



Instrukcja dla ucznia

1. Sprawdź, czy arkusz konkursowy z treścią zadań zawiera 7 stron. Ewentualny brak stron lub inne usterki zgłoś Komisji.
2. Sprawdź, czy w folderze konkursowym znajdują się pliki: *zapalki.in*, *zapalki.out*, *wynik.out*, *wynik100.out*, *dane.in*, *dane100.in*,
3. Czytaj uważnie treść zadań.
4. Rozwiązania zadań zapisuj regularnie w folderze konkursowym na pulpicie komputera.
5. Pamiętaj o poprawnym nazywaniu plików z rozwiązaniami zadań (zgodnie z instrukcją w treści zadań).
6. Rozwiązując zadania zadbaj o uniwersalność rozwiązań tak, aby działały prawidłowo **na innych zestawach danych wejściowych**.
7. Rozwiązując zadania, możesz tworzyć różne pliki traktując je, jako brudnopis. Zapisy w brudnopisach nie będą sprawdzane i oceniane.
8. Po zakończeniu pracy z zadaniami, zgłoś ten fakt Komisji poprzez podniesienie ręki.
9. W obecności przedstawiciela Komisji upewnij się że wgrałeś na platformę konkursową pliki z rozwiązaniami oraz plik *metryczka.xls*

Etap Wojewódzki

2023 r.

Czas pracy:

120 minut

**Liczba punktów do
uzyskania:**

60 punktów

P O W O D Z E N I A



ZADANIE 1. (0 – 40)

Plik z przykładowymi danymi wejściowymi: **dane.in**

Plik z danymi wyjściowymi: **wyniki.out**

*Dodatkowy plik do testowania rozwiązań z większą liczbą danych: **dane100.in***

Do oceny należy oddać plik zawierający w nazwie *Kod Ucznia*

(np. K01_zad1.cpp lub K01_zad1.c)

Opis organizacji danych w pliku wejściowym

W pierwszym wierszu pliku **dane.in** zapisana jest jedna liczba naturalna **n** oznaczająca ilość liczb znajdujących się w kolejnym wierszu pliku. Liczba **n** mieści się w przedziale (obustronnie) domkniętym od **1** do **1 000 000**. W kolejnym wierszu zapisanych jest **n** liczb z zakresu od **1** do **1 000 000** oddzielonych spacją. Liczby te stanowią dane do rozwiązania poniższych zadań.

Przykład organizacji danych w pliku *dane.in*:

```
5
3 4 2 1 5
```

Korzystając z powyższych informacji napisz program, który wykona czynności wymienione w punktach od A do E (przy każdym z nich podano maksymalną punktację możliwą do uzyskania):

A. (0 - 7)

Wczytaj liczby z pliku wejściowego do odpowiednich struktur danych oraz umieść w **kodzie źródłowym w postaci komentarzy** nazwy zmiennych wraz z typami danych jakie zostały użyte do zapamiętania powyższych informacji. Wyświetl na ekranie i zapisz do pliku wyjściowego dwie liczby oddzielone spacją oznaczające odpowiednio: liczbę najmniejszą i liczbę największą spośród wczytanych liczb.

Przykład

Dane wejściowe:

```
5
3 4 2 1 5
```

Dane wyjściowe:

```
1 5
```



B. (0 - 4)

Dla wczytanego zestawu danych policz, ile z nich jest podzielnych przez 2 a ile przez 7.

Wynik wypisz na ekranie i zapisz do pliku wyjściowego w postaci: pierwsza liczba oznacza ilość wystąpień liczb podzielnych przez dzielnik, który jako druga liczba oddzielony jest od niej dwukropkiem. Wyniki oddziel spacją, zachowując kolejność niemalejącą względem ilości wystąpień.

Przykład

Dane wejściowe:

10

2 1 3 1 8 16 4 14 21 28 21

Dane wyjściowe:

4:7 6:2

C. (0 - 7)

Dla wczytanego zestawu danych, policz ile jest liczb „doskonałych”.

„Liczba doskonała to taka **liczba naturalna, która jest równa sumie wszystkich swoich dzielników właściwych** (tzn. mniejszych od niej). Najmniejszą liczbą doskonałą jest $6 = 1 + 2 + 3$, następną $28 = 1+2+4+7+14$, a kolejne to 496, 8128, 33550336. Dotychczas znaleziono 50 liczb doskonałych – wszystkie parzyste” (za wikipedia.pl)

Wynik, jako ilość liczb spełniających kryterium liczby doskonałej, wyświetl na ekranie i zapisz do pliku wyjściowego.

Przykład 1

Dane wejściowe:

5

31 6 2 496 6

Dane wyjściowe:

3

Przykład 2

Dane wejściowe:

4

21 11 31 10

Dane wyjściowe:

0



D. (0 - 9)

Dla wczytanego zestawu danych, policz ile jest liczb „palindromów”.

Palindrom to liczba lub wyraz, który czytany z lewej do prawej i z prawej do lewej brzmi tak samo. Palindromami są na przykład wyrazy: kok, kajak itp. Istnieją też zdania będące palindromami, np.: „Kobyła ma mały bok”. Nas jednak będą interesowały palindromy liczbowe. Oto kilka przykładów: 1, 101, 11111, 123321.

Wynik, jako ilość liczb spełniających kryterium palindromu liczbowego, wyświetl na ekranie i zapisz do pliku wyjściowego.

Przykład 1

Dane wejściowe:

5

31 6 2 496 6

Dane wyjściowe:

3

Przykład 2

Dane wejściowe:

4

21 11 313 10

Dane wyjściowe:

2

E. (0 - 13)

Potraktuj dane z wejścia jako wartości kolejnych elementów tablicy. Szukamy w niej takiego miejsca, które dzieli tablicę w taki sposób, że różnica pomiędzy sumą liczb w pierwszej części a sumą liczb w drugiej części jest jak najbliższa zeru. W jednej z części musi znajdować się co najmniej jedna liczba. Wyjście powinno zawierać jedną liczbę naturalną równą minimalnej wartości bezwzględnej z różnicy pomiędzy jedną częścią a drugą.

Przy ocenie brana będzie pod uwagę optymalność Twojego rozwiązania. Postaraj się, aby jego złożoność obliczeniowa była liniowa.

Przykład 1

Dane wejściowe:

5

3 4 2 1 5

Dane wyjściowe:

1

Wyjaśnienie dla przykładu 1: dzieląc tablicę na dwie części zawierające liczby: 3, 4 oraz 2, 1, 5, otrzymamy różnicę ich sum równą $7-8=-1$. Wartość bezwzględna z liczby -1 to 1.

Przykład 2

Dane wejściowe:

5

10 1 6 1 2

Dane wyjściowe:

0

Wyjaśnienie dla przykładu 2: dzieląc tablicę na dwie części zawierające liczby: 10 oraz 1, 6, 1, 2 otrzymamy różnicę ich sum równą $10-10=0$.

ZADANIE 2. (0 – 20)

Plik z przykładowymi danymi wejściowymi: **zapalki.in**

Plik z danymi wyjściowymi: **zapalki.out**

Do oceny należy oddać plik zawierający w nazwie *Kod Ucznia*

(np. K01_zad2.cpp lub K01_zad2.c)

Opis organizacji danych w pliku wejściowym

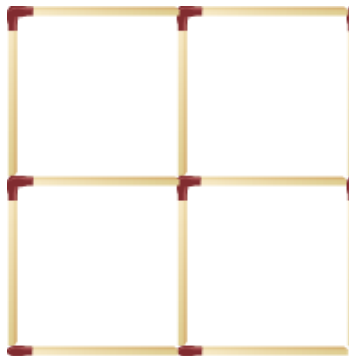
W pierwszym wierszu pliku **zapalki.in** zapisana jest jedna liczba naturalna **n** oznaczająca ilość liczb zapisanych w kolejnym wierszu pliku. Liczba **n** mieści się w przedziale (obustronnie) domkniętym od **1** do **1 000**. W kolejnym wierszu zapisanych jest **n** liczb całkowitych z zakresu od **1** do **1 000 000** oddzielonych spacją. Liczby te stanowią dane do rozwiązania poniższych zadań i oraz określają liczbę kwadratów, które chcemy ułożyć z zapalek.

Przykład organizacji danych w pliku *zapalki.in*:

4

1 4 16 5

Twoim zadaniem jest obliczyć minimalną liczbę zapalek potrzebną do tego, aby na płaszczyźnie utworzyć figurę zbudowaną z **N** kwadratów o bokach równych długości pojedynczej zapalki. Zapalki nie mogą być w żaden sposób dzielone i powinny stykać się końcami w wierzchołkach kwadratów, które tworzą. Poprawna konstrukcja z zapalek złożona z czterech kwadratów wygląda tak:



Rysunek pochodzi z serwisu: szaloniczby.pl



Napisz program, który wczyta zestaw danych z pliku `zapalki.in` i dla kolejnej danej z wejścia wyznaczy minimalną liczbę zapalek, niezbędnych do utworzenia na płaszczyźnie figury zbudowanej z kwadratów w ilości zgodnej z wczytaną wartością. Dla kolejnych wczytanych danych wypisz wynik na ekranie i zapisz do pliku wyjściowego o nazwie `zapalki.out` oddzielając je spacją.

Przy ocenie brana będzie pod uwagę optymalność Twojego rozwiązania. Postaraj się, aby działało ono w czasie stałym.

Przykład

Dane wejściowe:

4

1 4 16 5

Dane wyjściowe:

4 12 40 15

UWAGI DO ROZWIĄZAŃ

- Dane zawarte w przykładowych plikach wejściowych *dane.in*, *dane100.in*, *zapalki.in*, są poprawne (nie wymagają sprawdzania poprawności).
- W rozwiązaniach należy zadbać o odpowiednie dobranie struktur i typów danych do przechowywanych zawartości, poprawną inicjalizację zmiennych, czytelność i przejrzystość kodu oraz **zastosowanie algorytmu o jak najmniejszej złożoności czasowej**.
- Rozwiązania będą testowane i oceniane **na innych danych niż załączone do zadania** (zgodnych ze specyfikacją zadania).
- Do oceny należy przesłać tylko plik źródłowy (z rozszerzeniem **cpp** lub **c**) realizujący rozwiązanie zadania 1 (z punktów od A do E) oraz zadania 2. Nie należy załączać pliku wykonywalnego **exe**.
- Program powinien składać się tylko z jednego pliku i korzystać jedynie ze standardowych bibliotek (*iostream*, *fstream*, *cstdlib*, *stdio.h*, *cmath*, *algorithm*).
- Dane do programu powinny być pobierane z pliku wejściowego. Można to wykonać poprzez **wczytanie danych z pliku wejściowego** lub poprzez **przekierowanie standardowego wejścia**.
- Rozwiązania nie mogą tworzyć plików tymczasowych, procesów czy wątków, korzystać z funkcji sieciowych, oczekiwać na interakcję użytkownika, uruchamiać innych programów, korzystać z zewnętrznych bibliotek (oprócz wymienionych powyżej).
- Przed rozwiązaniami punktów zadania należy wyświetlić na ekranie i zapisać w pliku wyjściowym odpowiednią literę określającą poszczególne części zadania.



Przykład informacji, jaka powinna zostać zapisana w pliku wyjściowym oraz pojawić się na ekranie po uruchomieniu programu:

Dla pliku wejściowego *dane100.in* o zawartości jak w opisie zadania, program powinien wyświetlić na ekranie oraz zapisać w pliku wyjściowym *wynik100.out* następujące informacje:

A
6 32593

B
18:7 60:2

C
11

D
4

E
13492

Dla pliku wejściowego *zapalki.in* o zawartości jak w opisie zadania, program powinien wyświetlić na ekranie oraz zapisać w pliku wyjściowym *zapalki.out* następujące informacje:

4 7 10 12 15 17 20 22 24 27 29 31 34 36 38 40 43 45 47 262
