją: moment przyjścia klienta (liczba sekund, które upłynęły od otwarcia urzędu) oraz czas jego obsługi w sekundach. Momenty przyjścia tworzą ciąg niemalejący. Klientów numerujemy od 1 (numer klienta = numer wiersza). Klienci są obsługiwani w kolejności przyjścia. Jeżeli okienko jest wolne, obsługa klienta zaczyna się w chwili jego przyjścia; w przeciwnym razie klient czeka, aż okienko się zwolni (obsługa może zacząć się dokładnie w chwili zakończenia obsługi poprzedniego klienta).

Przykład: klienci przychodzą w chwilach 0, 10 i 100, a ich obsługa trwa odpowiednio 50, 30 i 20 sekund. Przy jednym okienku klient 1 jest obsługiwany od 0 do 50, klient 2 – od 50 do 80 (czekał 40 sekund), a klient 3 – od 100 do 120 (nie czekał).

Pierwsze wiersze pliku klienci.txt:

```
127 94
168 64
289 69
348 47
362 20
```

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku **wyniki4.txt**. Każdą odpowiedź poprzedź numerem oznaczającym zadanie.

Do Twojej dyspozycji jest plik **klienci_przyklad.txt** w identycznym formacie. Odpowiedzi dla danych z tego pliku podano pod treściami zadań – możesz na nich sprawdzać działanie swojego programu.

Uwaga: Pamiętaj, że Twój program musi ostatecznie działać dla pliku **klienci.txt**.

Zadanie 4.1. (0–2)

4.1.
0–1–2

Urząd ma jedno okienko. Podaj moment (liczbę sekund od otwarcia urzędu), w którym zakończy się obsługa ostatniego klienta.

Postać odpowiedzi: Jedna liczba całkowita.

Dla pliku **klienci_przyklad.txt** poprawna odpowiedź to:

```
513
```

[kod zadania: inf-3-4a]

Zadanie 4.2. (0–3)

4.2.
0–1–2–3

Urząd ma jedno okienko. Czas oczekiwania klienta to różnica między momentem rozpoczęcia jego obsługi a momentem jego przyjścia. Podaj najdłuższy czas oczekiwania oraz numer klienta, który czekał najdłużej (jeżeli jest kilku takich klientów – podaj najmniejszy numer).

Postać odpowiedzi: Dwie liczby w jednym wierszu: czas oczekiwania i numer klienta.

Dla pliku **klienci_przyklad.txt** poprawna odpowiedź to:

```
226 8
```

[kod zadania: inf-3-4b]

Zadanie 7. Klub biegacza

W pliku **treningi.txt** zapisano 1676 treningów członków klubu biegacza z okresu od 1 stycznia do 30 czerwca 2026 r. Każdy biegacz trenuje co najwyżej raz dziennie. Pierwszy wiersz pliku jest wierszem nagłówkowym, a dane w wierszach rozdzielono znakami tabulacji. W każdym wierszu zapisano:

- ***Data*** – data treningu (rrrr-mm-dd)
- ***Biegacz*** – pseudonim biegacza
- ***Dystans*** – dystans w km (z dwoma miejscami po przecinku)
- ***Czas*** – czas biegu w formacie g:mm:ss

Przykład:

Data	Biegacz	Dystans	Czas
2026-01-01	bartek77	10,97	1:04:03
2026-01-01	ela_m	4,74	0:23:13
2026-01-01	hubi	7,86	0:45:29
2026-01-01	iga_s	5,12	0:27:31

Z wykorzystaniem danych zawartych w pliku oraz dostępnych narzędzi informatycznych wykonaj podane zadania. Odpowiedzi zapisz w pliku **wyniki7.txt**. Odpowiedź do każdego zadania poprzedź numerem tego zadania.

Zadanie 7.1. (0–2)

7.1.
0–1–2

Podaj pseudonimy trzech biegaczy, którzy w całym okresie przebiegli łącznie najdłuższy dystans (od największego), wraz z tymi dystansami.

Zadanie 7.2. (0–3)

7.2.
0–1–2–3

Wykonaj zestawienie liczby treningów o dystansie (w km) z przedziałów: [0; 5), [5; 10), [10; 15), [15; 21,1), 21,1 i więcej. Na podstawie zestawienia utwórz wykres kolumnowy. Pamiętaj o czytelnym opisie wykresu: tytuł, opisy osi, przedziały na osi X.

Zadanie 7.3. (0–2)

7.3.
0–1–2

Tempo biegu to czas biegu podzielony przez dystans (w sekundach na kilometr). Spośród treningów o dystansie co najmniej 10 km wybierz trening o najlepszym (najmniejszym) tempie. Podaj pseudonim biegacza, datę treningu oraz tempo w sekundach na kilometr zaokrąglone do liczby całkowitej. *Przykład:* bieg na dystansie 10,50 km w czasie 0:52:30 (3150 s) ma tempo 300 s/km.

Zadanie 7.4. (0–3)

7.4.
0–1–2–3

Serią nazywamy ciąg kolejnych dni kalendarzowych, w każdym z których biegacz odbył trening. Podaj pseudonim biegacza, który miał najdłuższą serię, długość tej serii oraz datę jej pierwszego dnia. Jest jeden taki biegacz. *Przykład:* biegacz, który trenował 3, 4, 5 i 7 marca, ma serię 3-dniową (od 3 marca) i serię 1-dniową (7 marca).

CO JESZCZE JEST W ŚRODKU

Matura 2027 z informatyki, poziom rozszerzony – pakiet 4 arkuszy próbnych

Podgląd pokazał 8 z 127 stron. Pełna wersja: 127 stron A4, PDF do druku i DOCX do edycji.

- Matura 2027 z informatyki – poziom rozszerzony. Arkusz próbny 4 z kluczem
- Matura 2027 z informatyki – poziom rozszerzony. Arkusz próbny 1 z kluczem
- Matura 2027 z informatyki – poziom rozszerzony. Arkusz próbny 2 z kluczem
- Matura 2027 z informatyki – poziom rozszerzony. Arkusz próbny 3 z kluczem

Kup pełną wersję – plik od razu na e-mail
pierwszalawka.pl/material/matura-informatyka-pr-2027-pakiet-4-arkuszy