INFORMATOR
o **egzaminie maturalnym** z **informatyki**

od roku szkolnego 2022/2023

Centralna Komisja
Egzaminacyjna Warszawa 2021

Zespół redakcyjny:

Iwona Arcimowicz (CKE)
prof. dr hab. Krzysztof Diks (Uniwersytet Warszawski)
dr Janusz.Jabłonowski (Uniwersytet Warszawski)
Agata Kordas Łata
prof. dr hab. Krzysztof Loryś (Uniwersytet Wrocławski)
dr Lech Duraj (Uniwersytet Jagielloński)
dr Anna Kwiatkowska (Uniwersytet Mikołaja Kopernika w Toruniu)
Romuald Rostecki (OKE Gdańsk)
dr Wioletta Kozak (CKE)
dr Marcin Smolik (CKE)

Recenzenci:

dr Marcin Engel
dr Piotr Chrzastowski-Wachtel
Joanna Śmigielska
dr Tomasz Karpowicz (recenzja językowa)

Informator został opracowany przez Centralną Komisję Egzaminacyjną we współpracy z okręgowymi komisjami egzaminacyjnymi.

Centralna Komisja Egzaminacyjna
ul. Józefa Lewartowskiego 6, 00-190 Warszawa
tel. 22 536 65 00
sekretariat@cke.gov.pl

Okręgowa Komisja Egzaminacyjna w Gdańsku
ul. Na Stoku 49, 80-874 Gdańsk
tel. 58 320 55 90
komisja@oke.gda.pl

Okręgowa Komisja Egzaminacyjna w Jaworznie
ul. Adama Mickiewicza 4, 43-600 Jaworzno
tel. 32 616 33 99
oke@oke.jaworzno.pl

Okręgowa Komisja Egzaminacyjna w Krakowie
os. Szkolne 37, 31-978 Kraków
tel. 12 683 21 99
oke@oke.krakow.pl

Okręgowa Komisja Egzaminacyjna w Łomży
al. Legionów 9, 18-400 Łomża
tel. 86 473 71 20
sekretariat@oke.lomza.pl

Okręgowa Komisja Egzaminacyjna w Łodzi
ul. Ksawerego Praussa 4, 94-203 Łódź
tel. 42 634 91 33
sekretariat@lodz.oke.gov.pl

Okręgowa Komisja Egzaminacyjna w Poznaniu
ul. Gronowa 22, 61-655 Poznań
tel. 61 854 01 60
sekretariat@oke.poznan.pl

Okręgowa Komisja Egzaminacyjna w Warszawie
pl. Europejski 3, 00-844 Warszawa
tel. 22 457 03 35
info@oke.waw.pl

Okręgowa Komisja Egzaminacyjna we Wrocławiu
ul. Tadeusza Zielińskiego 57, 53-533 Wrocław
tel. 71 785 18 94
sekretariat@oke.wroc.pl

Spis treści

1. Opis egzaminu maturalnego z informatyki **5**
Wstęp **5**
Zadania na egzaminie **5**
Opis arkusza egzaminacyjnego **6**
Zasady oceniania **6**

2. Przykładowe zadania z rozwiązaniami **9**

3. Informacja o egzaminie maturalnym z informatyki dla absolwentów niesłyszących… **89**

Uchwała Rady Głównej Nauki i Szkolnictwa Wyższego oraz Konferencji Rektorów Akademickich Szkół Polskich o informatorach maturalnych od 2023 roku …….. **109**

1. Opis egzaminu maturalnego z informatyki

WSTĘP

Informatyka jest przedmiotem dodatkowym na egzaminie maturalnym. Ten przedmiot może być zdawany tylko na poziomie rozszerzonym.

Egzamin maturalny z informatyki sprawdza, w jakim stopniu uczeń IV klasy szkoły ponadpodstawowej, V klasy technikum oraz II klasy branżowej szkoły II stopnia spełnia wymagania określone w podstawie programowej kształcenia ogólnego dla szkoły ponadpodstawowej[1]..

Informator zawiera przykładowe zadania egzaminacyjne wraz z rozwiązaniami i odniesieniami do wymagań podstawy programowej. Zadania w *Informatorze* nie wyczerpują możliwych typów zadań, które mogą wystąpić w arkuszu egzaminacyjnym. Nie odnoszą się również do wszystkich wymagań z zakresu informatyki określonych w podstawie programowej. Dlatego *Informator* nie może być jedyną ani nawet główną wskazówką do planowania procesu kształcenia w szkole. Tylko realizacja wszystkich wymagań z podstawy programowej, zarówno ogólnych, jak i szczegółowych, może zapewnić zdobycie przez uczniów wiedzy i umiejętności informatycznych zgodnych z tą podstawą, a przez to ich właściwe przygotowanie do egzaminu maturalnego.

Przed przystąpieniem do dalszej lektury *Informatora* warto zapoznać się z ogólnymi zasadami obowiązującymi na egzaminie maturalnym od roku szkolnego 2022/2023. Są one określone w rozporządzeniu Ministra Edukacji i Nauki z dnia 26 lutego 2021 r. w sprawie egzaminu maturalnego (Dz.U. poz. 482) oraz – w skróconej formie – w części ogólnej *Informatora o egzaminie maturalnym od roku szkolnego 2022/2023*, dostępnej na stronie internetowej Centralnej Komisji Egzaminacyjnej (https://cke.gov.pl/) i na stronach internetowych okręgowych komisji egzaminacyjnych.

ZADANIA NA EGZAMINIE

W arkuszu egzaminacyjnym znajdą się zarówno zadania zamknięte, jak i otwarte oraz praktyczne.

Zadania zamknięte to takie, w których uczeń wybiera odpowiedź spośród podanych. Wśród zadań zamkniętych znajdą się m.in. zadania wyboru wielokrotnego, zadania typu prawda-fałsz oraz zadania na dobieranie.

[1] Rozporządzenie Ministra Edukacji Narodowej z dnia 30 stycznia 2018 r. w sprawie podstawy programowej kształcenia ogólnego dla liceum ogólnokształcącego, technikum oraz branżowej szkoły II stopnia (Dz.U. z 2018 r. poz. 467, z późn. zm.).

Zadania otwarte to takie, w których uczeń przedstawia przygotowane samodzielnie rozwiązanie. Wśród zadań otwartych znajdą się m. in.:

- zadania z luką, wymagające uzupełnienia luk w podanym zdaniu lub krótkim tekście
- zadania krótkiej odpowiedzi, wymagające stworzenia krótkiego tekstu
- zadanie rozszerzonej odpowiedzi, np. wymagające zaprojektowania i zapisania algorytmu.

Zadania praktyczne, to zadania, które wymagają użycia komputera i zapisania komputerowej realizacji rozwiązania np. programu w wybranym języku programowania, bazy danych, arkusza kalkulacyjnego itp.

Zadania mogą występować pojedynczo lub w wiązkach tematycznych. Część zadań wymaga umiejętności skorzystania z materiałów źródłowych dołączonych do arkusza, np. plików z danymi.

W zadaniach egzaminacyjnych szczególny nacisk kładzie się na sprawdzanie umiejętności związanych z algorytmiką i programowaniem. Arkusz zawiera m.in. kilka zadań algorytmiczno-programistycznych o zróżnicowanym poziomie trudności, z danymi różnych typów, wymagających zaprojektowania algorytmów i ich zaprogramowania. Każdy arkusz zawiera także zadania wymagające przeprowadzenia analizy zadanego algorytmu. Rozwiązanie większości zadań wymaga użycia komputera.

OPIS ARKUSZA EGZAMINACYJNEGO

Egzamin maturalny z informatyki trwa 210 minut[2].

Zdający przez cały czas trwania egzaminu ma do dyspozycji autonomiczne stanowisko komputerowe i może korzystać wyłącznie z programów i danych zapisanych na dysku twardym oraz na innych nośnikach stanowiących wyposażenie stanowiska lub otrzymanych z arkuszem egzaminacyjnym.

W arkuszu egzaminacyjnym część zadań wymaga jedynie zapisu odpowiedzi w arkuszu papierowym. Rozwiązanie większości zadań wymaga użycia komputera i zapisania komputerowej realizacji rozwiązania. W zadaniach tego typu rozwiązanie, w którym brakuje komputerowej realizacji, nie będzie oceniane.

Za wykonanie wszystkich zadań można uzyskać maksimum 50 punktów, w tym do około 60% punktów można uzyskać za zadania dotyczące analizy algorytmów i programowania. Większość zadań (około 90% punktów) to zadania otwarte.

[2] Czas trwania egzaminu może zostać wydłużony w przypadku uczniów ze specjalnymi potrzebami edukacyjnymi, w tym niepełnosprawnych, oraz w przypadku cudzoziemców. Szczegóły są określane w *Komunikacie dyrektora Centralnej Komisji Edukacyjnej w sprawie szczegółowych sposobów dostosowania warunków i form przeprowadzania egzaminu maturalnego* w danym roku szkolnym.

ZASADY OCENIANIA

Zadania zamknięte i zadania otwarte z luką

Zadania zamknięte są oceniane – w zależności od maksymalnej liczby punktów, jaką można uzyskać za rozwiązanie danego zadania – zgodnie z następującymi zasadami:

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna albo odpowiedź niepoprawna albo brak odpowiedzi.

2 pkt – odpowiedź całkowicie poprawna.
1 pkt – odpowiedź częściowo poprawna lub odpowiedź niepełna.
0 pkt – odpowiedź niepoprawna albo brak odpowiedzi.

Zadania otwarte krótkiej odpowiedzi

Za rozwiązanie zadania otwartego krótkiej odpowiedzi będzie można otrzymać od 0 do 3 punktów. Zasady oceniania będą opracowywane odrębnie dla każdego zadania.

Za każde poprawne rozwiązanie, inne niż opisane w zasadach oceniania, można przyznać maksymalną liczbę punktów, o ile rozwiązanie jest merytorycznie poprawne, zgodne z poleceniem i warunkami zadania.

Zadania otwarte rozszerzonej odpowiedzi

Za rozwiązanie zadania otwartego rozszerzonej odpowiedzi będzie można otrzymać od 0 do 5 punktów. Schemat oceniania będzie opracowywany odrębnie dla każdego zadania.

Zadania praktyczne wymagające użycia komputera i zapisania komputerowej realizacji

Za rozwiązanie zadania praktycznego można będzie otrzymać od 0 do 4 punktów. Do oceny w takim zadaniu należy przekazać pliki zawierające komputerowe realizacje rozwiązań oraz odpowiedzi zapisane w pliku tekstowym lub w arkuszu egzaminacyjnym – zgodnie z treścią zadania.

W zadaniach tego typu oceniane są rzeczywiste efekty i osiągnięte rezultaty przez zdającego, tj.: wyniki obliczeń w arkuszu kalkulacyjnym, wyniki symulacji, odpowiedzi uzyskane za pomocą kwerend, wyniki działania programu napisanego przez zdającego. Dołączenie komputerowej realizacji zadania (czyli programu, arkusza kalkulacyjnego lub bazy danych) jest wymagane.

2. Przykładowe zadania z rozwiązaniami

W *Informatorze* dla każdego zadania podano:

- liczbę punktów możliwych do uzyskania za jego rozwiązanie (po numerze zadania)
- najważniejsze wymagania ogólne i szczegółowe, które są sprawdzane w tym zadaniu
- zasady oceniania rozwiązania zadania
- poprawne rozwiązanie zadania zamkniętego albo przykładowe rozwiązania zadania otwartego.

Symbol kartki zamieszczony w nagłówku zadania zwraca uwagę na to, że zadanie nie wymaga użycia komputera i odpowiedź do zadania należy zapisać tylko w miejscu na to przeznaczonym w arkuszu egzaminacyjnym. W przypadku pozostałych zadań do oceny należy przekazać pliki zawierające komputerowe realizacje rozwiązań oraz odpowiedzi zapisane w pliku tekstowym lub w arkuszu egzaminacyjnym – zgodnie z treścią zadania. Brak plików zawierających realizacje komputerowe rozwiązań w przypadku tych zadań jest traktowany jako brak rozwiązania.

Zadanie 1.

Segmentem nazwiemy spójny ciąg elementów tablicy składający się z **co najmniej 1 elementu.**

Przykład: dla tablicy *A* = [1, 8, 4, 2, 7, 9] segmentami są ciąg 1,8,4 oraz ciąg 8,4,2,7, natomiast nie jest segmentem ciąg 8,2,7,9 (bo w tablicy *A* pomiędzy liczbą 8 a liczbą 2 jest liczba 4)**.**

Zadanie 1.1. (0–1)

Dana jest tablica A liczb całkowitych o następującej zawartości:

A = [2, -3, 1, -7, 4, -2, -1, 5, -3, 2, -1].

Podaj wartość pierwszego oraz wartość ostatniego elementu segmentu o maksymalnej sumie (w tej tablicy jest tylko jeden taki segment).

Odpowiedź _____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Wymaganie szczegółowe

I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: […] c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie […].

Zasady oceniania
1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie
4, 5

Komentarz
Segment o maksymalnej sumie (suma = 6) to: 4,-2,-1, 5.

Zadanie 1.2. (0–1)

Dana jest *n*-elementowa tablica *A* o zawartości [1, 2, 3, ..., *n* – 1, *n*]. Podaj liczbę segmentów w tej tablicy.

Odpowiedź _____

Wymaganie ogólne
I. Rozumienie, analizowanie i rozwiązywanie problemów.

Wymaganie szczegółowe
Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: […] c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie […].

Zasady oceniania
1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna lub brak odpowiedzi.

Rozwiązanie
n (*n* + 1) / 2

Zadanie 1.3. (0–3)

Elementy pewnej 1000-elementowej tablicy *A* zapisano kolejno w pliku `dane1_3.txt`. Każda z liczb w pliku `dane1_3.txt` należy do przedziału od [-100, 100] i jest zapisana w oddzielnym wierszu.
Napisz program wyznaczający największą sumę segmentu tablicy A.

Do oceny oddajesz:

- plik `zadanie1_3.txt` zawierający jedną liczbę będącą odpowiedzią do zadania (największą sumą)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach):

_____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: […] c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie […];
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);
II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów.

Zasady oceniania

3 pkt – odpowiedź poprawna – program dający w wyniku poprawną sumę.
2 pkt – program poprawny składniowo, ale znajdujący największą sumę elementów podciągu zaczynającego się zawsze od pierwszego elementu tablicy.
1 pkt – program poprawny składniowo wyliczający sumy elementów podciągów, ale nieznajdujący największej sumy.
0 pkt – odpowiedź niepoprawna albo brak odpowiedzi.

Rozwiązanie

2265

Fragment przykładowego programu:

```
ifstream plik("dane1_3.txt");
ofstream wynik("zadanie1_3.txt");
const int n = 1000;
```

```
int t[n];
for (int i = 0; i < n; i++)
   plik >> t[i];
int max = -100;
for (int dl = n; dl >= 1; dl--) {
   for (int pocz = 0; pocz <= n - dl; pocz++) {
      int suma = 0;
      for (int k = 0; k < dl; k++)
         suma += t[k + pocz];
      if (suma > max)
         max = suma;
       }
}
wynik << max << endl;
```

Komentarz:

Powyższy program przykładowy ma złożoność sześcienną. Można napisać programy o znacznie lepszej (kwadratowej, a nawet liniowej) złożoności, ale ten program jest bardzo prosty i wynika wprost z definicji problemu, a dla danych z pliku `dane1_3.txt` jest wystarczająco szybki.

Zadanie 1.4. (0–4)

Elementy pewnej tablicy *A* o 100 000 elementów zapisano kolejno w pliku `dane1_4.txt`. Każda z liczb w pliku `dane1_4.txt` należy do przedziału od [-100, 100] i jest zapisana w oddzielnym wierszu.

Przyjmujemy, że pierwszy element tablicy ma indeks równy 1. Napisz program wypisujący indeks pierwszego i indeks ostatniego elementu segmentu o największej sumie. W tablicy *A* jest tylko jeden taki segment, **a suma jego elementów jest dodatnia.**

Do oceny oddajesz:

- plik `zadanie1_4.txt` zawierający odpowiedź do zadania zapisaną w jednym wierszu (dwie liczby oddzielone spacją będące odpowiednio indeksem pierwszego i indeksem ostatniego elementu segmentu o największej sumie)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach) _____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: […] c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie […];
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);
II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów.

Zasady oceniania

4 pkt – odpowiedź poprawna oraz program dający w wyniku początek i koniec segmentu o największej sumie.
3 pkt – wyniki częściowo poprawne (np. przesunięte o jeden).
2 pkt – podano tylko jeden kraniec segmentu.
1 pkt – podano maksymalną sumę zamiast końców segmentu.
0 pkt – odpowiedź niepoprawna albo brak odpowiedzi.

Rozwiązanie

63669 70769

Fragment przykładowego programu:

```
ifstream plik ("dane1_4.txt");
ofstream wynik ("zadanie1_4.txt");
const long n=100000;
int t[n];
for (long i=0; i<n; i++)
    plik>>t[i];
long maks;  // największa suma niepustego segmentu
long maks_pocz;  // początek najlepszego niepustego segmentu
long maks_kon;  // koniec najlepszego niepustego segmentu
long ost_suma; // najlepsza niepusta suma segmentu na końcu przejrzanej części tablicy
long ost_pocz; // początek najlepszego niepustego segmentu na końcu przejrzanej części tablicy

// pierwszy rozważany niepusty segment to t[0]:
maks = ost_suma = t[0];
maks_pocz = maks_kon = ost_pocz = 0;

for (long i = 1; i < n; i++) {
   if (ost_suma >= 0)  // Czy ost_suma + t[i] >= t[i]?
```

```
        ost_suma += t[i];
    else {
        ost_suma = t[i];
        ost_pocz = i;
    }
    if (maks < ost_suma) {
        maks = ost_suma;
        maks_pocz = ost_pocz;
        maks_kon = i;
    }
}

wynik << maks_pocz + 1 << " " << maks_kon + 1 << endl;
// Tablica, której wartości były w pliku wejściowym miała indeksy przesunięte o 1
// musimy zatem uwzględnić poprawkę na przesuniecie indeksów względem tablicy t,
// do której wczytano dane z pliku
```

Zadanie 2.

Wyrażeniem nawiasowym nazywamy dowolny skończony ciąg nawiasów: „[” i „]”. Przykładowo: [[ ] [ ]]

Wyrażenie nawiasowe jest poprawne, jeśli:
– jest wyrażeniem pustym (nie zawiera żadnych nawiasów)
– jest postaci AB, gdzie A i B są poprawnymi wyrażeniami nawiasowymi
– jest postaci [A], gdzie A jest poprawnym wyrażeniem nawiasowym

Przykład: wyrażenia [[ ]] oraz [[ ] [ ]] są poprawne. Niepoprawne jest za to wyrażenie [ ]] [ ]].

Niech w_1, w_2, …, w_n będą kolejnymi nawiasami w pewnym wyrażeniu nawiasowym W. Przyjmijmy teraz, że z każdym nawiasem otwierającym „[” wiążemy liczbę +1, a z każdym nawiasem zamykającym „]” – liczbę -1. Niech s_i będzie liczbą związaną z nawiasem w_i. Wówczas **głębokością nawiasu w_k w wyrażeniu W** nazywamy sumę:

$$S_k = s_1 + s_2 + \ldots + s_k$$

Głębokością wyrażenia W nazwiemy największą głębokość jego nawiasów, czyli maksimum z liczb S_k.

Zadanie 2.1. (0–1)

Wskaż, czy dane wyrażenie nawiasowe jest poprawne. Wpisz **Tak**, jeśli wyrażenie jest poprawne lub **Nie** – jeśli nie jest poprawne.

Wyrażenie nawiasowe	Poprawne (Tak/Nie)
[]	Tak
[][]	
[[][]][]	
[][[][[][[]]]]	
[[][][[][]][]]	

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Wymaganie szczegółowe

Zdający:

1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
3. wyróżnia w problemie podproblemy i charakteryzuje: metodę połowienia, stosuje podejście zachłanne i rekurencję;
4. porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji.

Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Wyrażenie nawiasowe	Poprawne (tak/nie)
[]	Tak
[][]	Tak
[[][]][]	Nie
[][[][[][[]]]]	Tak
[[][][[][]][]]	Tak

Zadanie 2.2. (0–1)

Dla zadanych przykładów policz głębokość wyrażenia.

Wyrażenie nawiasowe	Głębokość
[]	1
[][]	
[[][]]	
[[][[]]]	
[[[[][]][]]]	

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Wymaganie szczegółowe

Zdający:

1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
3. wyróżnia w problemie podproblemy i charakteryzuje: metodę połowienia, stosuje podejście zachłanne i rekurencję.
4. porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji.

Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Wyrażenie nawiasowe	Głębokość
[]	1
[][]	1
[[][]]	2
[[][[]]]	3
[[[[][]][]]]	4

Zadanie 2.3. (0–3)

Dane w pliku `dane2_3.txt` zapisano w oddzielnych wierszach. W każdym wierszu znajduje się **poprawne** wyrażenie nawiasowe złożone z nawiasów kwadratowych (nieoddzielonych żadnym znakiem).
Napisz program, który dla zadanych wyrażeń nawiasowych w pliku `dane2_3.txt` obliczy ich głębokości.

Do oceny oddajesz:

- plik `zadanie2_3.txt` zawierający odpowiedź do zadania (głębokości kolejnych wyrażeń, każda w oddzielnym wierszu)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.2. do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;
I.5. sprawdza poprawność działania algorytmów dla przykładowych danych
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);
II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów.

Zasady oceniania

3 pkt – odpowiedź poprawna i program dający poprawne wyniki.
2 pkt – program dający wyniki różniące się zawsze o +/- 1 od prawidłowych.
1 pkt – program dający poprawne wyniki dla przynajmniej połowy danych.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Dla danych z pierwszych czterech wierszy w pliku dane2_3.txt:
[][][]
[[[]]]
[[][][]]
[[[]]]

Odpowiedź
1
3
2
3

Fragment przykładowego programu:

```
ifstream dane("dane2_3.txt");
ofstream wynik("zadanie2_3.txt");

string wiersz;
while (dane >> wiersz) {
   int suma = 0;
   int max = 0;

   for (string::size_type i = 0; i < wiersz.size(); i++) {
      if (wiersz[i] == '[') {
         suma++;
         if (suma > max)
            max = suma;
      } else
         suma--;
   }
   wynik << max << endl;
}
```

Zadanie 2.4. (0–3)

Napisz program, który dla wyrażeń nawiasowych zapisanych w pliku `dane2_4.txt` sprawdzi, czy są one poprawne. Dane w pliku zapisano po jednym wyrażeniu w wierszu, podobnie jak w pliku o nazwie `dane2_3.txt`.

Do oceny oddajesz:

- plik `zadanie2_4.txt` zawierający odpowiedź do zadania (w kolejnych wierszach odpowiedzi „tak”, jeśli wyrażenie jest poprawne i „nie” – jeśli nie jest poprawne)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.2. do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;
I.3. wyróżnia w problemie podproblemy i charakteryzuje: metodę połowienia, stosuje podejście zachłanne i rekurencję;
I.5. sprawdza poprawność działania algorytmów dla przykładowych danych
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);
II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów.

Zasady oceniania

3 pkt – odpowiedź poprawna i program dający poprawne wyniki.
2 pkt – odpowiedź poprawna dla przynajmniej dwóch typów przykładów z pliku (np. dla przykładów typu [[[ ]]] oraz [] [] []).
1 pkt – odpowiedź poprawna dla tylko jednego typu przykładów z pliku (np. tylko dla przykładów typu [[[ ]]]).
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Dla danych z pierwszych czterech wierszy pliku dane2_4.txt:
[][][]
[[[]]]
[[][][]]
[[[]]]]

Odpowiedź
tak
tak
tak
nie

Fragment przykładowego programu:

```
ifstream dane("dane2_4.txt");
ofstream wynik("zadanie2_4.txt");

string wiersz;
while (dane >> wiersz) {
   int balans = 0;
   bool poprawne = true;
```

```
    for (string::size_type i = 0; i < wiersz.size(); i++) {
        if (wiersz[i] == '[')
            balans++;
        else { // wiersz[i] == ']'
            balans--;
            if (balans < 0) {
                poprawne = false;
                break;
            }
        }
    }
  if (poprawne && balans == 0)
      wynik << "tak\n";
  else
      wynik << "nie\n";
}
```

Zadanie 3.

Rozważamy przedziały domknięte liczb całkowitych. Każdy taki przedział można opisać parą liczb całkowitych [*a, b*], ***a*** ≤ *b*, w której *a* oznacza początek przedziału, natomiast *b* jest jego końcem.
Do przedziału [*a, b*] należą wszystkie liczby całkowite *c* spełniające nierówności *a* ≤ *c* ≤ *b*.
Liczbę *b* – *a* + 1 nazywamy długością przedziału.

Dla dwóch przedziałów P i Q mówimy, że **P zawiera się w Q**, gdy każda liczba z należąca do przedziału P należy także do przedziału Q. O przedziale Q mówimy wtedy, że zawiera przedział P.

Łańcuchem przedziałów nazywamy każdy skończony ciąg przedziałów P_1, P_2, …, P_k, w którym każdy przedział o numerze większym od 1 zawiera przedział go poprzedzający w tym ciągu. Liczbę k nazywamy długością łańcucha.

Przykład 1:
Rozważmy 6 przedziałów: A = [-2,4], B = [-4,3], C = [-3,6], D = [0,3], E = [1,1], F = [7,9].
Przedział A ma długość 7. Przedział C zawiera przedział A, natomiast przedziały E, D, A, C tworzą łańcuch o długości 4.

W pliku `dane3.txt` zapisano 2023 par liczb całkowitych z przedziału [-2023, 2023], po jednej parze w wierszu. Para liczb *a*, *b* w wierszu opisuje przedział [*a, b*]. Liczby w wierszu są rozdzielone pojedynczym znakiem odstępu, a pierwsza z liczb w parze nigdy nie jest większa od drugiej. Wiadomo, że przedziały w pliku `dane3.txt` się nie powtarzają (w każdym wierszu znajduje się inna para liczb) i że nie wszystkie przedziały mają tę samą długość.

Przykład 2:
Poniższe dane opisują przedziały z przykładu 1:
-2 4
-4 3
-3 6
0 3
1 1
7 9

Napisz **programy** dające odpowiedzi do poniższych zadań.

Zadanie 3.1. (0–2)

Podaj dwie najmniejsze (różne) liczby, które są długościami przedziałów zapisanych w pliku `dane3.txt`.

Do oceny oddajesz:

- plik `zadanie3_1.txt` zawierający odpowiedź do zadania zapisaną w jednym wierszu (dwie różne liczby oddzielone spacją będące najmniejszymi z liczb, które są długościami przedziałów zapisanych w pliku dane3.txt)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.2. do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2;
II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – poprawna jedna liczba wyniku.
0 pkt – odpowiedź niepoprawna albo brak odpowiedzi.

Rozwiązanie

3, 4

Fragment przykładowego programu:

```
const int n = 2023;
int min=4047;
int min2=min;

ifstream dane ("dane3.txt");
ofstream wynik("zadanie3_1.txt");

 for (int i = 0; i < n; i++) {
      int pocz, kon;
      dane >> pocz >> kon;
      int dlug = kon - pocz + 1;
      if (dlug < min) {
          min2 = min;
          min = dlug;
      } else if (dlug < min2)
         min2 = dlug;
  }
wynik<< min<< " " <<min2;
```

Zadanie 3.2. (0–2)

Wyznacz długość przedziału, która się powtarza najczęściej wśród przedziałów zapisanych w pliku `dane3.txt`. Gdy jest więcej takich długości, podaj największą z nich.

Do oceny oddajesz:

- plik `zadanie3_2.txt` zawierający odpowiedź do zadania zapisaną w jednym wierszu (jedna liczba równa najczęstszej długości przedziału)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
II.1. projektuje i tworzy rozbudowane programy w procesie rozwiązywania problemów, wykorzystuje w programach dobrane do algorytmów struktury danych, w tym struktury dynamiczne i korzysta z dostępnych bibliotek dla tych struktur;
II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów:
c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie;
II.4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym;
I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach oraz algorytmy:
b) znajdowania określonego elementu w zbiorze: lidera, idola, elementu w zbiorze uporządkowanym metodą binarnego wyszukiwania.

Zasady oceniania

2 pkt – podanie poprawnej odpowiedzi.
1 pkt – zliczenie ilości wystąpień różnych długości przedziałów.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

350
(występuje 9 razy)

Fragment przykładowego programu:

```
const int n = 4048;
ifstream dane("dane3.txt");
ofstream wynik("zadanie3_2.txt");

int dlugosc[n];
for (int i = 0; i < n; i++)
   dlugosc[i] = 0;

int pocz, kon;
while (dane >> pocz >> kon)
   dlugosc[kon - pocz + 1 ]++;
```

```
int dl = -1, ile = -1;
for (int i = 1; i < n; i++) {
   if (dlugosc[i] >= ile) {
      ile = dlugosc[i];
      dl = i;
     }
  }
wynik << dl << endl;
```

Zadanie 3.3. (0–3)

Oblicz długość najdłuższego łańcucha przedziałów, który można utworzyć z przedziałów zapisanych w pliku `dane3.txt`.

Wskazówka: Dla każdego przedziału można obliczyć długość najdłuższego łańcucha, którego ten przedział jest początkiem, w kolejności od przedziałów najkrótszych do przedziałów najdłuższych.

Do oceny oddajesz:

- plik `zadanie3_3.txt` zawierający odpowiedź do zadania zapisaną w jednym wierszu (jedna liczba będąca długością najdłuższego łańcucha przedziałów)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
II.1. projektuje i tworzy rozbudowane programy w procesie rozwiązywania problemów, wykorzystuje w programach dobrane do algorytmów struktury danych, w tym struktury dynamiczne i korzysta z dostępnych bibliotek dla tych struktur;
II.2. stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów;
II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;
I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów:

c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie;

Zasady oceniania

3 pkt – podanie poprawnej odpowiedzi.
2 pkt – podanie odpowiedzi różniącej się o 1 od prawidłowej.
1 pkt – podanie długości jednego podciągu np. zaczynającego się od pierwszego przedziału.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

29

Fragment przykładowego programu:

```
struct Przedzial{  // informacja o przedziale
   int pocz;     // początek przedziału
   int kon;      // koniec przedziału
   int dlugosc;  // długość najdłuższego łańcucha, którego początkiem jest ten przedział
  Przedzial (int pocz, int kon, int dlugosc): pocz (pocz), kon (kon), dlugosc (dlugosc) {}
 };

bool zawierasie (Przedzial e1, Przedzial e2) { // czy e1 zawiera się w e2
    return e2.pocz <= e1.pocz && e1.kon <= e2.kon;
 }
bool krotszy (Przedzial e1, Przedzial e2) {
    return (e1.kon - e1.pocz) < (e2.kon - e2.pocz);
 }
int main () {
    ifstream dane("dane3.txt");
    ofstream wynik("zadanie3_3.txt");

    vector<Przedzial> przedzialy;
    int pocz, kon;

   // wczytujemy dane
    while (dane >> pocz >> kon)
       przedzialy.push_back (Przedzial (pocz, kon, 1));
// początkowo najdłuższy znaleziony łańcuch zaczynający się w [pocz, kon]
// składa się wyłącznie z tego przedziału, więc ma długość 1
// sortujemy przedziały względem długości
    sort(przedzialy.begin(), przedzialy.end(), krotszy);

 // dla przedziałów tej samej długości kolejność obojętna, bo nie zawierają się w sobie
 // dla i-tego przedziału liczymy maksymalną dł. łańcuchów o początku w tym przedziale
    for (vector<Przedzial>::size_type i = 1; i < przedzialy.size(); i++)
 // Maksymalne długości łańcuchów zaczynających się od przedziałów krótszych niż i-ty
 // zostały wyznaczone w poprzednich obrotach pętli
```

```
    for (vector<Przedzial>::size_type j = 0; j < i; j++)
        // przeglądamy wszystkie przedziały nie dłuższe niż i-ty
        if (zawierasie(przedzialy[j], przedzialy[i]))
          // sprawdzamy, czy można i warto dodać łańcuch o początku w j-ty przedziale
          //do łańcucha zaczynającego się od i-tego przedziału
          if (przedzialy[i].dlugosc < przedzialy[j].dlugosc + 1)
            przedzialy[i].dlugosc = przedzialy[j].dlugosc + 1;
    // Znajdujemy największą długość łańcucha
     int maks = 1;
     for (Przedzial& elt: przedzialy)
        if (elt.dlugosc > maks)
           maks = elt.dlugosc;
   // Wypisujemy najlepszy
    wynik << maks << endl;
}
```

Zadanie 4.

Jednym z najpopularniejszych algorytmów porządkowania jest sortowanie przez wstawianie. W tym zadaniu odwołujemy się do zapisu algorytmu podanego poniżej:

Dane:

n – dodatnia liczba całkowita

A[1..n] – tablica zawierająca ciąg n liczb całkowitych $x_1, x_2, \ldots, x_n$, gdzie $A[i] = x_i$

Wynik:

tablica A zawierająca te same dane, ale uporządkowane niemalejąco,
tzn. dla każdego $i = 2, \ldots, n$, $A[i] \geq A[i-1]$

Algorytm SortW:

dla $i = 2, 3, \ldots, n$ **wykonaj**
    $v \leftarrow A[i]$
    $j \leftarrow i$
    dopóki $(j > 1)$ oraz $(v < A[j - 1])$ **wykonaj**
        $A[j] \leftarrow A[j - 1]$
        $j \leftarrow j - 1$
    $A[j] \leftarrow v$

Uwaga: przyjmij, że jeśli warunek $(j > 1)$ nie jest spełniony, to warunek $(v < A[j - 1])$ nie jest już sprawdzany.

Na potrzeby tego zadania przyjmiemy, że przypisanie $A[j] \leftarrow A[j - 1]$ jest **operacją dominującą**.

Zadanie 4.1. (0–1)

Podaj minimalną i maksymalną liczbę operacji dominujących w algorytmie SortW dla zadanego *n*.

Odpowiedź:
minimalna: _____

maksymalna: _____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.3. objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność.

Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Minimalna (optymistyczny przypadek): 0
Maksymalna: *n**(*n* – 1) / 2

Uwaga: akceptujemy zapis maksymalnej liczby operacji jako sumy np. $\sum_{i=2}^{n}(i-1)$.

Zadanie 4.2. (0–1)

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.

Pesymistyczna złożoność obliczeniowa algorytmu SortW mierzona liczbą operacji dominujących jest

A. sześcienna.
B. kwadratowa.
C. liniowa.
D. logarytmiczna.

Wymaganie ogólne
I. Rozumienie, analizowanie i rozwiązywanie problemów.

Wymaganie szczegółowe
Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.3. objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność.

Zasady oceniania
1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie
B

Zadanie 4.3. (0–2)
W pliku `dane4.txt` zapisano ciąg x złożony z 2023 różnych liczb całkowitych $x_1, x_2, \ldots, x_{2023}$ z przedziału [1, 2023], po jednej liczbie w wierszu – w wierszu i-tym liczbę x_i.

Podaj największe takie i, dla którego liczba par (x_i, x_j) takich, że $x_i > x_j$ oraz $i > j$ jest największa.

Do oceny oddajesz:
- plik `zadanie4_3.txt` zawierający odpowiedź do zadania zapisaną w jednym wierszu (jedna liczba będąca szukaną wartością i)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Wymaganie ogólne
I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe
Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.3. objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;

II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2;
II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;
I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach [..].

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – wynik różny o 1 (np. z powodu indeksowania od 0).
0 pkt – odpowiedź niepoprawna albo brak odpowiedzi.

Rozwiązanie

największe i – 1979

Fragment przykładowego programu:

```
ifstream plik("dane4.txt");
ofstream wynik("zadanie4_3.txt");

 const int n = 2023;
 int x[n];
 for (int i=0; i<n; i++)
       plik >> x[i];

  int liczba=0, maxi=0;

  for(int i=1; i<n; i++){
      int par=0;
      for(int j=0; j<i; j++)
         if (x[i] > x[j])
             par++;

      if (par >= liczba){
        liczba=par;
        maxi=i;
      }
   }
wynik<<maxi+1;
```

Zadanie 5.

W tym zadaniu zajmujemy się algorytmami działającymi na *n*-elementowej tablicy liczb całkowitych A[1..*n*], gdzie *n* jest dodatnią liczbą całkowitą.

Poniżej zapisano rekurencyjną procedurę W, której parametrem jest liczba całkowita *j* z przedziału [1, *n*]:

```
procedura W(j):
    jeśli j > 1 to
        jeśli A[j] < A[j-1] to
            v ← A[j]
            A[j] ← A[j-1]
            A[j-1] ← v
            W(j-1)
```

Oto **przykładowa**, 10-elementowa tablica A[1..10] o zawartości [2,4,6,8,10,9,7,5,3,1].

Zadanie 5.1. (0–2)

Procedurę **W** wywołano dwukrotnie: najpierw z parametrem 7, **a następnie** z parametrem 9. Podaj zawartość tablicy *A* po pierwszym i po drugim wywołaniu.

Odpowiedź:

Po pierwszym wywołaniu z parametrem 7: _____

Po drugim wywołaniu z parametrem 9: _____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.3. objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;
I.4. porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji.

Zasady oceniania

2 pkt – odpowiedź poprawna dla obu wywołań.
1 pkt – odpowiedź poprawna dla jednego wywołania.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Po pierwszym wywołaniu z parametrem 7: 2 4 6 7 8 10 9 5 3 1
Po drugim wywołaniu z parametrem 9: 2 3 4 6 7 8 10 9 5 1

Zadanie 5.2. (0–1)

Poniższe algorytmy ***S1*** i ***S2*** różnią się tylko kolejnością wywoływania procedury ***W***.

Algorytm *S1*:

dla *i* = 2, …, *n* **wykonaj**
W(*i*)

Algorytm *S2*:

dla *i* = *n*, *n*-1, …, 2 **wykonaj**
W(*i*)

Podaj zawartość tablicy *A*[1..*n*] = [*n*, *n* – 1, …, 1] po wykonaniu algorytmu ***S1***.

Odpowiedź: _____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.3. objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;
I.4. porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji.

Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Zawartość tablicy A po wykonaniu algorytmu *S1*
1, 2, …, *n*.

Zadanie 5.3. (0–2)

Dla każdego z algorytmów *S1*, *S2* zdecyduj, czy jest on algorytmem sortującym. Odpowiedź uzasadnij.

Odpowiedź:
Algorytm *S1* (tak/nie) _____

Algorytm *S2* (tak/nie) _____

Uzasadnienie:

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.3. objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;
I.4. porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji.

Zasady oceniania

2 pkt – odpowiedź poprawna dla obu algorytmów i poprawne uzasadnienie.
1 pkt – odpowiedź poprawna, ale bez uzasadnienia albo z niepoprawnym uzasadnieniem
ALBO
odpowiedź poprawna dla jednego algorytmu i poprawne uzasadnienie.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

S1 tak (jest algorytmem sortującym)
S2 nie (nie jest algorytmem sortującym)

Uzasadnienie:
W algorytmie *S1* po wywołaniu *W*(*i*) element stojący na miejscu *i*-tym jest przesuwany „w lewo" w odpowiednie miejsce uporządkowanego fragmentu tablicy A[1..*i*–1]. Tak naprawdę algorytm *S1* jest algorytmem sortowania przez wstawianie, gdzie wstawianie odbywa się za pomocą rekurencyjnej procedury *W*.

Algorytm *S2* nie posortuje np. 3-elementowej tablicy A = [3,1,2]. Po jego wykonaniu A = [1, 3, 2].

Zadanie 5.4. (0–3)

W wybranej przez siebie notacji zapisz nierekurencyjną wersję procedury ***W***.

Uwaga: W zapisie algorytmu możesz wykorzystać tylko operacje arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie, dzielenie całkowite, reszta z dzielenia), instrukcje porównania, instrukcje sterujące i przypisania do zmiennych lub samodzielnie napisane funkcje, wykorzystujące wyżej wymienione operacje.

Algorytm:

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.3. objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2;
I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach […].

Zasady oceniania

3 pkt – poprawny algorytm
w tym:
1 pkt – za poprawne inicjowanie zmiennych
1 pkt – za poprawną konstrukcję pętli
1 pkt – poprawne instrukcje w pętli.
0 pkt – odpowiedź niepoprawna albo brak odpowiedzi.

Rozwiązanie

Przykładowy zapis algorytmu w pseudokodzie:

```
procedura W(j):
 jeśli j > 1
    v ← A[j]
    i ← j
    dopóki (i > 1) oraz (A[i-1] > v) wykonaj
                A[i] ← A[i-1]
                 i ← i-1
    A[i] ← v
```

Zadanie 6.

W pliku `dane6.txt` zapisano 2023 napisy, po jednym w każdym wierszu. Każdy napis ma długość 100 i jest zbudowany wyłącznie z cyfr dziesiętnych 0, 1, …, 9. Wśród napisów nie ma napisu zbudowanego z samych zer.

Dla liczby całkowitej p spełniającej warunek $2 \le p \le 10$, powiemy, że złożony z samych cyfr napis jest ***p*-minimalny**, jeśli zawiera cyfrę $p - 1$ i nie zawiera cyfr większych od $p - 1$. Innymi słowy, będzie tak, gdy p jest najmniejszą podstawą systemu pozycyjnego, w którym taki napis da się zinterpretować, jako pewna liczba całkowita.

Przykład:
Oto przykład danych (zapisanych w pliku `dane6przyklad.txt`) złożonych z 5 napisów, z których każdy ma długość 10 i zbudowany jest z cyfr od 0 do 9:
2001030035
0010100001
7111190009
5550001110
0000000005

Napisz program(-y), który(-e) dający(-e) odpowiedzi do poniższych zadań. Uzyskane odpowiedzi zapisz zgodnie z poleceniami przy każdym zadaniu.

Zadanie 6.1. (0–3)

Dla każdego $p = 2, 3, \ldots, 10$, podaj, ile liczb z pliku `dane6.txt` jest p-minimalnych.

Przykład:
Dla przykładowych danych mamy:

Podstawa p	liczba liczb p-minimalnych
2	1
3	0
4	0
5	0
6	3
7	0
8	0
9	0
10	1

Odpowiedź:

Podstawa p	**liczba liczb *p*-minimalnych**
2	
3	
4	
5	
6	
7	
8	
9	
10	

Do oceny oddajesz:

- odpowiedź zapisaną w tabeli powyżej
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach).

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:

I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);

I.2. stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach: […] zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi;

I.3. objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;

II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2;

II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach;

I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: […] b) wykonywania działań na liczbach w systemach innych niż dziesiętny.

Zasady oceniania

3 pkt – odpowiedź poprawna.
2 pkt – odpowiedź poprawna dla 8 systemów.
1 pkt – odpowiedź z wartościami mniejszymi o 1 lub przypisanie wyników do największych cyfr, a nie podstaw.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Podstawa p	**liczba liczb *p*-minimalnych**
2	225
3	239
4	229
5	232
6	238
7	198
8	220
9	210
10	232

Fragment przykładowego programu:

```
ifstream plik("dane6.txt");
const int n = 2023;
const int dl = 100;

int c[10]= {0,0,0,0,0,0,0,0,0,0};
string s;
int max;

for (int i=0; i < n; i++){
   plik >> s;
   max = 0;
   for (int j=0; j<dl; j++)
   {
      int t = s[j]-'0';
      if (t > max)
           max=t;
   }
   c[max]++;
}
```

```
for (int i=1; i < 10; i++)
   cout << i+1 << " " << c[i] << endl;
```

Zadanie 6.2. (0–3)

Dla każdego *p* = 2, 3, …, 10 wskaż wśród *p*-minimalnych liczb zapisanych w pliku `dane6.txt` taką, której suma cyfr jest największa.

Uwaga: Jeśli w pliku nie ma liczby *p*-minimalnej dla danego *p*, to nie podajemy dla niej wyniku.

Przykład:
Dla przykładowych danych mamy:

podstawa	liczba *p*-minimalna z największą sumą cyfr
2	0010100001
6	5550001110
10	7111190009

Do oceny oddajesz:

- plik `zadanie6_2.txt` zawierający odpowiedź do zadania (w każdym wierszu pliku dwie liczby oddzielone spacją: oznaczające podstawę *p* oraz liczbę *p*-minimalną o największej sumie cyfr)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.2. stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach: […], zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi;
I.3. objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2;

II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;
I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach;
I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: […] d) zamiany wyrażenia na postać w odwrotnej notacji polskiej i obliczanie jego wartości na podstawie tej postaci.

Zasady oceniania

3 pkt – odpowiedź poprawna.
2 pkt – odpowiedź z maksymalnymi sumami cyfr zamiast wypisanych liczb z pliku.
1 pkt – odpowiedź poprawna dla przynajmniej jednej podstawy systemu pozycyjnego.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

2
11111101110100001111101101111101011001111011000011010111011111011111100010
11111010100101001101110111
3
22021122021101200110210202222210120211211222221121102202212201011122222102
02110011102022211201222212
4
31023030233333232220003102130310333112232131233002010223332123322331220022
33220223030330222123110303
5
20332433010200444421120243032033444444124310341141444311101420444443333023
30120243324132423443224230
6
04244543451524345042553152321052413135224501444214554155135004552124551533
50255121412341354435230533
7
44405553624206102355362465512664352034643061363556454666226460614341131546
56346003302120251560312564
8
75110361612470675146632745605604636241363414372375160761272244756745650046
42766472364553635576317744
9
73074668871775566617275308314737485865727271466671583784416658006283862735
87852686608173458128256575
10
43637842270897660385152489678629262528566176868888788367744489699071436640
99174980839005479979326163

Fragment przykładowego programu:

```
ifstream plik("dane6.txt");
ofstream wynik("zadanie6_2.txt");
```

```
const int n = 2023;
const int dl = 100;
int maxs[10]={0,0,0,0,0,0,0,0,0,0};
string liczby[10];
string s;

for(int i=0; i < n; i++){

   plik >> s;
   int max=0;
   int suma=0;
   for(int j=0; j<dl; j++){
    int t = s[j]-'0';
    suma += t;
    if(t>max)
       max=t;
   }
   if (suma > maxs[max]){

      maxs[max]=suma;
      liczby[max]=s;
    }

}

for (int i=1; i < 10; i++)
   if (maxs[i]>0)
        wynik << i+1 << " " << liczby[i] << endl;
```

Zadanie 6.3. (0–2)

Dla dodatniej, parzystej liczby całkowitej n powiemy, że napis $s[1..n]$ jest **antypalindromem**, jeżeli $s[i] \neq s[n - i + 1]$, dla każdego $i = 1, 2, \ldots, n/2$.

Zapisz wszystkie antypalindromy z pliku `dane6.txt` (każdy w osobnym wierszu, z zachowaniem kolejności z pilku z danymi) w pliku `zadanie6_3.txt`. W osobnym, ostatnim wierszu pliku zapisz ich liczbę.

Przykład:
Dla danych przykładowych antypalindromem jest tylko jeden napis: 7111190009.

Do oceny oddajesz:

- plik `zadanie6_3.txt` zawierający odpowiedź do zadania (w każdym wierszu pliku jeden antypalindrom i liczba antypalindromów w ostatnim wierszu pliku)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.2. stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach: […], zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi;
I.3. objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2;
II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;
I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach;
I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów […] wykonywania działań na liczbach w systemach innych niż dziesiętny.

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź poprawna bez liczby antypalindromów
ALBO
podana tylko poprawna liczba antypalindromów.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

14863571230330690483692339776957319481338573283002786460810007311554983485279735295202746116730687 72

88161000430076445258374062631112335058830726585705022618365814508170357802437654523472161202176703 21

53386843320088227451102048899149759918270810392395300858949919123721068843789650766513129969281330 04

72345115624756735503557135172931285542327323024478487179948877173011747409807198918068375855395457 31

4

Fragment przykładowego programu:

```
bool anty(string s){
    for (int i=0; i<s.length() / 2; i++)
      if (s[i] == s[s.length() - 1 - i])
          return false;
   return true;
}

int main(){
  const int n = 2023;
  ifstream plik("dane6.txt");
  ofstream wynik("zadanie6_3.txt");
  string s;

   int ile=0;
   for (int i=0; i<n; i++) {
      plik>>s;
      if (anty(s)) {
         ile++;
         wynik << s << endl;
       }
   }
   wynik << ile;
}
```

Zadanie 7.

Pewien tekst w języku polskim zapisano z użyciem wielkich liter alfabetu angielskiego (tzn. literę Ą zastąpiono literą A, literę Ć literą C itd.). W tekście nie ma żadnych innych znaków (odstępów, znaków interpunkcyjnych, cyfr itp.). Następnie zaszyfrowano go za pomocą prostego szyfru podstawieniowego i zapisano w pliku `szyfrogram.txt`, z zachowaniem podziału na wiersze.

Z wykorzystaniem dostępnych narzędzi informatycznych oraz danych z pliku `szyfrogram.txt` rozwiąż poniższe zadania.

Fragment pliku szyfrogram.txt:
FACOTXKNFEBMTTCVURBCFZKPDCAHAMNRLCTKB
XMVXKIAOYRNYXKNYXTXKPYCFUMPXALUYKNICTKB
FROXCNAYLMFMYEBMTXKLMUYEPMTKOCHK
OYPYRLXCPJUCFKPXCITKARLMFMAXLMNVUCHK
FKNFMUYKLEOJMVMNUYFXMNICTBEYCHBTXKJM

Zadanie 7.1. (0–2)

Przeprowadź analizę częstości występowania znaków w pliku `szyfrogram.txt`. Dla każdej litery alfabetu angielskiego określ, ile razy występuje ona w pliku.

Do oceny oddajesz:

- plik `zadanie7_1.txt` zawierający odpowiedź do zadania (zawierający 26 wierszy: w *i*-tym wierszu znajduje się *i*-ta kolejna litera, a po niej, oddzielona pojedynczym odstępem liczba jej wystąpień w tekście)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych;
II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;
II.4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym;
I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy [...].

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – podano poprawną liczbę wystąpień dla przynajmniej 5 liter.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

A 41
B 42
C 98
D 7

```
E 18
F 43
G 0
H 36
I 17
J 32
K 91
L 34
M 86
N 39
O 50
P 44
Q 0
R 31
S 0
T 60
U 38
V 16
W 2
X 92
Y 68
Z 15
```

Fragment przykładowego programu:

```
ifstream plik ("szyfrogram.txt");
ofstream zapis ("zadanie7_1.txt");

const int n = 26;
int t[n];

for (char i = 0; i < n; i++)
   t[i]=0;

char c;

while (plik >> c)
   t[c - 'A']++;

for (char i = 'A'; i <= 'Z'; i++)
   zapis << i << " " << t[i-'A'] << endl;
```

Zadanie 7.2. (0–1)

Poniżej podano „średnie” liczby wystąpień poszczególnych liter na 1000 znaków w tekście zapisanym w języku polskim[3]. Poniższe zestawienie zapisane jest także w pliku `czestosc.txt`.

A 98
B 16
C 44
D 39
E 91
F 2
G 17
H 7
I 92
J 15
K 42
L 41
M 36
N 60
O 86
P 34
Q 0
R 38
S 50
T 32
U 18
V 0
W 43
X 0
Y 31
Z 68

Uwaga: W celu uproszczenia, na potrzeby zadania, zakładamy, że liczby wystąpień poszczególnych liter w zaszyfrowanym tekście są dokładnie takie same jak w podanym zestawieniu.

Odszyfruj poniższe słowo, zaszyfrowane tak, jak cały szyfrogram. Odpowiedź zapisz poniżej.

Słowo zaszyfrowane: CAIMURJH

Słowo jawne: _____

[3] Dane zmodyfikowano na potrzeby zadania.

Wymaganie ogólne
I. Rozumienie, analizowanie i rozwiązywanie problemów.

Wymaganie szczegółowe
Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
I.2. 2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: [..] b) na tekstach: porównywania tekstów, wyszukiwania wzorca w tekście metodą naiwną, szyfrowania tekstu metodą Cezara i przedstawieniową.

Zasady oceniania
1 pkt – odpowiedź poprawna
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie
ALGORYTM

Zadanie 7.3. (0–3)
Korzystając z pliku czestosc.txt, odszyfruj cały tekst z pliku `szyfrogram.txt`.

Do oceny oddajesz:
- plik `zadanie7_3.txt` zawierający odpowiedź do zadania (odszyfrowany tekst)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach).

Wymaganie ogólne
I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe
Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych;
II.3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

II.4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym;
I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy [...].

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź z błędami przy maksymalnie dwóch źle odkodowanych literach.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

WLASNIEDWUKONNABRYKAWJECHALMLODYPANEKIOBIEGLSZYDZIEDZINIECZAWR
OCILPRZEDGANEKWYSIADLZPOWOZUKONIEPORZUCONESAMESZCZYPIACTRAWEC
IAGNELYPOWOLIPODBRAMEWEDWORZEPUSTOBODRZWIODGANKUZAMKNIETOZAS
ZCZEPKAMIIKOLKIEMZASZCZEPKIPRZETKNIETOPODROZNYDOFOLWARKBNIEBIEGL
SLBGZAPYTACODEMKNALWBIEGLDODOMBPRAGNALGOPOWITACDAWNODOMBNIE
WIDZIALBOWDALEKIMMIESCIEKONCZYLNABKIKONCADOCZEKALNARESZCIEWBIEG
AIOKIEMCHCIWIESCIANYSTARODAWNEOGLADACZBLEJAKOSWEZNAJOMEDAWNET
EZSAMEWIDZISPRZETYTEZSAMEOBICIAZKTOREMISIEZABAWIACLBBILODPOWICIAL
ECZMNIEJWIELKIEMNIEJPIEKNENIZSIEDAWNIEJZDALYITEZSAMEPORTRETYNASCIA
NACHWISIALYTBKOSCIBSZKOWCZAMARCEKRAKOWSKIEJZOCZYMAPODNIESIONYM
IWNIEBOMIECZOBBRACZTRZYMATAKIMBYLGDYPRZYSIEGALNASTOPNIACHOLTARZ
OWZETYMMIECZEMWYPEDZIZPOLSKITRZECHMOCARZOWALBOSAMNANIMPADNIED
ALEJWPOLSKIEJSZACIESIEDZIREJTANZALOSNYPOWOLNOSCISTRACIEWREKBTRZY
MNANOZOSTRZEMZWROCONYDOLONAAPRZEDNIMLEZYFEDONIZYWOTKATONADAL
EJJASINSKIMLODZIANPIEKNYIPOSEPNYOBOKKORSAKTOWARZYSZJEGONIEODSTE
PNYSTOJANASZANCACHPRAGINASTOSACHMOSKALISIEKACWROGOWAPRAGAJBZS
IEWKU

Przykładowy program

```
const int n = 26;
int t[n];

for (char i = 0; i < n; i++)
  t[i]=0;

ifstream szyfr ("szyfrogram.txt");
char c;

while (szyfr >> c)
  t[c - 'A']++;

szyfr.close ();
ifstream czestosc ("czestosc.txt");
char litera, klucz[n];
int ile;
```

```
while (czestosc >> litera >> ile) {
   c = 'A';
   while (t[c - 'A'] != ile)
       c++;
       klucz[c - 'A'] = litera;
   }

ofstream zapis ("zadanie7_3.txt");
szyfr.open ("szyfrogram.txt");

string s;
while (getline (szyfr, s)) {
   for (int i = 0; i < s.length(); i++)
      zapis << klucz[s[i] – 'A'];
      zapis << endl;
}
```

Zadanie 8.

W pliku `dane8.txt` zapisano ciąg x złożony z 2023 różnych liczb całkowitych $x_1, x_2, \ldots, x_{2023}$ z przedziału [1,2023], po jednej liczbie w każdym wierszu.

Napisz **programy** dające odpowiedzi do zadań 8.1–8.3.

Zadanie 8.1. (0–2)

Luką w ciągu liczbowym nazywamy wartość bezwzględną różnicy dwóch sąsiednich elementów w tym ciągu. Ile jest parzystych, a ile – nieparzystych luk w ciągu x?

Przykład:
W ciągu 2,4,10,6,8,1,3,7,9,5 jest 8 luk parzystych i 1 luka nieparzysta.

Odpowiedź:

Liczba luk parzystych _____

Liczba luk nieparzystych _____

Do oceny oddajesz:
- odpowiedź zapisaną powyżej
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

_____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2;
II.4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym;
I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy;
I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów:
c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie.

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – za poprawną odpowiedź tylko dla luk parzystych lub nieparzystych.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Parzystych 1014
Nieparzystych 1008

Fragment przykładowego programu:

```
ifstream plik("dane8.txt");
const int n = 2023;
int t[n];

int p=0, np=0;
for (int i=0; i<n; i++)
   plik >> t[i];

for (int i=1; i<n; i++)
{
   if (abs(t[i]-t[i-1])%2==0) p++;
   else np++;
}

 cout << p << " " << np << endl;
```

Zadanie 8.2. (0–2)

Podaj, ile jest nieuporządkowanych par liczb w ciągu x, tzn. takich par (x_i, x_j), że $x_i > x_j$ oraz $i < j$.

Uwaga: x_i, x_j nie muszą być sąsiednimi elementami ciągu.

Przykład:
W ciągu 2,4,10,6,8,1,3,7,9,5 jest 19 nieuporządkowanych par.

Odpowiedź:

Liczba par _____

Do oceny oddajesz:
- odpowiedź zapisaną powyżej
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

_____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2;
II.4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym;
I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy;
I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów:
c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie.

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź nieuwzględniająca warunku $i < j$.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

1037727

Fragment przykładowego programu:

```
const int n = 2023;
ifstream plik ("dane8.txt");int t[n];
for (int i = 0; i < n; i++)
   plik >> t[i];
long ile=0;
for (int i = 0; i < n-1; i++)
   for(int j=i+1; j < n; j++)
      if (t[i] > t[j]) ile++;
cout << ile << endl;
```

Zadanie 8.3. (0–2)

Podaj długość najdłuższego podciągu rosnącego w ciągu *x* złożonego z kolejnych elementów.

Przykład:
W ciągu 2,4,10,6,8,1,3,7,9,5 najdłuższy podciąg rosnący złożony z kolejnych elementów ma długość 4. Jest nim podciąg 1,3,7,9.

Odpowiedź:

Długość najdłuższego podciągu: _____

Do oceny oddajesz:

- odpowiedź zapisaną powyżej
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

_____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2;
II.4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym;
I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy;
I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów:
c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie.

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź różniąca się o 1 od poprawnej.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

6

Fragment przykładowego programu:

```
const int n = 2023;
ifstream plik ("dane8.txt");
int t[n];
for (int i = 0; i < n; i++)
   plik >> t[i];

int max = 1;
int dl = 1;
for (int i = 1; i < n; i++) {
   if (t[i – 1] <  t[i])
      dl++;
   else
      dl = 1;
   if (dl > max) {
      max = dl; dl = 1;
   }
 }
```

```
cout << max << endl;
```

Zadanie 8.4. (0–4)

Podaj, ile wynosi długość najdłuższego podciągu rosnącego w ciągu *x*.

Przykład:
Najdłuższym podciągiem rosnącym w ciągu 2,4,10,6,8,1,3,7,9,5 jest np. podciąg 2,4,6,8,9 o długości 5. Taką samą długość ma podciąg 2,4,6,7,9.

Odpowiedź:

Długość najdłuższego podciągu _____

Do oceny oddajesz:
- odpowiedź zapisaną powyżej
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

_____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
I.1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania);
objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;
II.1. projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2;
II.4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym;
I+II.1. zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy;
I+II.2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów:
c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie.

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź różniąca się o 1 od poprawnej.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

85

Fragment przykładowego programu:

```
const int n = 2023;
ifstream plik ("dane8.txt");
int t[n];

for (int i = 0; i < n; i++)
   plik >> t[i];

int dl[n]; // długość najdłuższego podciągu kończącego się na i-tym elemencie
int max = 0;

for (int i = 0; i < n; i++) {
   dl[i] = 1;    // początkowo najdłuższy podciąg składa się tylko z elementu t[i]
   // próbujemy rozszerzyć go w lewo o podciąg zakończony na j-tym elemencie
   for (int j = 0;  j < i; j++)
      if (t[j] < t[i])
         if (dl[j] + 1 > dl[i])
            dl[i] =  dl[j] + 1;
    // obliczyliśmy dl[i], sprawdzamy czy jest najlepsze:
    if (dl[i] > max)
       max = dl[i];
}
cout << max;
```

Uwaga: podane przykładowe rozwiązanie nie jest czasowo optymalne, ma koszt kwadratowy.

Zadanie 9. (0–2)

Odwrotna notacja polska.

a) Uzupełnij tabelę.

Wyrażenie zapisane w konwencjonalnej notacji algebraicznej	Wyrażenie zapisane w odwrotnej notacji polskiej
((2 + 3) * 4 + 5 * (4 – 6)) / 2	
	3 4 3 + * 1 4 3 5 + / - -

b) Uzupełnij tabelę.

Wyrażenie zapisane w odwrotnej notacji polskiej	Wartość liczbowa wyrażenia
2 3 + 5 1 + -	
3 4 5 + 4 2 - * +	

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:

2. wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: d) zamiany wyrażenia na postać w odwrotnej notacji polskiej i obliczanie jego wartości na podstawie tej postaci.

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – za każde poprawnie uzupełnione dwa wiersze tabeli a i b.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

a)

Wyrażenie zapisane w konwencjonalnej notacji algebraicznej	Wyrażenie zapisane w odwrotnej notacji polskiej
((2 + 3) * 4 + 5 * (4 - 6)) / 2	**2 3 + 4 * 5 4 6 - * + 2 /**
(3*(4+3)) -(1-(4/(3+5)))	3 4 3 + * 1 4 3 5 + / - -

b)

Wyrażenie zapisane w odwrotnej notacji polskiej	Wartość liczbowa wyrażenia
2 3 + 5 1 + -	**(2+3)-(5+1) = -1**
3 4 5 + 4 2 - * +	**3+(4+5)*(4-2) = 21**

Zadanie 10.

Zakład mleczarski Miętowa Dolina specjalizuje się w produkcji ekologicznego masła. Miętowa Dolina sprzedaje swój produkt do kilkunastu sklepów ze zdrową żywnością. Codziennie spływają zamówienia, które przesyłane są rano na linię produkcyjną oraz do działu transportu. W pliku `zamowienia.txt` zapisano datę i wielkość zamówienia (w kilogramach), które dociera rano przed rozpoczęciem produkcji i ma wpływ na produkcję oraz na transport w tym dniu. Dane w wierszach oddzielone są znakiem tabulacji.

Przykładowy fragment pliku:

data	zamówienie
02/01/2018	299
03/01/2018	43
04/01/2018	296
05/01/2018	287

Z wykorzystaniem dostępnych narzędzi informatycznych oraz danych z pliku `zamowienia.txt` rozwiąż poniższe zadania. Odpowiedzi zapisz w miejscach do tego przeznaczonych w arkuszu oraz w pliku `wyniki10.txt`, a każdą z nich poprzedź numerem odpowiedniego zadania.

Zadanie 10.1. (0–3)

Dla każdego miesiąca od stycznia 2018 do grudnia 2019 podaj sumę kilogramów zamówionego masła. Dla wykonanego zestawienia sporządź wykres kolumnowy. Pamiętaj o prawidłowym opisie osi oraz o tytule wykresu.

Do oceny oddajesz plik **z realizacją zadania** o nazwie _____

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

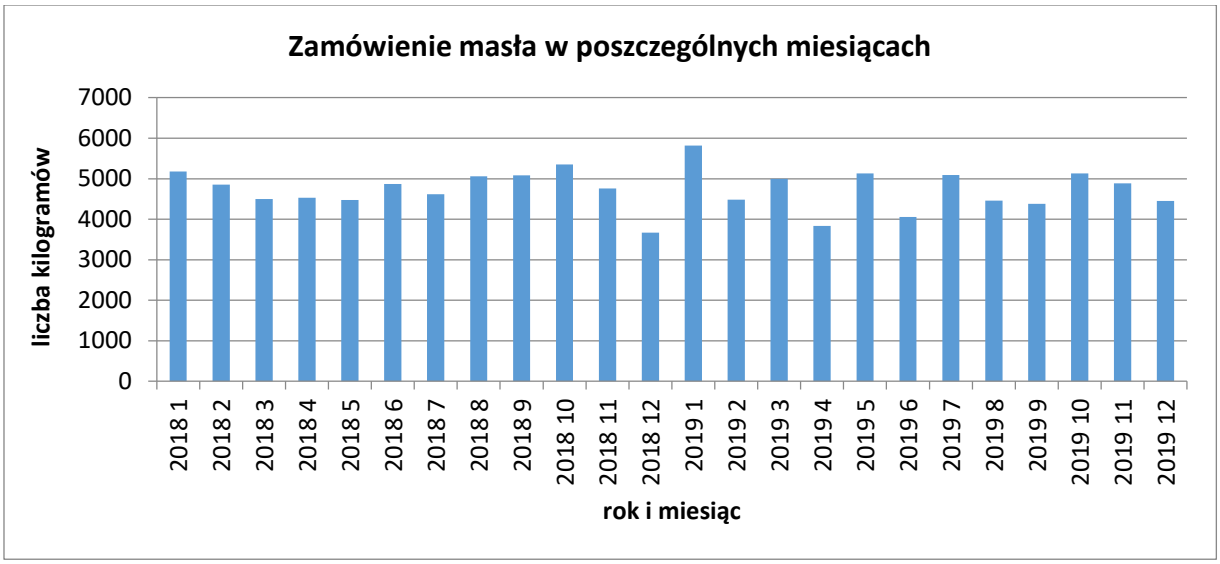
Zdający:
3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
c) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych.

Zasady oceniania

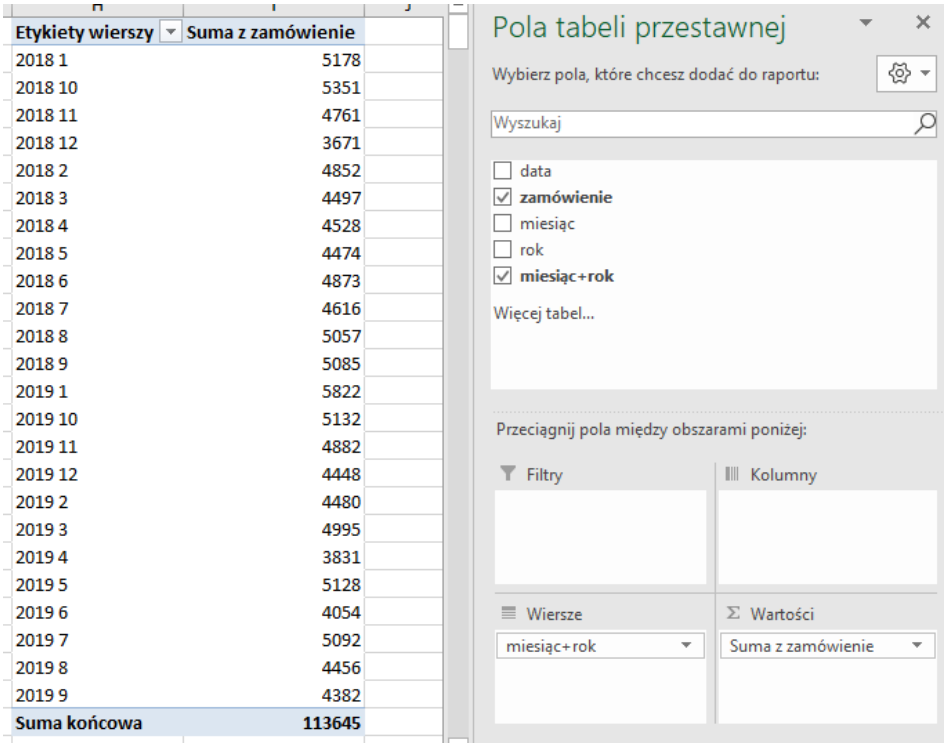
3 pkt – odpowiedź poprawna w tym:
- 2 pkt – wykres, w tym:
  - 1 pkt – odpowiedni typ wykresu i dobór danych,
  - 1 pkt – za prawidłowy opis;
- 1 pkt – zestawienie danych do wykresu.

0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie



Przykładowe wyliczenia w arkuszu kalkulacyjnym:
Aby wykonać zestawienie należy wyodrębnić z daty rok i miesiąc oraz wykonać obliczenia np. z pomocą tabeli przestawnej.



Etykiety wierszy	Suma z zamówienie
2018 1	5178
2018 10	5351
2018 11	4761
2018 12	3671
2018 2	4852
2018 3	4497
2018 4	4528
2018 5	4474
2018 6	4873
2018 7	4616
2018 8	5057
2018 9	5085
2019 1	5822
2019 10	5132
2019 11	4882
2019 12	4448
2019 2	4480
2019 3	4995
2019 4	3831
2019 5	5128
2019 6	4054
2019 7	5092
2019 8	4456
2019 9	4382
Suma końcowa	113645

Następnie wystarczy tylko wstawić wykres i odpowiednio go opisać.

Do oceny oddajemy plik arkusza kalkulacyjnego zawierający obliczenia oraz wykres.

Informacja do zadań 10.2 i 10.3.

Dział transportu realizuje dowóz za pomocą samochodu o ładowności 400 kg. Samochód wyjeżdża z zakładu dopiero wtedy, jeśli będzie wypełniony w 100%, czyli łączne zamówienie od ostatniego transportu wynosi co najmniej 400 kg. Jeśli łączne zamówienie jest niższe, to nie ma w tym dniu transportu. Jeśli łączne zamówienie jest wyższe niż 400 kg, to pozostała część zamówienia pojedzie następnym transportem (inaczej mówiąc każde zamówienie można dzielić – jeśli nie ma miejsca na całość, to wysyła się tę część, która się zmieści a reszta zamówienia jest obsługiwana w kolejnym transporcie). Jeżeli łączne zamówienie jest większe lub równe wielokrotności 400 kg, to w jednym dniu może odbyć się kilka transportów po 400 kg każdy.
Zakładamy, że każdego dnia samochód jest w stanie wykonać dowolną liczbę pełnych kursów.

Zadanie 10.2. (0–1)

Podaj liczbę dni, w których odbył się transport masła do sklepów.

Odpowiedź _____

Do oceny oddajesz plik z realizacją zadania o nazwie _____

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
c) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych;
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym.

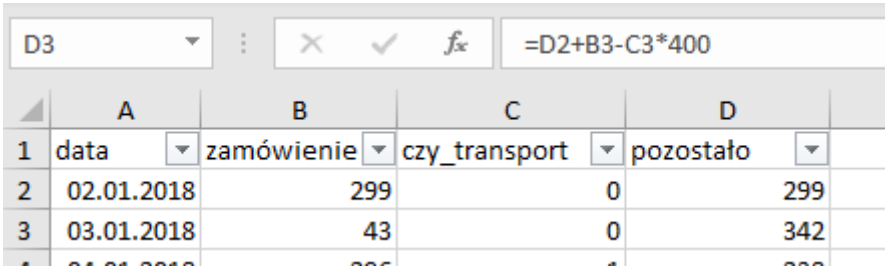
Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

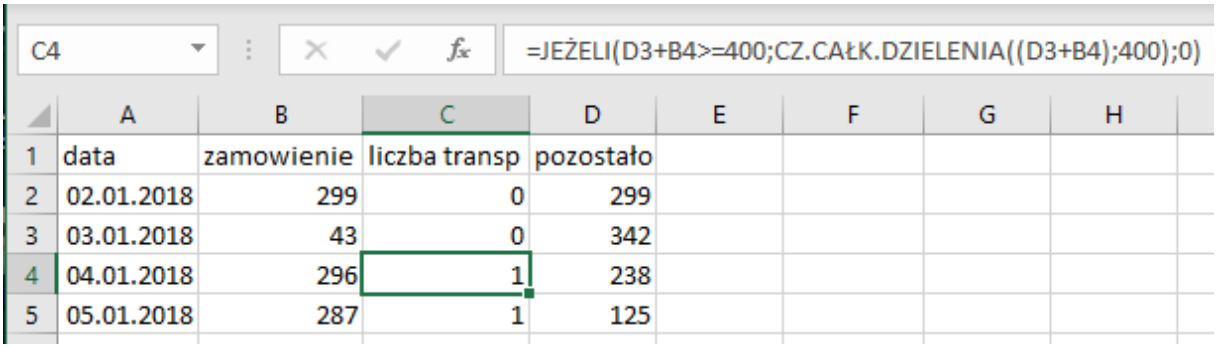
280 dni

Obliczenia:
Towar pozostały w magazynie



	A	B	C	D
1	data	zamówienie	czy_transport	pozostało
2	02.01.2018	299	0	299
3	03.01.2018	43	0	342

Czy będzie transport:



	A	B	C	D	E	F	G	H
1	data	zamowienie	liczba transp	pozostało				
2	02.01.2018	299	0	299				
3	03.01.2018	43	0	342				
4	04.01.2018	296	1	238				
5	05.01.2018	287	1	125				

Wynik otrzymujemy po zliczeniu np. funkcją *Licz.Jeżeli* dni, w których liczba transportów jest większa od 0.

Zadanie 10.3. (0–2)

Podaj daty, kiedy samochód wykonał co najmniej dwie dostawy w tym samym dniu.

Odpowiedź _____

Do oceny oddajesz plik z realizacją zadania o nazwie _____

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
c) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych;
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym.

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – brak jednej daty.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

05.07.2018
16.07.2018
26.03.2019
27.11.2019

Uwaga: Przy poprzednich obliczeniach wystarczy ukryć za pomocą filtru dni, kiedy nie było transportu lub był tylko jeden.

Informacja do zadań 10.4 i 10.5.

Standardowo linia produkcyjna ma wydajność 200 kg masła dziennie. W szczególnym przypadku, gdy zamówienie jest większe niż połowa porannej zawartości magazynu, produkcja w tym dniu wzrasta o 30%. Produkcja jest redukowana o 20% zawsze, gdy poranny stan magazynu jest większy niż 1500 kg (niezależnie od wielkości zamówienia).

Załóż, że w dniu 2.01.2018 roku rano w magazynie znajdowało się 1000 kg masła i biorąc pod uwagę zamówienia, opisany cykl produkcyjny oraz cykl transportowy, wykonaj symulację porannej zawartości magazynu Miętowej Doliny w okresie od 2 stycznia 2018 do 31 grudnia 2019 r.

Zadanie 10.4. (0–2)

Znajdź najdłuższy okres stabilizacji wielkości produkcji, czyli kolejne dni, w których produkcja masła była taka sama. Podaj początek i długość tego okresu.

Odpowiedź _____

Do oceny oddajesz plik z realizacją zadania o nazwie _____

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
c) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych;
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym.

Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

38 dni, początek 26.04.2018

Obliczenie wartości produkcji

F3 =JEŻELI(E3>1500;160;JEŻELI(B3>E3/2;260;200))

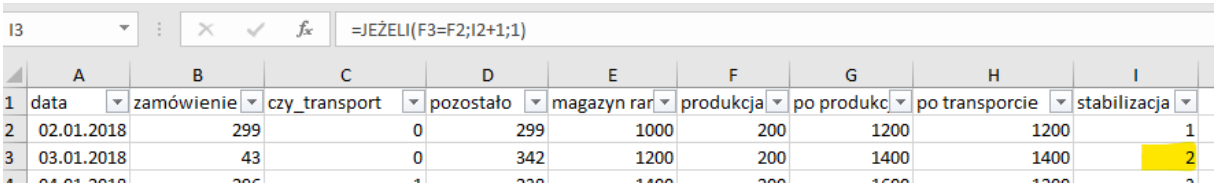
	A	B	C	D	E	F	G	H	I
1	data	zamówienie	czy_transport	pozostało	magazyn rar	produkcja	po produkc	po transporcie	stabilizacja
2	02.01.2018	299	0	299	1000	200	1200	1200	1
3	03.01.2018	43	0	342	1200	200	1400	1400	2

Stan magazynu po transporcie (jeśli był tego dnia)

H3 =G3-C3*400

	A	B	C	D	E	F	G	H	I
1	data	zamówienie	czy_transport	pozostało	magazyn rar	produkcja	po produkc	po transporcie	stabilizacja
2	02.01.2018	299	0	299	1000	200	1200	1200	1
3	03.01.2018	43	0	342	1200	200	1400	1400	2

Informacja, czy nastąpiła stabilizacja wielkości produkcji (każdy następny dzień stabilizacji jest zliczany).



I3 =JEŻELI(F3=F2;I2+1;1)

	A	B	C	D	E	F	G	H	I
1	data	zamówienie	czy_transport	pozostało	magazyn rar	produkcja	po produkc	po transporcie	stabilizacja
2	02.01.2018	299	0	299	1000	200	1200	1200	1
3	03.01.2018	43	0	342	1200	200	1400	1400	2

Następnie wystarczy w kolumnie I znaleźć maksymalną wartość i pobrać odpowiadającą jej datę początku szukanego okresu.

Zadanie 10.5. (0–2)

Jaki był najniższy i jaki najwyższy stan magazynu w opisanym w zadaniu przedziale czasowym?

Odpowiedź _____

Do oceny oddajesz plik z realizacją zadania o nazwie _____

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
c) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych;
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym.

Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Odpowiedź
min 20
max 1700

Uwaga: Wystarczy policzyć minimum i maksimum z odpowiedniej kolumny (w naszym przypadku – H). Wszystkie pozostałe obliczenia wykonano na potrzeby zadania 10.4.

Zadanie 11.

W pliku tekstowym `gaz.txt` w kolejnych kolumnach zapisano datę odczytu i wskazania licznika poboru gazu (w metrach sześciennych) w domu państwa Ciepłolubnych z okresu od 1.01.2002 do 31.12.2018. Wskazania licznika odczytywano każdego ostatniego dnia miesiąca.

Z wykorzystaniem dostępnych narzędzi informatycznych oraz danych z pliku `gaz.txt` rozwiąż poniższe zadania. Odpowiedzi zapisz w miejscach do tego przeznaczonych w arkuszu oraz w pliku `wyniki11.txt`, a każdą z nich poprzedź numerem odpowiedniego zadania.

Zadanie 11.1. (0–2)

Zakładamy, że 31.12.2001 r. licznik wskazywał 2083. Podaj daty odczytów wskazań licznika w miesiącach, dla których średnie zużycie gazu w metrach sześciennych na dobę było większe niż 12 m^3.

Odpowiedź _____

Do oceny oddajesz plik z realizacją zadania o nazwie _____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
c) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych;
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym.

Zasady oceniania

2 pkt – pełna odpowiedź poprawna.
1 pkt – odpowiedź niepełna (z pominięciem jednej daty).
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Odpowiedź:
28.02.2010
28.02.2011
28.02.2012
28.02.2013
28.02.2015

Przykładowe obliczenia:

Wykonujemy obliczenia dla każdego miesiąca.

E3 =D3/C3

	A	B	C	D	E
1	Data odczy	Odczyt licznil	liczba d	zuzycie	średnie na dobę
2	31.01.2002	2283	31	200	6,451612903
3	28.02.2002	2518	28	235	8,392857143
4	31.03.2002	2696	31	178	5,741935484
5	30.04.2002	2857	30	161	5,366666667
6	31.05.2002	2917	31	60	1,935483871
7	30.06.2002	2952	30	35	1,166666667
8	31.07.2002	2979	31	27	0,870967742
9	31.08.2002	3009	31	30	0,967741935
10	30.09.2002	3040	30	31	1,033333333
11	31.10.2002	3116	31	76	2,451612903

I filtrujemy dane.

Data odczy	Odczyt licznil	liczba d	zuzycie	średnie na dob
28.02.2010	16792	28	361	12,89285714
28.02.2011	19057	28	377	13,46428571
28.02.2012	21000	28	347	12,39285714
28.02.2013	23173	28	362	12,92857143
28.02.2015	27156	28	348	12,42857143

Zadanie 11.2. (0–2)

Wygeneruj zestawienie średniego zużycia gazu w każdym z 12 miesięcy (styczeń, luty, marzec itd.) w podanym przedziale czasowym 2002–2018. Dla swojego zestawienia utwórz wykres.

Do oceny oddajesz:
plik z realizacją zadania o nazwie _____

oraz wykres zapisany w pliku o nazwie _____

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:

c) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych;
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym.

Zasady oceniania

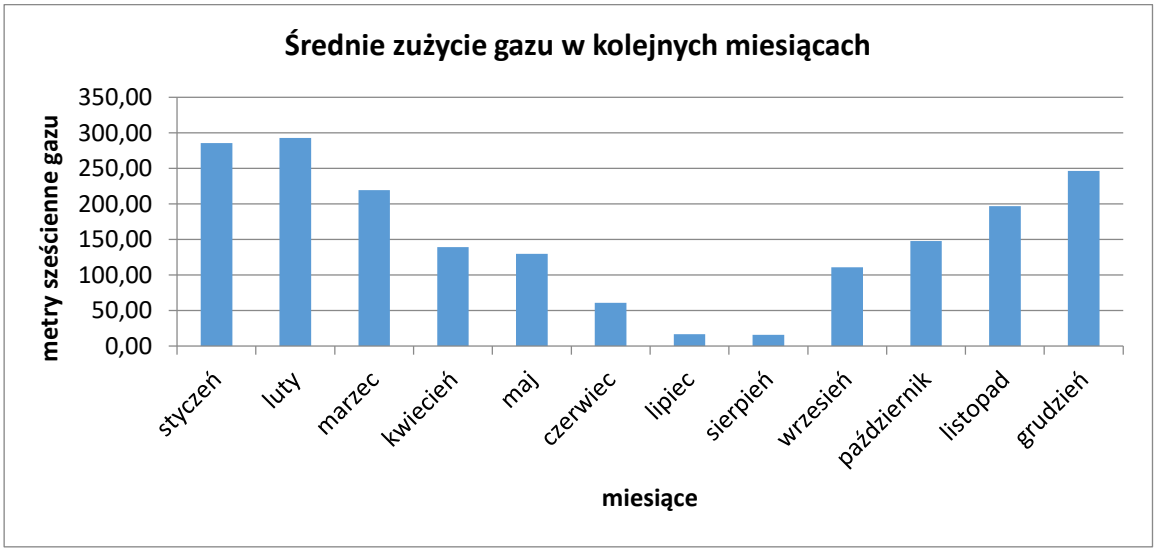
2 pkt – poprawny wykres.
1 pkt – utworzenie zestawienia do wykresu.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

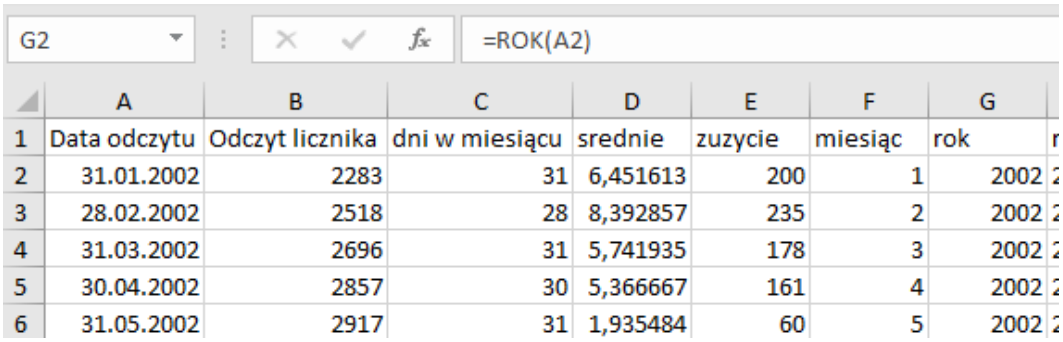
Odpowiedź

miesiąc	średnie zużycie
styczeń	285,47
luty	292,71
marzec	219,24
kwiecień	139,24
maj	129,76
czerwiec	61,00
lipiec	16,65
sierpień	15,94
wrzesień	110,71
październik	147,71
listopad	197,12
grudzień	246,29

Wykres



Wykonanie obliczeń:



	A	B	C	D	E	F	G
1	Data odczytu	Odczyt licznika	dni w miesiącu	srednie	zuzycie	miesiąc	rok
2	31.01.2002	2283	31	6,451613	200	1	2002
3	28.02.2002	2518	28	8,392857	235	2	2002
4	31.03.2002	2696	31	5,741935	178	3	2002
5	30.04.2002	2857	30	5,366667	161	4	2002
6	31.05.2002	2917	31	1,935484	60	5	2002

Wykonanie zestawienia za pomocą tabeli przestawnej lub funkcji średnia.jeżeli

miesiąc	**średnie zużycie**
styczeń	285,47
luty	292,71
marzec	219,24
kwiecień	139,24
maj	129,76
czerwiec	61,00
lipiec	16,65
sierpień	15,94
wrzesień	110,71
październik	147,71
listopad	197,12
grudzień	246,29

Zadanie 11.3. (0–2)

Opłata za gaz składa się z opłaty taryfowej (płacimy stałą miesięczną opłatę za sklasyfikowanie do danej taryfy) oraz z opłaty za zużyty gaz zależnej od ceny gazu w danym roku i liczby metrów sześciennych zużytego gazu.

Rodzaj taryfy zależy od ilości pobieranego miesięcznie gazu.

Nazwa taryfy	Pobór gazu	Opłata taryfowa
W1	poniżej 100 m^3	70 zł
W2	od 100 m^3 do 200 m^3	90 zł
W3	powyżej 200 m^3	120 zł

Cena gazu zapisana została także w pliku tekstowym `cena_gazu.txt`.

Na podstawie podanych wyżej informacji utwórz zestawienie łącznych rocznych opłat za gaz.

Do oceny oddajesz zestawienie zapisane w pliku o nazwie _____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:

3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:

c) gromadzi dane pochodzące z różnych źródeł w tabeli arkusza kalkulacyjnego, korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych, filtruje dane według kilku kryteriów, dobiera odpowiednie wykresy do zaprezentowania danych, analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych;

4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym.

Rozwiązanie

Rok	Suma z opłata miesięczna
2002	2 286,90 zł
2003	2 988,33 zł
2004	2 822,26 zł
2005	2 958,44 zł
2006	3 131,46 zł
2007	2 799,12 zł
2008	3 171,30 zł
2009	3 021,02 zł
2010	3 706,35 zł
2011	3 422,18 zł
2012	3 867,39 zł
2013	3 425,02 zł
2014	3 738,24 zł
2015	3 337,60 zł
2016	3 575,48 zł
2017	3 262,91 zł
2018	3 131,78 zł
Suma końcowa	**54 645,78 zł**

Przykładowe obliczenia z zastosowaniem tabeli przestawnej (fragment):

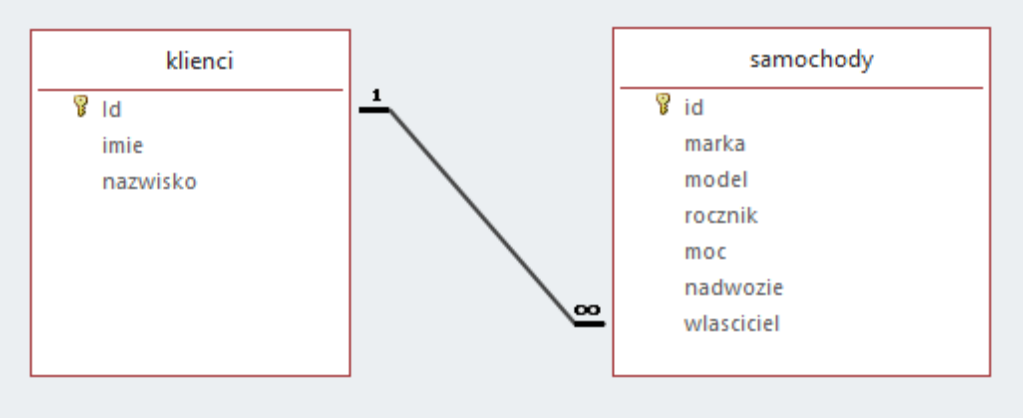
G2 =JEŻELI(D2<100;70;JEŻELI(D2<=200;90;120))

rozw_gaz.xlsx

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Data odczytu	Odczyt licznika	liczba dni	zuzycie	rok	cena	opłata taryfowa	opłata miesięczna			rok	cena		Etykiety wierszy	Suma z opłata miesięczna
2	31.01.2002	2283	31	200	2002	0,99	90	288			2001	0,97		2002	2 286,90 zł
3	28.02.2002	2518	28	235	2002	0,99	120	352,65			2002	0,99		2003	2 988,33 zł
4	31.03.2002	2696	31	178	2002	0,99	90	266,22			2003	0,99		2004	2 822,26 zł
5	30.04.2002	2857	30	161	2002	0,99	90	249,39			2004	0,98		2005	2 958,44 zł
6	31.05.2002	2917	31	60	2002	0,99	70	129,4			2005	1,02		2006	3 131,46 zł
7	30.06.2002	2952	30	35	2002	0,99	70	104,65			2006	1,02		2007	2 799,12 zł
8	31.07.2002	2979	31	27	2002	0,99	70	96,73			2007	1,04		2008	3 171,30 zł
9	31.08.2002	3009	31	30	2002	0,99	70	99,7			2008	1,05		2009	3 021,02 zł
10	30.09.2002	3040	30	31	2002	0,99	70	100,69			2009	1,07		2010	3 706,35 zł
11	31.10.2002	3116	31	76	2002	0,99	70	145,24			2010	1,11		2011	3 422,18 zł
12	30.11.2002	3222	30	106	2002	0,99	90	194,94			2011	1,18		2012	3 867,39 zł
13	31.12.2002	3393	31	171	2002	0,99	90	259,29			2012	1,23		2013	3 425,02 zł
14	31.01.2003	3613	31	220	2003	0,99	120	337,8			2013	1,23		2014	3 738,24 zł
15	28.02.2003	3891	28	278	2003	0,99	120	395,22			2014	1,23		2015	3 337,60 zł
16	31.03.2003	4151	31	260	2003	0,99	120	377,4			2015	1,2		2016	3 575,48 zł
17	30.04.2003	4401	30	250	2003	0,99	120	367,5			2016	1,21		2017	3 262,91 zł
18	31.05.2003	4553	31	152	2003	0,99	90	240,48			2017	1,21		2018	3 131,78 zł
19	30.06.2003	4639	30	86	2003	0,99	70	155,14			2018	1,22		Suma końcowa	54 645,78 zł

Zadanie 12.

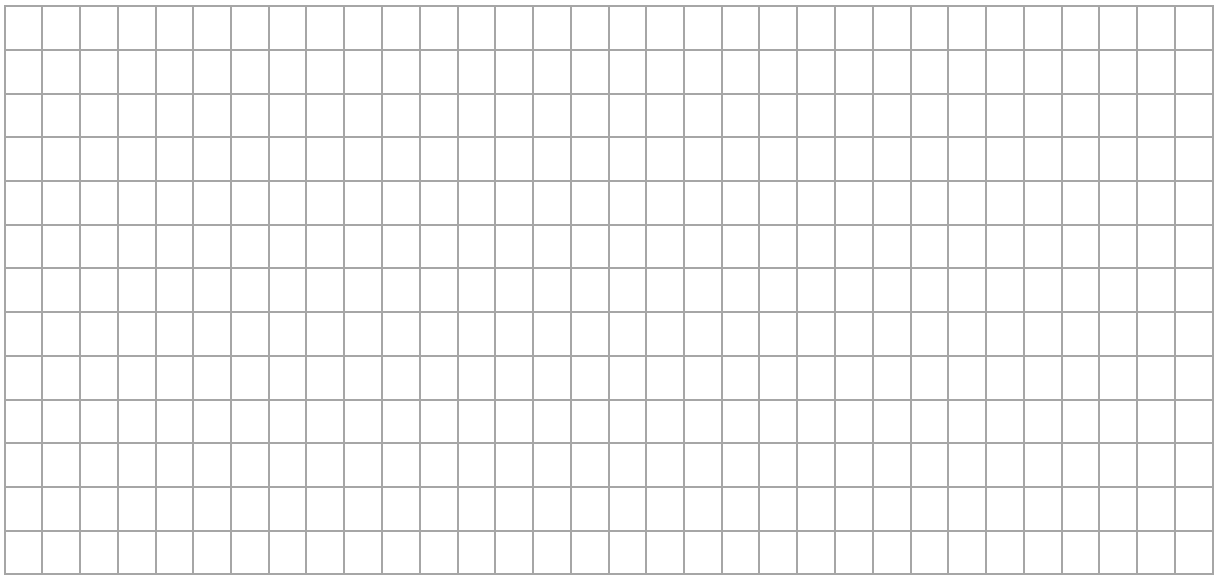
Dane są dwie tabele bazy danych: *klienci* i *samochody* połączone relacją jeden do wielu (patrz rysunek):



W tabelach przechowywane są informacje zgodne z nazwami pól (np. „marka” – marka samochodu). Pole *id* w tabeli *klienci* oznacza identyfikator klienta i jest kluczem głównym w tej tabeli. Pole *id* w tabeli *samochody* oznacza identyfikator samochodu. Pole o nazwie *wlasciciel* w tabeli samochody jest kluczem obcym w tej tabeli i oznacza identyfikator klienta, który jest właścicielem samochodu. Na potrzeby zadania zakładamy, że każdy samochód ma tylko jednego właściciela, natomiast każda osoba może być właścicielem jednego lub więcej niż jednego samochodu.

Zadanie 12.1. (0–1)

Zapisz **w języku SQL** zapytanie, które da w wyniku listę klientów (z id, imionami i nazwiskami) uporządkowaną alfabetycznie według nazwiska.



Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:
d) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie.

Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

```
SELECT *
FROM klienci
ORDER BY klienci.nazwisko;
```

Zadanie 12.2. (0–2)

Zapisz **w języku SQL** zapytanie, które da w wyniku listę marek i modeli samochodów, dla których *rocznik* zawiera się w przedziale [2010,2020] oraz *moc* jest większa niż 100 KM. Listę posortuj niemalejąco według nazwy marki i modelu.

Uwaga: Każda para *model* i *marka* w zestawieniu może pojawić się tylko raz (nawet jeśli jest kilka samochodów o tej samej marce i modelu spełniających warunki zadania).

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:
d) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie.

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź, w której zdający nie uwzględnia, że każda para *marka* i *model* może pojawić się w zestawieniu tylko raz (brak opcji DISTINCT).
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

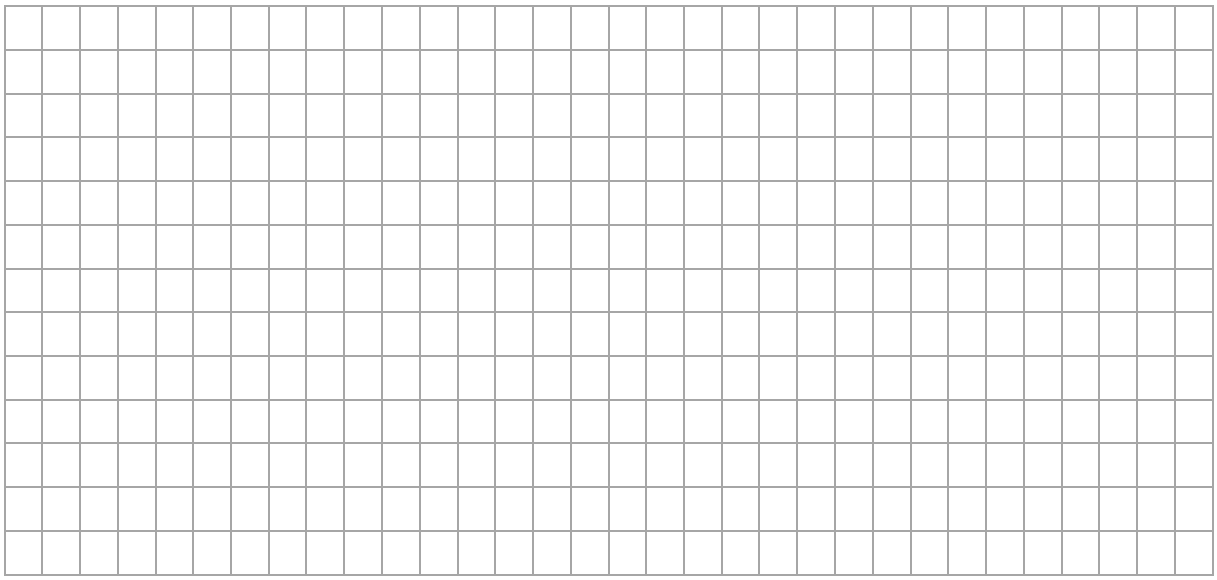
Przykładowe rozwiązanie

```
SELECT DISTINCT samochody.marka, samochody.model
FROM samochody
WHERE (samochody.moc>100) AND (samochody.rocznik Between 2010 And 2020)
ORDER BY samochody.marka, samochody.model;
```

Zadanie 12.3. (0–2)

Zapisz **w języku SQL** zapytanie, które da w wyniku listę klientów z ich id, imionami, nazwiskami oraz liczbą samochodów posiadanych przez każdego z nich. Lista powinna być posortowana nierosnąco według liczby samochodów.

Uwaga: uwzględnij, że niektórzy klienci mogą obecnie nie mieć żadnego samochodu (wtedy trzeba wypisać ich z liczbą samochodów zero)



Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:

4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:

d) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie.

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź, w której zdający nie uwzględnia, że klient może nie posiadać samochodu (INNER JOIN zamiast LEFT JOIN).
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Przykładowe rozwiązanie

```
SELECT klienci.imie, klienci.nazwisko, Count(samochody.id) AS ile
FROM klienci LEFT JOIN samochody ON klienci.Id = samochody.wlasciciel
GROUP BY klienci.imie, klienci.nazwisko
ORDER BY Count(samochody.id) DESC;
```

Zadanie 13.

Pracownicy firmy MATRIX zostali poproszeni o rejestrowanie swoich aktywności w celu wypracowania najlepszej umowy z firmą oferującą karty na wejścia do obiektów sportowych. Pracownicy MATRIX w okresie od 1.07.2018 do 31.12.2018, rejestrowali wszystkie swoje aktywności. Dane zapisano w trzech plikach tekstowych o nazwach: `pracownicy.txt`, `rodzaj_aktywnosci.txt`, `rejestr_aktywnosci.txt`. Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w wierszach rozdzielono znakami tabulacji.

Plik o nazwie `pracownicy.txt` zawiera informacje o 199 pracownikach MATRIX: PESEL pracownika (PESEL), nazwisko (nazwisko), imię (imie) oraz miejscowość, w której mieszka (miejscowosc).

Przykład:

PESEL	Nazwisko	Imie	Miejscowosc
99121573571	Abrich	Wojciech	Zory
77022270175	Adamczyk	Janusz	Gliwice
81071092233	Andrzejczyk	Piotr	Gliwice

Plik o nazwie `rodzaj_aktywnosci.txt` zawiera zestawienie aktywności wybieranych przez pracowników. W każdym wierszu znajduje się: identyfikator aktywności (ID_aktywnosci) oraz odpowiadająca mu nazwa aktywności (nazwa_aktywnosci)

Przykład:

ID_aktywnosci	nazwa_aktywnosci
1	aqua aerobik
2	boks
3	cycling

Plik o nazwie `rejestr_aktywnosci.txt` zawiera informacje o zrealizowanych aktywnościach. W każdym wierszu pliku znajdują się: data realizacji aktywności (data), godzina rozpoczęcia aktywności (godzina_rozp), czas trwania (Godzina_zak), identyfikator aktywności (ID_aktywnosci) oraz PESEL osoby, realizującej aktywność (PESEL).

Przykład:

Data	Godzina_rozp	Godzina_zak	ID_aktywnosci	PESEL
01/07/2018	17:00:00	17:45:00	17	01301144688
01/07/2018	20:00:00	21:15:00	22	84061632305
01/07/2018	16:00:00	16:45:00	16	82092387166

Z wykorzystaniem danych zawartych w podanych plikach oraz dostępnych narzędzi informatycznych wykonaj poniższe zadania. Każdą odpowiedź umieść w miejscu na to przeznaczonym w arkuszu oraz w pliku `wyniki13.txt` (i poprzedź ją oznaczeniem odpowiedniego zadania od 13.1. do 13.5.).

Do oceny oddajesz także pliki z realizacją każdego zadań (zawierający obliczenia, kwerendy lub napisane programy).

Zadanie 13.1. (0–2)

Podaj nazwy pięciu aktywności najczęściej realizowanych przez pracowników oraz dla każdej z nich zapisz liczbę realizacji tej aktywności.

Odpowiedź _____

Plik z realizacją zadania o nazwie _____

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:

3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
d) wyszukuje informacje, korzystając z bazy danych opartej na co najmniej dwóch tabelach, definiuje relacje, stosuje filtrowanie, formułuje kwerendy, tworzy i modyfikuje formularze, drukuje raporty;
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:
d) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie.

Zasady oceniania

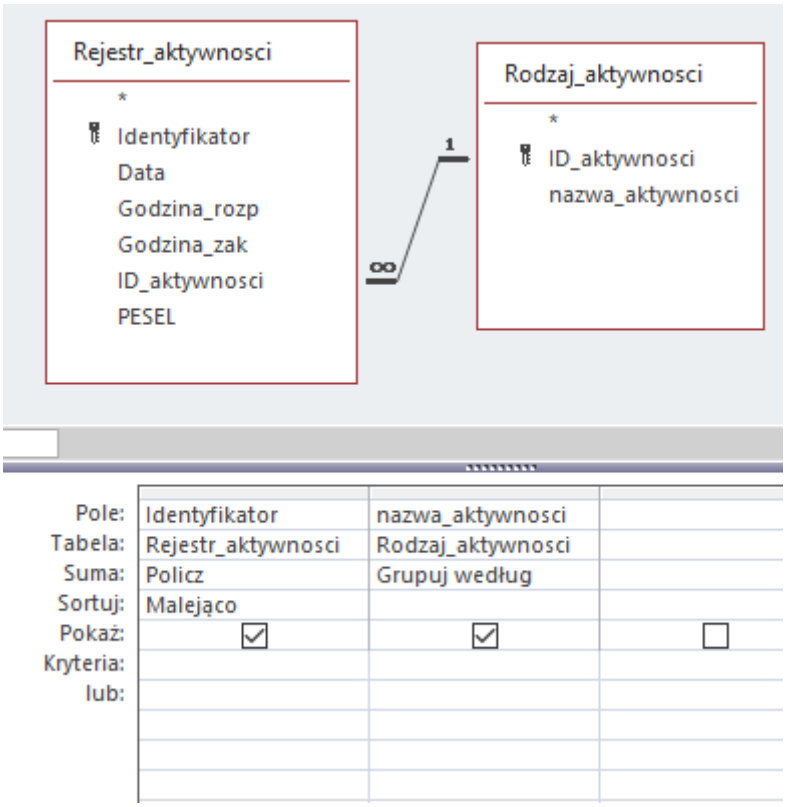
2 pkt – pełna odpowiedź poprawna.
1 pkt – podanie nazw aktywności bez liczby realizacji tych aktywności lub pominięcie albo podanie niepoprawnej nazwy jednej z aktywności.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Odpowiedź

silownia	193
scianka wspinaczkowa	185
sztuki walki	114
squash	102
nordic walking	101

Przykładowe wykonanie:




```
SELECT
Rodzaj_aktywnosci.nazwa_aktywnosci, Count(Rejestr_aktywnosci.Identyfikator) AS
PoliczOfIdentyfikator
FROM Rodzaj_aktywnosci INNER JOIN Rejestr_aktywnosci ON
Rodzaj_aktywnosci.ID_aktywnosci = Rejestr_aktywnosci.ID_aktywnosci
GROUP BY Rodzaj_aktywnosci.nazwa_aktywnosci
ORDER BY Count(Rejestr_aktywnosci.Identyfikator) DESC;
```

Zadanie 13.2. (0–1)

Podaj łączną liczbę godzin spędzonych na ściance wspinaczkowej przez wszystkich pracowników zamieszkałych w Gliwicach. Jako wynik podaj jedną liczbę. Jeśli wynikowa liczba godzin nie jest liczbą całkowitą, to minuty ostatniej, niepełnej godziny podaj w postaci części ułamkowej wyniku (np. 2 godziny i czterdzieści pięć minut to 2.75).

Odpowiedź _____

Plik z realizacją zadania o nazwie _____

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
d) wyszukuje informacje, korzystając z bazy danych opartej na co najmniej dwóch tabelach, definiuje relacje, stosuje filtrowanie, formułuje kwerendy, tworzy i modyfikuje formularze, drukuje raporty;
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:
d) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie.

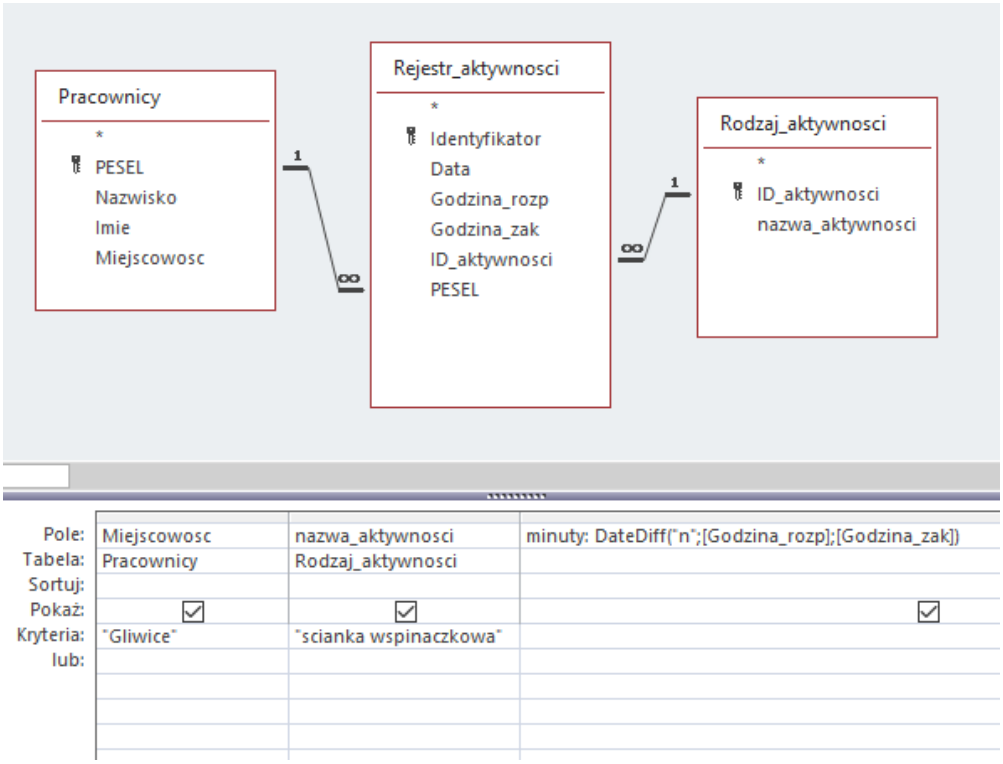
Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Odpowiedź
183h

Wybieramy dane konieczne do realizacji zadania i zliczamy łączny czas spędzany przez mieszkańców Gliwic na ściance wspinaczkowej np. w minutach.




```
SELECT
Pracownicy.Miejscowosc,Rodzaj_aktywnosci.nazwa_aktywnosci,
DateDiff("n",[Godzina_rozp],[Godzina_zak]) AS minuty
FROM Rodzaj_aktywnosci INNER JOIN (Pracownicy INNER JOIN Rejestr_aktywnosci
ON Pracownicy.PESEL = Rejestr_aktywnosci.PESEL) ON Rodzaj_aktywnosci.ID_aktywnosci
= Rejestr_aktywnosci.ID_aktywnosci
WHERE(((Pracownicy.Miejscowosc)="Gliwice")
AND ((Rodzaj_aktywnosci.nazwa_aktywnosci)="scianka wspinaczkowa"));
```

Uwaga: zamiast użycia funkcji DateDiff można także skorzystać ze zwykłego odejmowania – otrzymamy wtedy wynik liczony w dobach.

Następnie zliczamy sumę minut (10980) i wynik dzielimy przez 60.

Zadanie 13.3. (0–2)

Podaj imiona i nazwiska osób, które uprawiały co najmniej dwie aktywności w ciągu tego samego dnia. Nazwiska posortuj alfabetycznie.

Odpowiedź _____

Plik z realizacją zadania o nazwie _____

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
d) wyszukuje informacje, korzystając z bazy danych opartej na co najmniej dwóch tabelach, definiuje relacje, stosuje filtrowanie, formułuje kwerendy, tworzy i modyfikuje formularze, drukuje raporty;
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:
d) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie.

Zasady oceniania

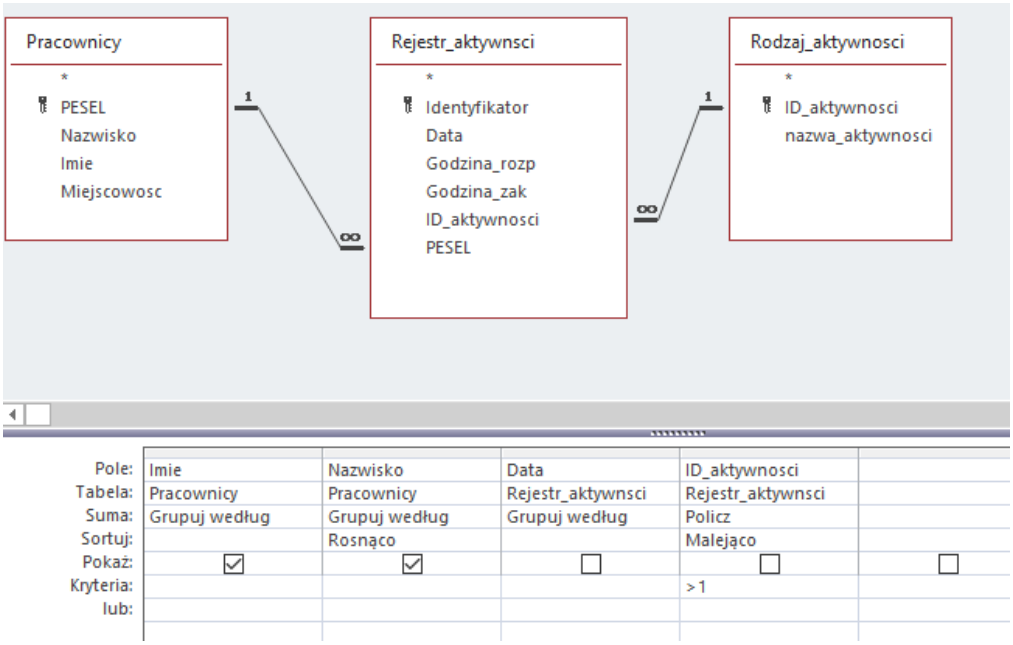
2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź niepełna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Odpowiedź:

Justyna	Andrzejewska
Jerzy	Durczok
Karolina	Gawron
Amelia	Herman
Rozalia	Jacak
Andrzej	Kowalski
Krystyna	Makowska
Justyna	Myszkowska
Bogumil	Nowak
Marek	Stasicki
Feliks	Synowski
Anna	Tobera
Marzena	Warszawska
Patrycja	Wilczynska

Przykładowe wykonanie:




```
SELECT Pracownicy.Imie, Pracownicy.Nazwisko
FROM Rodzaj_aktywnosci INNER JOIN (Pracownicy INNER JOIN Rejestr_aktywnosci ON
Pracownicy.PESEL = Rejestr_aktywnosci.PESEL) ON Rodzaj_aktywnosci.ID_aktywnosci =
Rejestr_aktywnosci.ID_aktywnosci
GROUP BY Pracownicy.Imie, Pracownicy.Nazwisko, Rejestr_aktywnosci.Data
HAVING (((Count(Rejestr_aktywnosci.ID_aktywnosci))>1))
ORDER BY Pracownicy.Nazwisko, Count(Rejestr_aktywnosci.ID_aktywnosci) DESC;
```

Zadanie 13.4. (0–2)

Pierwsze dwie cyfry numeru PESEL oznaczają ostatnie dwie cyfry roku urodzenia, natomiast przedostatnia cyfra koduje płeć: jeśli jest parzysta, to jest to PESEL kobiety, jeśli nieparzysta – to mężczyzny. W PESEL-u osób urodzonych po roku 1999 cyfry trzecia i czwarta oznaczają sumę numeru miesiąca urodzenia i liczby 20.

Przykład:
Osoba o numerze PESEL 85040186862 urodziła się 1 kwietnia 1985 roku i jest kobietą, a o numerze PESEL 01271948491 – 19 lipca 2001 roku i jest mężczyzną.

Podaj, z dokładnością do dwóch miejsc po przecinku, średni wiek kobiet i średni wiek mężczyzn w roku 2018, którzy kiedykolwiek w okresie badawczym korzystali z zajęć rolkowych (rolki).

Odpowiedź _____

Plik z realizacją zadania o nazwie _____

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
d) wyszukuje informacje, korzystając z bazy danych opartej na co najmniej dwóch tabelach, definiuje relacje, stosuje filtrowanie, formułuje kwerendy, tworzy i modyfikuje formularze, drukuje raporty;
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:
d) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie.

Zasady oceniania

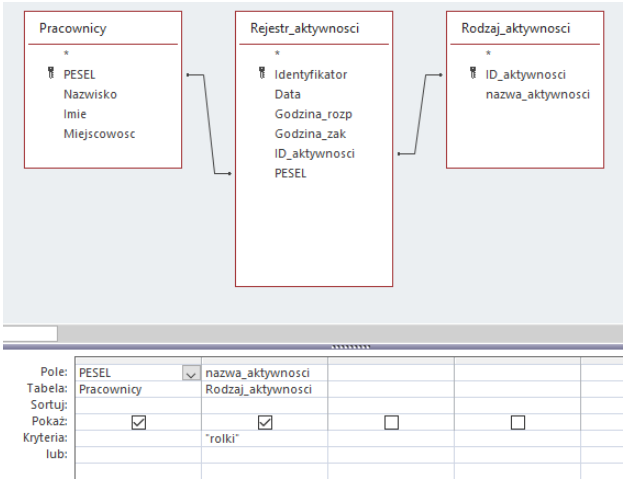
2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź niepełna (np. tylko dla kobiet lub tylko dla mężczyzn)
ALBO
odpowiedź otrzymana w przypadku kilkukrotnego uwzględnienia tej samej osoby – braku grupowania wyników.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Odpowiedź:

kobiety	34.40
mężczyźni	31.24

Przykładowe wykonanie:
Zadanie można wykonać na wiele sposobów.
Najpierw możemy wyszukać pracowników, którzy korzystali z zajęć rolkowych. Np.



```
SELECT Pracownicy.PESEL, Rodzaj_aktywnosci.nazwa_aktywnosci
FROM
(Pracownicy INNER JOIN Rejestr_aktywnosci
ON Pracownicy.PESEL = Rejestr_aktywnosci.PESEL) INNER JOIN Rodzaj_aktywnosci ON
Rejestr_aktywnosci.ID_aktywnosci = Rodzaj_aktywnosci.ID_aktywnosci
WHERE (((Rodzaj_aktywnosci.nazwa_aktywnosci)="rolki"));
```

Następnie obliczamy średni wiek.
Obliczenia można wykonać np. za pomocą arkusza kalkulacyjnego.

F2 =ŚREDNIA.JEŻELI(B2:B80;"k";E2:E80)

	A	B	C	D	E	F	G
1	PESEL	kobieta/	2 cyfry	rokur	wiek		
2	00220396736	m	00	2000	18	34,40476	
3	00221216048	k	00	2000	18	31,24324	
4	00312447739	m	00	2000	18		
5	01230712857	m	01	2001	17		

Te same obliczenia można oczywiście wykonać bezpośrednio w narzędziu bazodanowym np.:

pierwsze zapytanie dające tabelę wiek i płeć:

```
SELECT Avg(2018-
(Left([Pracownicy]![PESEL],2)+IIf((Mid([Pracownicy]![PESEL],3,2)<20),1900,2000))) AS
wiek2018, Mid([pracownicy]![pesel],10,1) Mod 2 AS płeć
FROM Rodzaj_aktywnosci INNER JOIN (Pracownicy INNER JOIN Rejestr_aktywnosci ON
Pracownicy.PESEL = Rejestr_aktywnsci.PESEL) ON Rodzaj_aktywnosci.ID_aktywnosci =
Rejestr_aktywnosci.ID_aktywnosci
WHERE (((Rodzaj_aktywnosci.nazwa_aktywnosci)="rolki"))
GROUP BY Pracownicy.PESEL;
```

i drugie podające obie średnie naraz (zakładamy, że to pierwsze nazywa sie 13_4_pom)

```
SELECT Avg([13_4_pom].wiek2018) AS AvgOfwiek2018, [13_4_pom].płeć
FROM 13_4_pom
GROUP BY [13_4_pom].płeć;
```

Zadanie 13.5. (0–2)

Podaj uporządkowany rosnąco zestaw numerów PESEL osób, które nigdy nie były na siłowni ani na jodze, ale przynajmniej raz korzystały z masażu.

Odpowiedź _____

Plik z realizacją zadania o nazwie _____

Wymaganie ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Wymaganie szczegółowe

Zdający:
3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
d) wyszukuje informacje, korzystając z bazy danych opartej na co najmniej dwóch tabelach, definiuje relacje, stosuje filtrowanie, formułuje kwerendy, tworzy i modyfikuje formularze, drukuje raporty;
4. przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:
d) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie.

Zasady oceniania

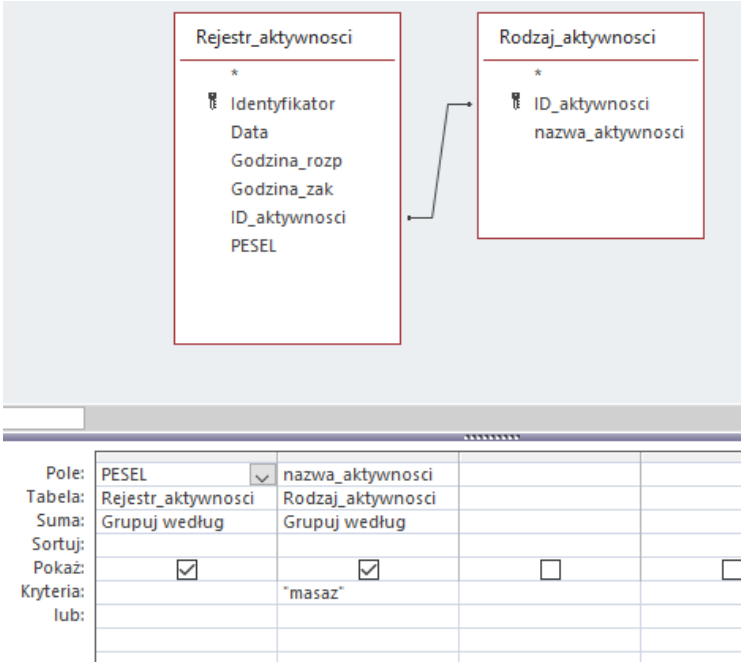
2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź niepełna (np. tylko dla kobiet lub tylko dla mężczyzn).
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Odpowiedź:
59121254845
61120859464
62011052982
67101888089
75040642244
78033117532
80052785782
81041832832
83070774646
87062086333
88060633398
89021968180
90071262308
93102278884
94032111339
97052873850
98072258726
99071947484

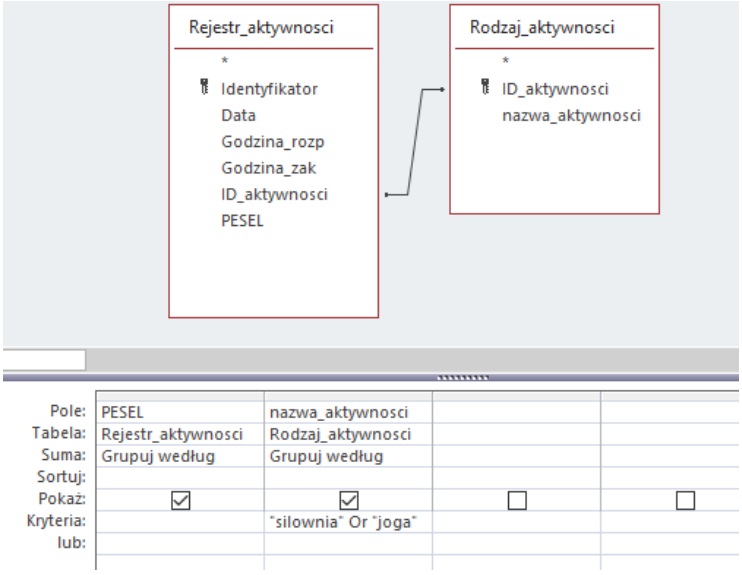
Przykładowe wykonanie:

Najpierw wyszukujemy PESELE osób, które korzystały z masażu.



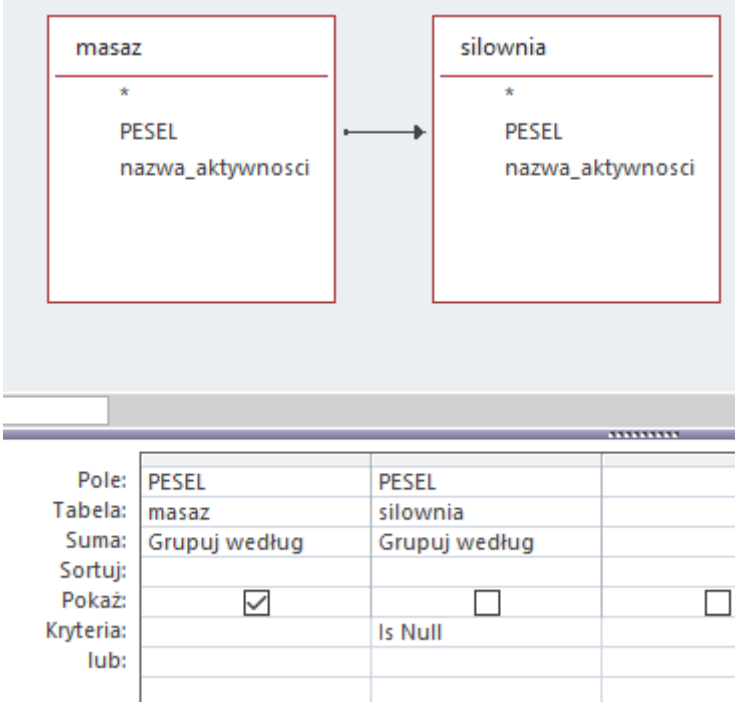

```
SELECT Rejestr_aktywnosci.PESEL, Rodzaj_aktywnosci.nazwa_aktywnosci
FROM Rejestr_aktywnosci
INNER JOIN Rodzaj_aktywnosci
ON
Rejestr_aktywnosci.ID_aktywnosci = Rodzaj_aktywnosci.ID_aktywnosci
GROUP BY Rejestr_aktywnosci.PESEL, Rodzaj_aktywnosci.nazwa_aktywnosci
HAVING (((Rodzaj_aktywnosci.nazwa_aktywnosci)="masaz"));
```

Następnie – osób, które korzystały z jogi lub siłowni.



```
SELECT Rejestr_aktywnosci.PESEL, Rodzaj_aktywnosci.nazwa_aktywnosci
FROM Rejestr_aktywnosci INNER JOIN Rodzaj_aktywnosci ON
Rejestr_aktywnosci.ID_aktywnosci = Rodzaj_aktywnosci.ID_aktywnosci
GROUP BY Rejestr_aktywnosci.PESEL, Rodzaj_aktywnosci.nazwa_aktywnosci
HAVING (((Rodzaj_aktywnosci.nazwa_aktywnosci)="silownia" Or
(Rodzaj_aktywnosci.nazwa_aktywnosci)="joga"));
```

Następnie trzeba wyszukać dane osób, które korzystały z masażu, ale nie korzystały z siłowni ani jogi (kwerenda wyszukująca niepasujące dane).




```
SELECT DISTINCT masaz.PESEL
FROM masaz LEFT JOIN silownia ON masaz.PESEL = silownia.PESEL
GROUP BY masaz.PESEL, silownia.PESEL
HAVING (((silownia.PESEL) Is Null));
```

Gdzie masaz jest nazwą kwerendy pierwszej a silownia – drugiej.

Innym możliwym sposobem otrzymania wyniku jest zastosowanie zapytania krzyżowego

Zadanie 14.

Systemy pozycyjne

Zadanie 14.1. (0–2)

Uzupełnij brakujące pola tabeli:

Liczba zapisana w systemie dziesiętnym	Liczba zapisana w systemie pozycyjnym o podstawie 3	Liczba zapisana w systemie pozycyjnym o podstawie 9
400		
	101201	
		2487

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Wymaganie szczegółowe

Zdający:

2. stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach: badania pierwszości liczby, zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi […].

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – za poprawnie uzupełnione przynajmniej trzy pola tabeli.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Liczba zapisana w systemie dziesiętnym	Liczba zapisana w systemie pozycyjnym o podstawie 3	Liczba zapisana w systemie pozycyjnym o podstawie 9
400	**112211**	**484**
289	101201	**351**
1861	**2112221**	2487

Zadanie 14.2. (0–1)

Zapis pewnej liczby w systemie pozycyjnym o podstawie 3 ma 20 cyfr. Ile cyfr miałby zapis tej samej liczby w systemie pozycyjnym o podstawie 9?

Odpowiedź _____

Wymaganie ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Wymaganie szczegółowe

Zdający:

2. stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy: a) na liczbach: badania pierwszości liczby, zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi […].

Zasady oceniania

3 pkt – odpowiedź poprawna w a i b,

w tym:

2 pkt – za poprawną odpowiedź a) (po 1 punkcie za każde poprawnie uzupełnione 3 pola w tabeli w a))

1 pkt – za poprawną odpowiedź b).

0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

10

Zadanie 15. (0–1)

Poniżej opisano pojęcia z zakresu technologii przesyłania danych w sieciach. Przyporządkuj właściwą nazwę, korzystając z poniższej listy, i wpisz ją w miejsce kropek.

VPN, NFC, BLUETOOTH, WiMAX, Streaming

……. połączenie tunelowe umożliwiające utworzenie poufnego połączenia klienta z serwerem prywatnej sieci swojej firmy, z wykorzystaniem infrastruktury sieci publicznej;

……. technologia bezprzewodowej komunikacji krótkiego zasięgu między różnymi urządzeniami zapewniająca zasięg do około 10 m, umożliwia przesyłanie dobrej jakości dźwięku, co zostało wykorzystują m.in. producenci bezprzewodowych słuchawek;

…….. technologia umożliwiająca bezprzewodową wymianę danych na odległość nie większą niż 20 cm.

Wymaganie ogólne
III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi.

Wymaganie szczegółowe
Zdający:
4. charakteryzuje sieć Internet, jej ogólną budowę i usługi, opisuje podstawowe topologie sieci komputerowej, przedstawia i porównuje zasady działania i funkcjonowania sieci komputerowej typu klient-serwer, peer-to-peer, opisuje sposoby identyfikowania komputerów w sieci;
1. zapoznaje się z możliwościami nowych urządzeń cyfrowych i towarzyszącego im oprogramowania.

Zasady oceniania
1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie
VPN
BLUETOOTH
NFC

Zadanie 16. (0–3)

Dane są adres IP komputera w wersji IPv4 192.168.0.7 oraz maska 255.255.255.240.

a) Podaj adres sieci, do której należy ten komputer.

_____

b) Czy komputer o adresie 192.168.0.17 należy do tej samej sieci? Uzasadnij odpowiedź.

Odpowiedź (tak/nie): _____

Uzasadnienie:

Wymaganie ogólne
III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi.

Wymaganie szczegółowe
Zdający:
4. charakteryzuje sieć Internet, jej ogólną budowę i usługi, opisuje podstawowe topologie sieci komputerowej, przedstawia i porównuje zasady działania i funkcjonowania sieci komputerowej typu klient-serwer, peer-to-peer, opisuje sposoby identyfikowania komputerów w sieci […].

Zasady oceniania
3 pkt –odpowiedź poprawna, w tym:
- 1 pkt – za odpowiedź a)
- 1 pkt – za odpowiedź b)
- 1 pkt – za uzasadnienie odpowiedzi b)

0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie
a) adres sieci 192.168.0.0
b) odpowiedź: nie
uzasadnienie:
Dla komputera: 192.168.0.17 (11000000.10101000.00000000.00010001)
oraz maski: 255.255.255.240 (11111111.11111111.11111111.11110000)
adres sieci to 11000000.10101000.00000000.00010000 czyli 192.168.0.16 – a więc inny niż dla komputera o adresie 192.168.0.7

Zadanie 17. (0–2)

Dane są adres IP komputera w wersji IPv4 192.168.0.7 oraz maska /28

a) Podaj maskę w zapisie dziesiętnym (tak jak adres IP komputera)

_____

b) Podaj adres rozgłoszeniowy sieci, do której należy podany komputer

_____

Miejsce na obliczenia

Wymaganie ogólne
III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi.

Wymaganie szczegółowe
Zdający:
4. charakteryzuje sieć Internet, jej ogólną budowę i usługi, opisuje podstawowe topologie sieci komputerowej, przedstawia i porównuje zasady działania i funkcjonowania sieci komputerowej typu klient-serwer, peer-to-peer, opisuje sposoby identyfikowania komputerów w sieci; konfiguruje przykładową lokalną sieć komputerową oraz bezprzewodowy dostęp do sieci Internet.

Zasady oceniania
1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie
a) 255.255.255.240 (11111111.11111111.11111111.1111 0000)
b) 192.168.0.15 (11000000.10101000.00000000.0000 1111)

Zadanie 18. (0–1)

Wskaż **nieprawidłowy** adres IPv6. Wybierz właściwą odpowiedź.

A. 2023:0db8:0000:0000:0000:0000:1428:57ab
B. 2023:0db8:0:0:0:0:11428:57ab
C. 2023:0db8:0:0::128:57ab
D. 2023:0db8::428:57ab
E. 2023:db8::28:57ab

Wymaganie ogólne
III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi.

Wymaganie szczegółowe
4. charakteryzuje sieć Internet, jej ogólną budowę i usługi, opisuje podstawowe topologie sieci komputerowej, przedstawia i porównuje zasady działania i funkcjonowania sieci komputerowej typu klient-serwer, peer-to-peer, opisuje sposoby identyfikowania komputerów w sieci; konfiguruje przykładową lokalną sieć komputerową oraz bezprzewodowy dostęp do sieci Internet.

Zasady oceniania
1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie
B

Zadanie 19. (0–2)

Wskaż, gdzie stosowany jest model barw CMYK oraz zapisz, jaki kolor oznacza każda z liter w tym modelu.

……..……..

Kolory:
C ……..
M ……..
Y ……..
K ……..

Wymaganie ogólne

III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych

Wymaganie szczegółowe

Zdający:
II. 3) przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami:
a) projektuje modele dwuwymiarowe i trójwymiarowe, tworzy i edytuje projekty w grafice rastrowej i wektorowej, wykorzystuje różne formaty obrazów, przekształca pliki graficzne, uwzględniając wielkość i jakość obrazów;
III.1. zapoznaje się z możliwościami nowych urządzeń cyfrowych i towarzyszącego im oprogramowania;
2. objaśnia funkcje innych niż komputer urządzeń cyfrowych i korzysta z ich możliwości.

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź niepełna: brak pierwszej części odpowiedzi albo brak najwyżej dwóch opisów kolorów.
0 pkt – odpowiedź niepoprawna albo brak odpowiedzi.

Rozwiązanie

Np. w poligrafii, drukarstwie wielobarwnym
C- cyan (cyjan), M- magenta (magenta), Y-yellow (żółty), K-black (czarny)

3. Informacja o egzaminie maturalnym z informatyki dla absolwentów niesłyszących

Informacje o egzaminie maturalnym z informatyki przedstawione w rozdziale 1. *Opis egzaminu maturalnego z informatyki* dotyczą również egzaminu dla absolwentów niesłyszących. Ponadto zdający niesłyszący przystępują do egzaminu maturalnego w warunkach i formie dostosowanych do potrzeb wynikających z ich niepełnosprawności.

Dostosowania obejmują:

- w odniesieniu do formy egzaminu maturalnego m.in.
  - zmianę sposobu sformułowania niektórych zadań (zamiana słów, zwrotów lub całych zdań), jeżeli mogłyby one być niezrozumiałe lub błędnie zrozumiane przez osoby niesłyszące (nie dotyczy to terminów typowych dla danej dziedziny wiedzy),
  - zmianę schematu punktowania niektórych zadań,
- w odniesieniu do warunków przeprowadzania egzaminu maturalnego m.in.
  - przedłużenie czasu przewidzianego na przeprowadzenie egzaminu,
  - możliwość korzystania ze słowników językowych.

Poniżej przedstawione zostały przykładowe zadania ilustrujące dostosowania dla absolwentów niesłyszących. Numeracja zadań odpowiada numeracji zadań w części 2.

Szczegółowe informacje dotyczące egzaminu dla zdających ze specjalnymi potrzebami edukacyjnymi, w tym niesłyszących, określone są w *Komunikacie dyrektora Centralnej Komisji Egzaminacyjnej w sprawie szczegółowych sposobów dostosowania warunków i form przeprowadzania egzaminu maturalnego* w danym roku szkolnym.

Zadanie 1.

Segment to spójny ciąg elementów tablicy składający się z **co najmniej 1 elementu.**

Przykład: dla tablicy A=[1, 8, 4, 2, 7, 9] segmentem jest ciąg 1,8,4 oraz ciąg 8,4,2,7 natomiast nie jest segmentem ciąg 8,2,7,9 (bo w tablicy A pomiędzy liczbami 8 i 2 jest 4)**.**

Zadanie 1.1. (0–1)

Dana jest tablica A liczb całkowitych o zawartości:

A=[2, -3, 1, -7, 4, -2, -1, 5, -3, 2, -1].

Napisz wartość pierwszego oraz wartość ostatniego elementu segmentu o maksymalnej sumie (w tej tablicy jest tylko jeden taki segment).

Odpowiedź _____

Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

4, 5

Komentarz

Segment o maksymalnej sumie (suma = 6) to: 4,-2,-1, 5.

Zadanie 1.2. (0–1)

Dana jest n-elementowa tablica A o zawartości [1, 2, 3, ... , n – 1, n]. Napisz liczbę segmentów w tej tablicy.

Odpowiedź _____

Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna lub brak odpowiedzi.

Rozwiązanie

n(n+1)/2

Zadanie 1.3. (0–3)

Elementy pewnej 1000-elementowej tablicy A zapisano kolejno w pliku `dane1_3.txt`. Każda z liczb w pliku `dane1_3.txt` należy do przedziału od [-100, 100] i zapisana jest w oddzielnym wierszu.
Napisz program wyznaczający największą sumę segmentu tablicy A.

Do oceny oddajesz:

- plik `zadanie1_3.txt` zawierający jedną liczbę; ta liczba to odpowiedź do zadania (największa suma)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach):

Zasady oceniania

3 pkt – odpowiedź poprawna – program dający w wyniku poprawną sumę.
2 pkt – program poprawny składniowo, ale znajdujący największą sumę elementów podciągu zaczynającego się zawsze od pierwszego elementu tablicy.
1 pkt – program poprawny składniowo wyliczający sumy elementów podciągów, ale nieznajdujący największej sumy.
0 pkt – odpowiedź niepoprawna albo brak odpowiedzi.

Rozwiązanie

2265

Fragment przykładowego programu:

```
ifstream plik("dane1_3.txt");
ofstream wynik("zadanie1_3.txt");
const int n = 1000;
int t[n];
for (int i = 0; i < n; i++)
   plik >> t[i];
int max = -100;
for (int dl = n; dl >= 1; dl--) {
   for (int pocz = 0; pocz <= n - dl; pocz++) {
      int suma = 0;
      for (int k = 0; k < dl; k++)
         suma += t[k + pocz];
      if (suma > max)
         max = suma;
       }
}
wynik << max << endl;
```

Komentarz:

Powyższy program przykładowy ma złożoność sześcienną. Można napisać programy o lepszej (kwadratowej lub liniowej) złożoności, ale ten program jest bardzo prosty i wynika wprost z definicji problemu. Dla danych z pliku `dane1_3.txt` program jest wystarczająco szybki.

Zadanie 1.4. (0–4)

Elementy pewnej tablicy A o 100 000 elementów zapisano kolejno w pliku `dane1_4.txt`. Każda z liczb w pliku `dane1_4.txt` należy do przedziału od <-100, 100> i zapisana jest w oddzielnym wierszu.

Pierwszy element tablicy ma indeks równy 1. Napisz program wypisujący indeks pierwszego i indeks ostatniego elementu segmentu o największej sumie. W tablicy A jest tylko jeden taki segment. **Suma elementów tego segmentu jest dodatnia.**

Do oceny oddajesz:

- plik `zadanie1_4.txt` zawierający odpowiedź do zadania zapisaną w jednym wierszu (dwie liczby oddzielone spacją będące odpowiednio numerem pierwszego i ostatniego segmentu o największej sumie)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach) _____

Zasady oceniania

4 pkt – odpowiedź poprawna oraz program dający w wyniku początek i koniec segmentu o największej sumie.
3 pkt – wyniki częściowo poprawne (np. przesunięte o jeden).
2 pkt – podano tylko jeden kraniec segmentu.
1 pkt – podano maksymalną sumę zamiast końców segmentu.
0 pkt – odpowiedź niepoprawna albo brak odpowiedzi.

Rozwiązanie

63669 70769

Fragment przykładowego programu:

```
ifstream plik ("dane1_4.txt");
ofstream wynik ("zadanie1_4.txt");
const long n=100000;
int t[n];
for (long i=0; i<n; i++)
    plik>>t[i];
long maks;  // największa suma niepustego segmentu
long maks_pocz;  // początek najlepszego niepustego segmentu
long maks_kon;  // koniec najlepszego niepustego segmentu
long ost_suma; // najlepsza niepusta suma segmentu na końcu przejrzanej części tablicy
long ost_pocz; // początek najlepszego niepustego segmentu na końcu przejrzanej części tablicy

// pierwszy rozważany niepusty segment to t[0]:
 maks = ost_suma = t[0];
maks_pocz = maks_kon = ost_pocz = 0;

for (long i = 1; i < n; i++) {
```

```
    if (ost_suma >= 0)  // Czy ost_suma + t[i] >= t[i]?
        ost_suma += t[i];
    else {
        ost_suma = t[i];
        ost_pocz = i;
    }
    if (maks < ost_suma) {
        maks = ost_suma;
        maks_pocz = ost_pocz;
        maks_kon = i;
    }
}

wynik << maks_pocz + 1 << " " << maks_kon + 1 << endl;

// Tablica, której wartości były w pliku wejściowym miała indeksy przesunięte o 1
// musimy zatem uwzględnić poprawkę na przesuniecie indeksów względem tablicy t,
// do której wczytano dane z pliku
```

Zadanie 6.

W pliku `dane6.txt` zapisano 2023 napisy. W każdym wierszu jest jeden napis. Każdy napis ma długość 100 i składa się tylko z cyfr dziesiętnych 0,1, …, 9. Nie ma napisu składającego się tylko z zer.

Dla liczby całkowitej p spełniającej warunek $2 \le p \le 10$, powiemy, że złożony z samych cyfr napis jest **p-minimalny**, jeśli zawiera cyfrę p - 1 i nie zawiera cyfr większych od p 1. Czyli będzie tak, gdy p jest najmniejszą podstawą systemu pozycyjnego, w którym taki napis da się zinterpretować jako pewną liczbę całkowitą.

Przykład:

Oto plik złożony z 5 napisów, każdy o długości 10 i zbudowany z cyfr 0 .. 9:

2001030035
0010100001
7111190009
5550001110
0000000005

Zadanie 6.1. (0–3)

Dla każdego p = 2, 3, …, 10, napisz, ile liczb z pliku `dane6.txt` jest p-minimalnych.

Przykład:

Dla przykładowego pliku mamy:

Podstawa p	Liczba liczb *p*-minimalnych
2	1
3	0
4	0
5	0

6	3
7	0
8	0
9	0
10	1

Odpowiedź:

Podstawa p	**Liczba liczb** *p*-minimalnych
2	
3	
4	
5	
6	
7	
8	
9	
10	

Do oceny oddajesz:

- odpowiedź zapisaną w tabeli powyżej
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Zasady oceniania

3 pkt – odpowiedź poprawna.
2 pkt – odpowiedź poprawna dla 8 systemów.
1 pkt – odpowiedź z wartościami mniejszymi o 1 lub przypisanie wyników do największych cyfr, a nie podstaw.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

Podstawa p	**Liczba liczb *p*-minimalnych**
2	225
3	239
4	229
5	232
6	238
7	198
8	220
9	210
10	232

Fragment przykładowego programu:

```
ifstream plik("dane6.txt");
const int n = 2023;
const int dl = 100;

int c[10]= {0,0,0,0,0,0,0,0,0,0};
string s;
int max;
for (int i=0; i < n; i++){
   plik >> s;
   max = 0;
   for (int j=0; j<dl; j++)
   {
      int t = s[j]-'0';
      if (t > max)
           max=t;
   }
   c[max]++;
}
for (int i=1; i < 10; i++)
   cout << i+1 << " " << c[i] << endl;
```

Zadanie 6.2. (0–3)

Dla każdego p = 2, 3, …, 10 wskaż wśród liczb *p*-minimalnych zapisanych w pliku `dane6.txt` przy podstawie *p* liczbę o największej sumie cyfr. Weź pod uwagę informacje z zadania 6.
Uwaga: Jeśli w pliku nie ma liczby *p*-minimalnej dla danego *p*, to nie podajemy dla tego *p* wyniku.

Przykład:
Dla przykładowego pliku mamy

podstawa	liczba z największą sumą cyfr
2	0010100001
6	5550001110
10	7111190009

Do oceny oddajesz:

- plik `zadanie6_2.txt` zawierający odpowiedź do zadania (w każdym wierszu pliku dwie liczby oddzielone spacją oznaczające podstawę *p* oraz liczbę *p*-minimalną z największą sumą cyfr)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Zasady oceniania

3 pkt – odpowiedź poprawna.
2 pkt – odpowiedź z maksymalnymi sumami cyfr zamiast wypisanych liczb z pliku.
1 pkt – odpowiedź poprawna dla przynajmniej jednej podstawy systemu pozycyjnego.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

2
1111110111010000111110110111110101100111101100001101011101111011111100010
1111101010010101001101110111
3
2202112202110120011021020222221012021121122222112110220221220101112222210202110011102022211201222212
4
3102303023333323222000310213031033311223213123300201022333212332233122002233220223030330222123110303
5
2033243301020044442112024303203344444412431034114144431110142044444333302330120243324132423443224230
6
0424454345152434504255315232105241313522450144421455415513500455212455153350255121412341354435230533
7
4440555362420610235536246551266435203464306136355645466622646061434113154656346003302120251560312564
8
7511036161247067514663274560560463624136341437237516076127224475674565004642766472364553635576317744
9
7307466887177556661727530831473748586572727146667158378441665800628386273587852686608173458128256575
10
4363784227089766038515248967862926252856617686888878836774448969907143664099174980839005479979326163

Fragment przykładowego programu:

```
ifstream plik("dane6.txt");
ofstream wynik("zadanie6_2.txt");
const int n = 2023;
const int dl = 100;
int maxs[10]={0,0,0,0,0,0,0,0,0,0};
string liczby[10];
string s;

for(int i=0; I < n; i++){
```

```
    plik >> s;
    int max=0;
    int suma=0;
    for(int j=0; j<dl; j++){
     int t = s[j]-'0';
     suma += t;
     if(t>max)
         max=t;
    }
    if (suma > maxs[max]){

        maxs[max]=suma;
        liczby[max]=s;
     }

}

for (int i=1; i < 10; i++)
    if (maxs[i]>0)
          wynik << i+1 << " " << liczby[i] << endl;
```

Zadanie 6.3. (0–2)

Dla dodatniej, parzystej liczby całkowitej n powiemy, że napis s[1..*n*] jest **antypalindromem**, jeżeli s[i] ≠ s[*n*-i+1], dla każdego i = 1, 2, …, *n*/2.
Antypalindrom ≠ palindrom

Zapisz wszystkie antypalindromy z pliku `dane6.txt` (każdy antypalindrom zapisz w oddzielnym wierszu; pamiętaj, aby kolejność była taka sama jak w pliku z danymi) w pliku zadanie6_3.txt. W oddzielnym, ostatnim wierszu pliku zapisz liczbę antypalindromów.

Przykład:
Dla przykładowego pliku antypalindromem jest tylko jeden napis: 7111190009.

Do oceny oddajesz:

- plik `zadanie6_3.txt` zawierający odpowiedź do zadania (w każdym wierszu pliku jeden antypalindrom i liczba antypalindromów w ostatnim wierszu pliku)
- plik(i) z komputerową realizacją zadania (kodem programu) o nazwie (nazwach)

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź poprawna bez liczby antypalindromów
ALBO
podana tylko poprawna liczba antypalindromów.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

14863571230330690483692339776957319481338573283002786460810007311554983485279735295202746116730687 72

88161000430076445258374062631112335058830726585705022618365814508170357802437654523472161202176703 21

53386843320088227451102048899149759918270810392395300858949919123721068843789650766513129969281330 04

72345115624756735503557135172931285542327323024478487179948877173011747409807198918068375855395457 31

4

Fragment przykładowego programu:

```
bool anty(string s){
    for (int i=0; i<s.length() / 2; i++)
       if (s[i] == s[s.length() – 1 - i])
           return false;
   return true;
}

int main(){
  const int n = 2023;
  ifstream plik("dane6.txt");
  ofstream wynik("zadanie6_3.txt");
  string s;

   int ile=0;
   for (int i=0; i<n; i++) {
      plik>>s;
      if (anty(s)) {
         ile++;
         wynik << s << endl;
       }
   }
   wynik << ile;
}
```

Zadanie 10.

Zakład mleczarski produkuje ekologiczne, naturalne masło. Zakład mleczarski sprzedaje masło do kilkunastu sklepów ze zdrową żywnością. Zakład codziennie dostaje zamówienia. Zamówienia przesyłane są rano na linię produkcyjną oraz do działu transportu.
W pliku `zamowienia.txt` zapisano datę i wielkość zamówienia (w kilogramach). Dane w wierszach pliku oddzielone są znakiem tabulacji. Zakład otrzymuje zamówienie rano przed rozpoczęciem produkcji. Zamówienie ma wpływ na produkcję oraz na transport w tym dniu.

Przykładowy fragment pliku:

data	zamówienie
02/01/2018	299
03/01/2018	43
04/01/2018	296
05/01/2018	287

Wykorzystaj dostępne narzędzia informatyczne oraz dane z pliku zamówienia.txt. Rozwiąż poniższe zadania. Odpowiedzi zapisz do pliku `wyniki10.txt`. Przed każdą odpowiedzią napisz numer zadania.

Zadanie 10.1. (0–3)

Dla każdego miesiąca od stycznia 2018 do grudnia 2019 podaj sumę kilogramów zamówionego masła. Dla wykonanego zestawienia utwórz wykres kolumnowy. Prawidłowo podpisz osie wykresu. Napisz tytuł wykresu.

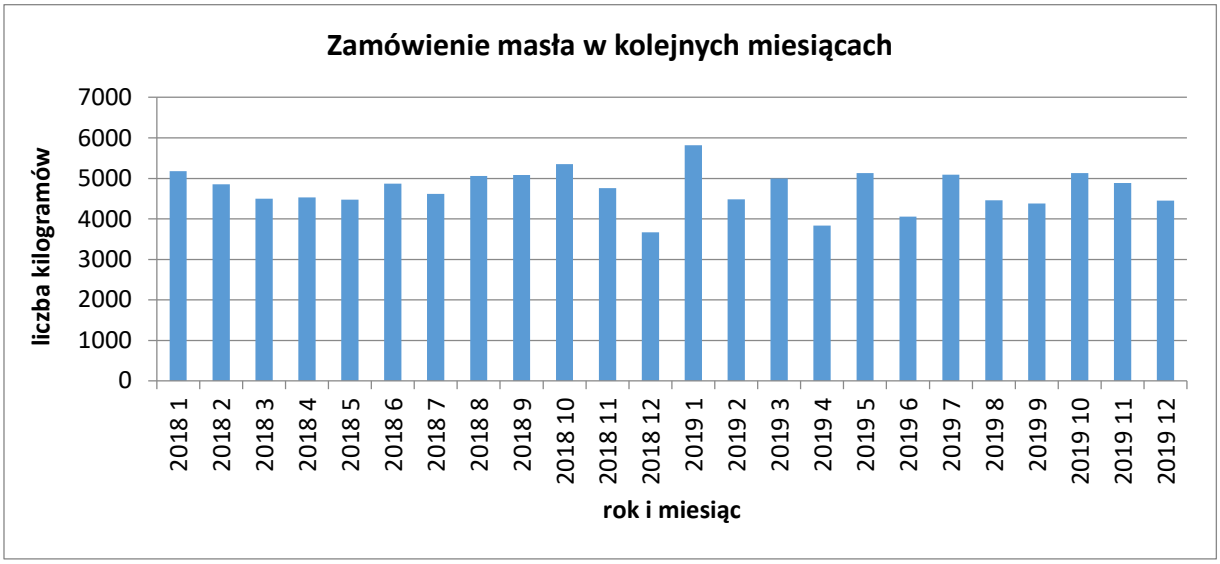
Do oceny oddajesz plik **z realizacją zadania** o nazwie _____

Zasady oceniania

3 pkt – odpowiedź poprawna w tym:
- 2 pkt – wykres, w tym:
  - 1 pkt – odpowiedni typ wykresu i dobór danych,
  - 1 pkt – za prawidłowy opis;
- 1 pkt – zestawienie danych do wykresu.

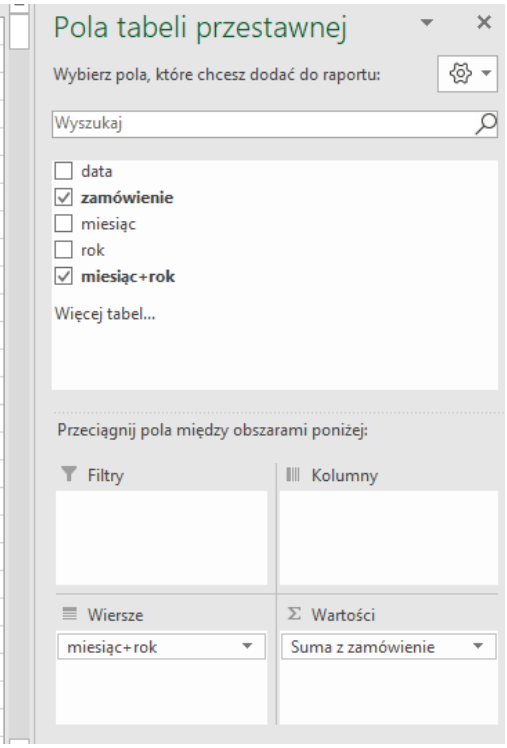
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie



Przykładowe wyliczenia w arkuszu kalkulacyjnym:
Aby wykonać zestawienie należy wybrać z daty rok i miesiąc oraz wykonać obliczenia np. z pomocą tabeli przestawnej albo całość obliczeń wykonać w tabeli przestawnej.

Etykiety wierszy	Suma z zamówienie
2018 1	5178
2018 10	5351
2018 11	4761
2018 12	3671
2018 2	4852
2018 3	4497
2018 4	4528
2018 5	4474
2018 6	4873
2018 7	4616
2018 8	5057
2018 9	5085
2019 1	5822
2019 10	5132
2019 11	4882
2019 12	4448
2019 2	4480
2019 3	4995
2019 4	3831
2019 5	5128
2019 6	4054
2019 7	5092
2019 8	4456
2019 9	4382
Suma końcowa	113645



Następnie wystarczy wstawić wykres i prawidłowo go opisać.

Do oceny oddajemy plik arkusza kalkulacyjnego zawierający obliczenia oraz wykres.

Informacja do zadań 10.2. i 10.3.

Dział transportu wozi masło samochodem o ładowności 400 kg. Ładowność to maksymalny ciężar masła, jaki można włożyć na samochód. Samochód wyjeżdża z zakładu, gdy będzie wypełniony w 100%, to znaczy: łączne zamówienie od ostatniego transportu wynosi co najmniej 400 kg. Jeśli łączne zamówienie jest niższe, to nie będzie w tym dniu transportu. Jeśli łączne zamówienie jest wyższe niż 400 kg, to pozostała część zamówienia pojedzie następnym transportem (to znaczy, że każde zamówienie można dzielić – jeśli nie ma miejsca na całość, to wysyła się tę część, która się zmieści a reszta zamówienia jest obsługiwana w kolejnym transporcie). W sytuacji, gdy łączne zamówienie jest większe lub równe wielokrotności 400 kg to w jednym dniu może odbyć się kilka transportów po 400 kg każdy.
Zakładamy, że każdego dnia samochód jest w stanie wykonać dowolną liczbę pełnych kursów.

Zadanie 10.2. (0–1)

Podaj liczbę dni, w których odbył się transport masła do sklepów.

Odpowiedź _____
Do oceny oddajesz plik z realizacją zadania o nazwie _____

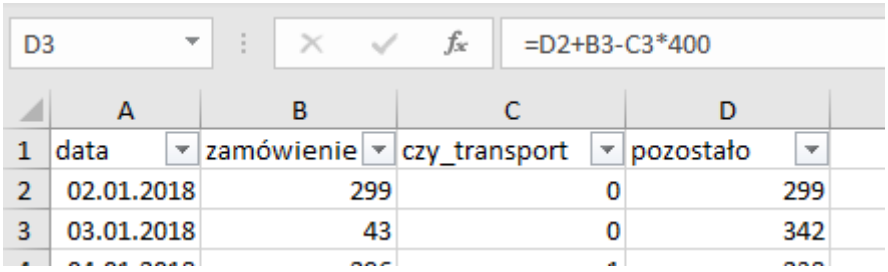
Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

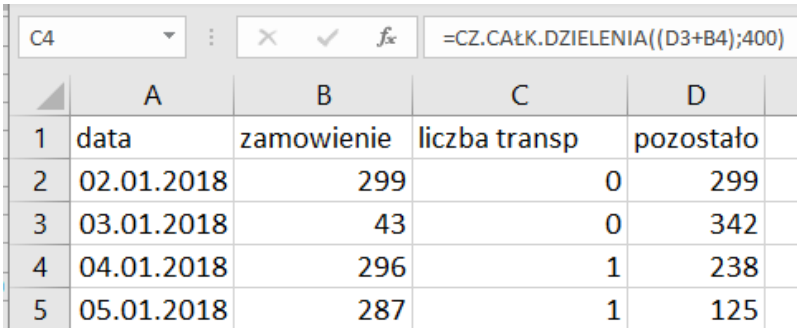
280 dni

Obliczenia:
Towar pozostały w magazynie



	A	B	C	D
1	data	zamówienie	czy_transport	pozostało
2	02.01.2018	299	0	299
3	03.01.2018	43	0	342

Czy będzie transport:



	A	B	C	D
1	data	zamowienie	liczba transp	pozostało
2	02.01.2018	299	0	299
3	03.01.2018	43	0	342
4	04.01.2018	296	1	238
5	05.01.2018	287	1	125

Wynik otrzymujemy zliczając funkcją *Licz.Jeżeli* liczbę dni, w których liczba transportów jest większa od 0.

Zadanie 10.3. (0–2)

Podaj daty, kiedy samochód wykonał co najmniej dwie dostawy w tym samym dniu.

Odpowiedź _____

Do oceny oddajesz plik z realizacją zadania o nazwie _____

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – brak jednej daty.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

05.07.2018
16.07.2018
26.03.2019
27.11.2019

Przy wcześniejszych obliczeniach wystarczy ukryć (schować) za pomocą filtru dni, kiedy nie było transportu, lub był tylko jeden transport.

Informacja do zadań 10.4 i 10.5.

Najczęściej zakład mleczarski produkuje 200 kg masła dziennie. Czasami, gdy zamówienie jest większe niż połowa porannej zawartości magazynu produkcja w tym dniu wzrasta o 30%. Produkcja jest zmniejszana o 20% zawsze, gdy rano w magazynie jest więcej niż 1500 kg masła (nieważne, jak duże jest wtedy zamówienia).

W dniu 2.01.2018 roku rano w magazynie znajdowało się 1000 kg masła. Weź pod uwagę zamówienia, opisany cykl produkcyjny oraz cykl transportowy. Wykonaj symulację porannej zawartości magazynu w okresie od 2 stycznia 2018 do 31 grudnia 2019 r.

Zadanie 10.4. (0–2)

Znajdź najdłuższy okres stabilizacji – kolejne dni, kiedy wielkość produkcji nie zmieniała się (znajdź kolejne dni, w których produkcja masła była taka sama). Podaj początek i długość tego okresu.

Odpowiedź _____
Do oceny oddajesz plik z realizacją zadania o nazwie _____

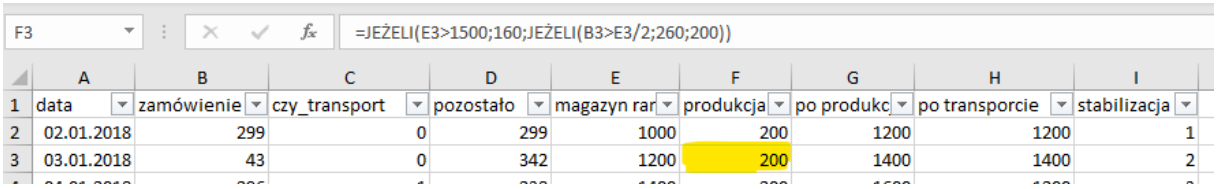
Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

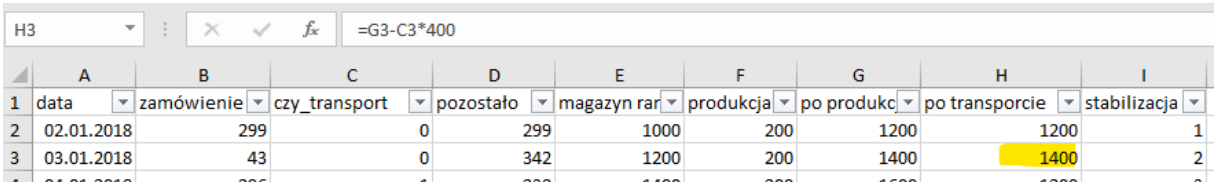
38 dni, początek 26.04.2018

Obliczenie wartości produkcji



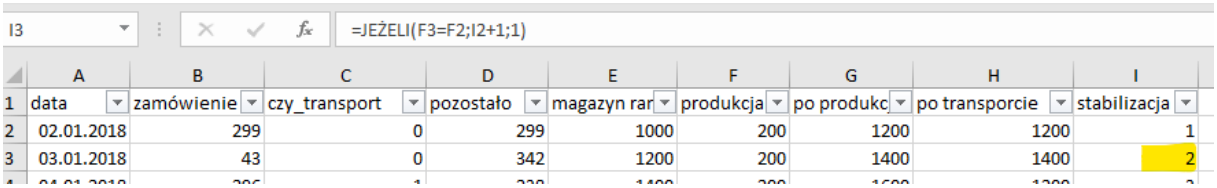
	A	B	C	D	E	F	G	H	I
1	data	zamówienie	czy_transport	pozostało	magazyn rar	produkcja	po produkc	po transporcie	stabilizacja
2	02.01.2018	299	0	299	1000	200	1200	1200	1
3	03.01.2018	43	0	342	1200	200	1400	1400	2

Stan magazynu po transporcie (jeśli był tego dnia)



	A	B	C	D	E	F	G	H	I
1	data	zamówienie	czy_transport	pozostało	magazyn rar	produkcja	po produkc	po transporcie	stabilizacja
2	02.01.2018	299	0	299	1000	200	1200	1200	1
3	03.01.2018	43	0	342	1200	200	1400	1400	2

Informacja, czy nastąpiła stabilizacja – czy nie zmieniała się wielkość produkcji (każdy następny dzień, kiedy produkcja masła jest taka sama jak poprzedniego dnia jest zliczany).



	A	B	C	D	E	F	G	H	I
1	data	zamówienie	czy_transport	pozostało	magazyn rar	produkcja	po produkc	po transporcie	stabilizacja
2	02.01.2018	299	0	299	1000	200	1200	1200	1
3	03.01.2018	43	0	342	1200	200	1400	1400	2

Następnie wystarczy w kolumnie I znaleźć maksymalną wartość i pobrać odpowiadającą jej datę początku szukanego okresu.

Zadanie 10.5. (0–2)

Jaki był najniższy i jaki najwyższy stan magazynu w opisanym w zadaniu przedziale czasowym?

Odpowiedź _____

Do oceny oddajesz plik z realizacją zadania o nazwie _____

Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

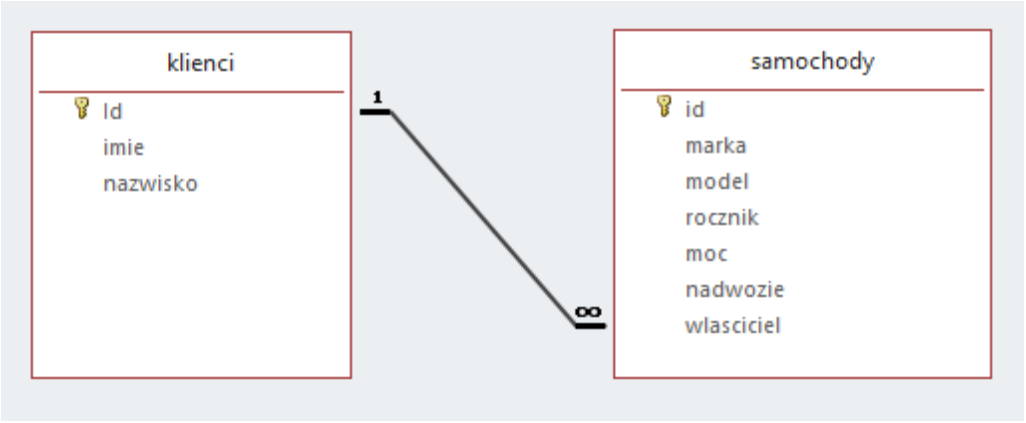
Rozwiązanie

Odpowiedź
min 20 max 1700

Wystarczy policzyć minimum i maksimum z odpowiedniej kolumny (w naszym zadaniu H), wszystkie pozostałe obliczenia zostały wykonane w czasie rozwiązywania zadania 10.4.

Zadanie 12.

Dane są dwie tabele bazy danych: *klienci* i *samochody* połączone relacją jeden do wielu (patrz rysunek):



W tabelach zapisane są informacje zgodne z nazwami pól (np. „marka” – marka samochodu). Pole *id* w tabeli *klienci* oznacza identyfikator klienta i jest kluczem głównym w tej tabeli. Pole *id* w tabeli *samochody* oznacza identyfikator samochodu. Pole o nazwie *wlasciciel* w tabeli samochody jest kluczem obcym w tej tabeli i oznacza identyfikator klienta, który jest właścicielem samochodu. Na potrzeby zadania zakładamy, że każdy samochód ma tylko jednego właściciela, natomiast każda osoba może być właścicielem jednego lub więcej niż jednego samochodu.

Zadanie 12.1. (0–1)

Zapisz **w języku SQL** zapytanie, które da w wyniku listę klientów (z id, imionami i nazwiskami) uporządkowaną alfabetycznie według nazwiska

Zasady oceniania

1 pkt – odpowiedź poprawna.
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Rozwiązanie

```
SELECT *
FROM klienci
ORDER BY klienci.nazwisko;
```

Zadanie 12.2. (0–2)

Zapisz **w języku SQL** zapytanie, które da w wyniku listę marek i modeli samochodów, dla których *rocznik* zawiera się w przedziale [2010,2020] oraz *moc* jest większa niż 100 KM. Listę posortuj niemalejąco wg nazwy marki i modelu.

Uwaga: Każda para *model* i *marka* w zestawieniu może pojawić się tylko raz (nawet jeśli jest kilka samochodów tej samej marki i modelu spełniających warunki zadania).

Zasady oceniania

2 pkt – odpowiedź poprawna.
1 pkt – odpowiedź, w której zdający nie uwzględnie, że każda para *marka* i *model* może pojawić się w zestawieniu tylko raz (brak DISTINCT)
0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

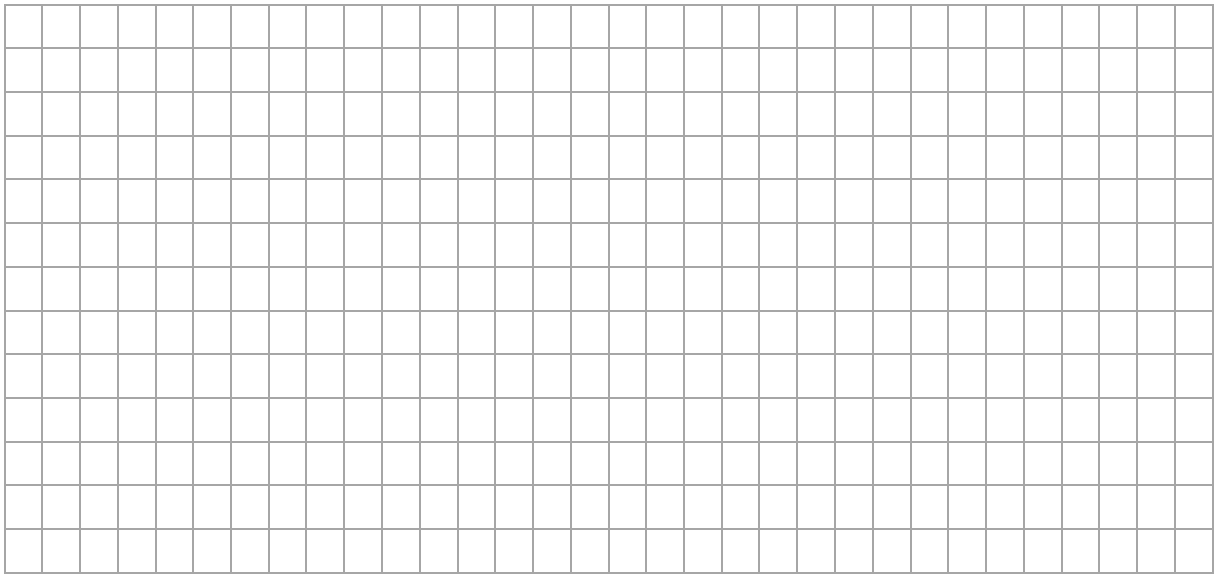
Przykładowe rozwiązanie

```
SELECT DISTINCT samochody.marka, samochody.model
FROM samochody
WHERE (samochody.moc>100 AND (samochody.rocznik Between 2010 And 2020)
ORDER BY samochody.marka, samochody.model;
```

Zadanie 12.3. (0–2)

Zapisz **w języku SQL** zapytanie, które da w wyniku listę klientów z ich id, imionami, nazwiskami oraz liczbą samochodów, które mają. Lista musi być posortowana malejąco według liczby samochodów.

Uwaga: pamiętaj, że część klientów może nie mieć żadnego samochodu (wtedy trzeba wypisać ich z liczbą samochodów zero)



Zasady oceniania

2 pkt – odpowiedź poprawna.

1 pkt – odpowiedź, w której zdający nie bez uwzględnienia, że klient może nie posiadać samochodu (INNER JOIN zamiast LEFT JOIN).

0 pkt – odpowiedź niepełna lub niepoprawna albo brak odpowiedzi.

Przykładowe rozwiązanie

```
SELECT klienci.imie, klienci.nazwisko, Count(samochody.id) AS ile
FROM klienci LEFT JOIN samochody ON klienci.Id = samochody.wlasciciel
GROUP BY klienci.imie, klienci.nazwisko
ORDER BY Count(samochody.id) DESC;
```

Uchwała Rady Głównej Nauki i Szkolnictwa Wyższego oraz Konferencji Rektorów Akademickich Szkół Polskich o informatorach maturalnych od 2023 roku

Rada Główna Nauki i Szkolnictwa Wyższego

KONFERENCJA REKTORÓW AKADEMICKICH SZKÓŁ POLSKICH KRASP

Uchwała
Rady Głównej Nauki i Szkolnictwa Wyższego
oraz
Konferencji Rektorów Akademickich Szkół Polskich
z dnia 19 listopada 2020 r.
w sprawie informatorów o egzaminie maturalnym od roku 2022/2023

Rada Główna Nauki i Szkolnictwa Wyższego oraz Konferencja Rektorów Akademickich Szkół Polskich systematycznie i z uwagą obserwują egzamin maturalny, który stanowi podstawowe źródło informacji o poziomie przygotowania kandydatów na studia w polskich uczelniach. W zgodnej opinii obu instytucji przedstawicielskich środowiska akademickiego polski system egzaminacyjny dostarcza w tym zakresie w pełni wiarygodnych i opracowanych na czas danych. Zaufanie do tego systemu przekonująco potwierdziła KRASP wiosną tego roku, deklarując wolę odłożenia rekrutacji do szkół wyższych do czasu, gdy będzie możliwe bezpieczne przeprowadzenie egzaminu maturalnego w roku pandemii.

Drugim, bardzo istotnym zadaniem, które konsekwentne realizuje polski system egzaminacyjny jest wskazywanie za pomocą publikowanych materiałów kierunku rozwoju kształcenia w poszczególnych przedmiotach w stronę umiejętności złożonych, niezbędnych zarówno na studiach, jak i – w coraz większym stopniu – w życiu codziennym.

Ważną rolę w tym procesie odgrywają informatory o egzaminie maturalnym. Z jednej strony wydobywają oraz ilustrują za pomocą zadań najważniejsze wymagania podstawy programowej kształcenia ogólnego. Z drugiej strony, wiarygodnie informują kolejne pokolenia maturzystów o strukturze egzaminu maturalnego oraz o sposobie oceniania ich prac. W naszej opinii, przedłożone do zaopiniowania informatory dobrze realizują te cele.

Środowisko akademickie docenia starania systemu egzaminacyjnego o to, by systematycznie doskonalić swoją pracę i deklaruje dalsze wsparcie merytoryczne tych działań.

prof. dr hab. Zbigniew Marciniak
Przewodniczący Rady Głównej
Nauki i Szkolnictwa Wyższego

prof. dr hab. Inż. Arkadiusz Mężyk
Przewodniczący Konferencji Rektorów
Akademickich Szkół Polskich

EGZAMIN MATURALNY **Informatyka**

Z opinii Recenzentów:

W informatorze przedstawiono przykładowe zadania egzaminacyjne wraz z rozwiązaniami i odniesieniami do wymagań podstawy programowej. Zawarto w nim zadania różnego typu, sprawdzające umiejętność rozumienia i analizowania problemów oraz programowania i rozwiązywania problemów z wykorzystaniem komputera. Główny nacisk położono przy tym na układanie i programowanie algorytmów. Celem publikacji jest zapoznanie ucznia i nauczyciela z rodzajami zadań, które mogą pojawić się na egzaminie maturalnym, ze sposobem ich formułowania oraz różnymi metodami przedstawiania rozwiązań (na przykład pliki z kodem programu, pliki zawierające obliczone rozwiązania, zestawienia w arkuszu kalkulacyjnym czy też odpowiedzi wpisywane w arkusz).

dr Marcin Engel

Ogólne wrażenie jest bardzo dobre. Zadania przemyślane, o właściwym stopniowaniu trudności w kolejnych etapach rozwiązywania problemu. […]

dr Piotr Chrzastowski-Wachtel

Przedstawiony w informatorze opis egzaminu i sposobu jego punktowania jest zwięzły, precyzyjny i zrozumiały. Dla mnie jako nauczycielki, ważne jest to szczególnie, ponieważ pozwoli mi to lepiej rozpocząć przygotowania uczniów do egzaminu w zmienionej formule. […]
Mocna stroną zestawu zadań są zadania z programowania i myślenia algorytmicznego, które stanowią znaczną część informatora. Wśród nich mamy zadania z operacji na tekstach i znakach: szyfrowanie, sprawdzanie czy słowo jest palindromem, szukanie wzorca w tekście. Poruszany jest problem sortowania i wyszukiwania elementów w zbiorze. Mamy też zadania z rozumienia rekurencji. […]
W grupie zadań programistyczno-algorytmicznych są zadania i pytania łatwe, średnie i trudne. Rozwiązujący musi wymyślić rozwiązanie, dobrać odpowiednie struktury danych, a następnie poprawnie zaprogramować zadanie. Wśród zadań algorytmicznych są też przykłady zadań z analizy i rozumienia algorytmu, gdzie rozwiązanie zadania wystarczy zapisać na arkuszu maturalnym bez użycia komputera. Przykładowe rozwiązania zadań przy użyciu komputera są jasno i zrozumiale opisane. […]
Cieszy mnie umieszczenie w informatorze zamkniętych i otwartych zadań niewymagających użycia komputera. Do tej pory takie pytania pojawiały się w części A1. W egzaminie maturalnym, który opisuje Informator, części A1 i A2 zostały połączone. Podanie sporej ilości przykładowych pytań pozwoli zdającemu oswoić się z nową formą matury i lepiej do niej przygotować. […]
Podsumowując, uważam, że *Informator o egzaminie maturalnym z informatyki od roku szkolnego 2022/2023* będzie bardzo dobrym drogowskazem dla nauczycieli pokazującym w jakim kierunku mają przygotowywać uczniów do egzaminu maturalnego. […]

Joanna Śmigielska, XIV LO im. Stanisława Staszica w Warszawie

ISBN 978-83-66725-25-6