DIAMENT

Numer ID

H	5	3	8

EGZAMIN KOŃCZĄCY KURS 2024/2025

INFORMATYKA

Poziom rozszerzony

WYPEŁNIA KURSANT

WINDOWS 11 X64
(system operacyjny)

MICROSOFT OFFICE 365
(program użytkowy)

PYCHARM
(środowisko programistyczne)

Kod arkusza
2425-01

Data: **25 kwietnia 2025 r.**

Godzina rozpoczęcia: **16:30**

Czas trwania: **210 minut**

Liczba punktów do uzyskania: **50**

Przed rozpoczęciem pracy:

- Sprawdź, czy masz **właściwy** arkusz egzaminacyjny – właściwa **formuła, przedmiot** i **poziom**.
- Jeśli masz **niewłaściwy** arkusz – zgłoś się do zespołu komisyjnego, **nie rozrywaj banderol!**
- Jeśli masz **właściwy** arkusz – rozerwij banderole po otrzymaniu takiego polecenia od zespołu. Przeczytaj instrukcję znajdującą się na następnej stronie.

Scanned with CamScanner

Instrukcja dla kursanta:

- Sprawdź, czy twój arkusz posiada dokładnie 19 stron (zadania 1-7)
- Sprawdź, czy zespół komisyjny podał informację o tym, gdzie znajdują się pliki do zadań.
- Sprawdź, czy zespół komisyjny podał informację o tym, gdzie deponować rozwiązania zadań.
- Jeśli po sprawdzeniu powyższych trzech punktów czegoś brakuje, zgłoś to do zespołu komisyjnego.
- Na pierwszej stronie oraz karcie odpowiedzi wpisz swój numer identyfikacyjny
- Na pierwszej stronie wpisz zadeklarowane przez ciebie: system operacyjny, program użytkowy i środowisko programistyczne
- Komputerowe rozwiązania zadań umieść w folderze o nazwie twojego numeru identyfikacyjnego.
- W przypadku zadań programistycznych z użyciem komputera w folderze umieszczaj tylko pliki z rozszerzeniem odpowiadającym zadeklarowanemu przez ciebie oprogramowaniu (*.cpp lub *.py). **Nie umieszczaj** całych folderów z projektem!
- W przypadku zadań programistycznych bez użycia komputera (zadania oznaczone symbolem 🗎) rozwiązania zapisz bezpośrednio na arkuszu
- W przypadku zadań z arkuszem kalkulacyjnym i bazą danych w folderze umieszczaj tylko pliki z rozszerzeniem odpowiadającym zadeklarowanemu przez ciebie oprogramowaniu
- W przypadku, jeśli rozwiązaniem zadania lub jego części jest baza danych utworzona z wykorzystaniem MySQL (MariaDB), w folderze umieść treści zapytań w języku SQL (np. w pliku *.txt) oraz wyeksportowaną całą bazę w formacie *.sql
- Pliki oddawane do oceniania nazwij dokładnie tak, jak nakazuje treść zadania, lub tak, jak zadeklarowałeś w arkuszu (nazwy plików w arkuszu powinny mieć również dopisane rozszerzenie). Pliki o innych nazwach **nie będą sprawdzane!**
- **Przed zakończeniem egzaminu** zapisz ostateczne wersje plików w folderze, oraz skompresuj ten folder (*.7zip, *.zip, *.rar, *.tar.gz itp.)
- Pisz wyraźnie i używaj tylko długopisu/pióra z czarnym atramentem. Nie używaj korektora. Nie wpisuj nic w kolumnach na karcie odpowiedzi.
- Zapisy w brudnopisie nie będą sprawdzane i oceniane (chyba, że w zadaniu zabraknie miejsca i zostanie dopisana odpowiednia adnotacja)

Scanned with CamScanner

Zadanie 1. Tajemniczy kod agentów

W tajnej operacji służby specjalnej wykorzystywany jest system szyfrowania oparty na ciągu binarnym. Procedura generowania tego ciągu działa następująco:

1. Agent wybiera początkowy ciąg binarny S (zwany „prefiksem") oraz ustala liczbę kroków k.
2. W każdym kroku wykonuje się następującą operację: do aktualnego ciągu dopisuje się jego negację – ciąg, w którym każda cyfra jest zamieniana: 0→1 oraz 1→0.

Powyższą procedurę dokładniej opisuje poniższy algorytm:

```
FUNKCJA NEGACJA(CIĄG):
  wynik ← pusty ciąg
  DLA KAŻDEGO znaku z CIĄG wykonuj:
    JEŻELI znak = '0' TO
        dołącz '1' do wynik
    W PRZECIWNYM RAZIE
        dołącz '0' do wynik
  ZWRÓĆ wynik

FUNKCJA TRANSFORMACJA(S, k):
  JEŻELI k = 0 TO
    ZWRÓĆ S
  W PRZECIWNYM RAZIE
    T ← TRANSFORMACJA(S, k - 1)
    N ← NEGACJA(T)
    ZWRÓĆ T + N
```

T<- "0"
N<- "1" } "01"

T<- "01"
N<- "10" } "01 10"

T<- "0110"
N<- "1001" } "01101001"

Zadanie 1.1. (0-3)

Uzupełnij poniższą tabelę, podając wynik działania funkcji TRANSFORMACJA(S,k):

S	k	Wynik transformacji
"0"	3	"01101001"
"1"	2	"1001"
"01"	3	"0110,1001,10010110"
"101"	3	"101,010 | 010101 | 010101101010"

T<- "1" } "10"
N<- "0"

T<- "10"
N<- "01"

$3 \cdot 2^3 = 24$

Zadanie 1.2. (0-3)

Przyjmij, że wynikiem działania TRANSFORMACJA(S,k) dla S="0" jest ciąg T.

- Jaka jest **największa liczba kolejnych takich samych znaków** (czyli najdłuższy fragment typu „000..." lub „111...") w ciągu T dla **k = 7**?

  2

- Jaka jest **suma znaków** ciągu T dla **k=n** (gdzie n to dowolna liczba całkowita większa od 0)?

  [redacted] 2^{n-1}

  2^{n-1}

  k=7 -> 6
  k=1 -> 1
  k=4 ->
  2^3

- Dla jakiej **najmniejszej** liczby kroków wynik transformacji zawiera **co najmniej** 100 jedynek?

  k=8

  $2^6=64$
  $2^7=128$

Zadanie 1.3. (0-2)

Ciąg T został utworzony przez wykonanie funkcji TRANSFORMACJA("0", 4).

k=4
S="0"

Ile **unikalnych** spójnych podciągów długości 4 występuje w tym ciągu?

9

"0"
0110 1001 1001 0110

$2^4 - 16$

0110 0100
1101 1001 1100
1010 0011 0010
1011

Zadanie 2. Liczby

Liczbą dozwolonych systemów liczbowych nazywamy maksymalną liczbę systemów (do systemu szesnastkowego), w których można zapisać daną wartość.

{0 1} {0, 1 ... F}

Przykład:

Liczbę 1101 możemy zapisać w systemach od dwójkowego do szesnastkowego
Liczbę 2025 możemy zapisać w systemach od szóstkowego do szesnastkowego
Liczbę 131D49F możemy zapisać tylko w systemie szesnastkowym

Zadanie 2.1. (0-3)

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który dla danej liczby n (zapisanej w systemie od 2 do 16) obliczy jej liczbę dozwolonych systemów liczbowych, a następnie określ złożoność obliczeniową twojego algorytmu.

Przykład:

*Dla liczby 2137 wynikiem jest **9** (systemy od ósemkowego do szesnastkowego)*
*Dla liczby 1BCD9 wynikiem jest **3** (systemy od czternastkowego do szesnastkowego)*

Uwaga: W zapisie algorytmu możesz korzystać tylko z instrukcji sterujących, operatorów arytmetycznych: dodawania, odejmowania, mnożenia, dzielenia, dzielenia całkowitego i reszty z dzielenia; operatorów logicznych, porównań, instrukcji przypisania lub samodzielnie napisanych funkcji i procedur wykorzystujących powyższe operacje. **Zabronione** jest używanie funkcji wbudowanych oraz operatorów innych niż wymienione, dostępnych w językach programowania.

Specyfikacja:

Dane: L- długość napisu n

n – napis przedstawiający liczbę zapisaną w systemie od 2 do 16

Wynik:

k – liczba dozwolonych systemów liczbowych liczby *n*

Język programowania: .....Python..... (Pseudokod/C++/Python)

Algorytm:

```
maxi=0
digits= "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
for i in range(len(n)):
    znak= n[i]
    for j in range(L)
        if digits[j]==znak:
            if j > maxi:
                maxi=j
```


Oprawa: Bartosz Wój

~~k = 16 - (maxi+1) + 1~~

~~print(k)~~

```
maxi = 0
idx = 0
while idx < L:
    znak = n[idx]
    ord(znak)
    if ord("0") <= znak <= ord("9"):
        c = ord(znak) - ord("0")
    elif ord("A") <= ord(znak) <= ord("F"):
        c = ord(znak) - ord("A") + 10

    if c > maxi:
        maxi = c

    idx += 1

k = 16 - (maxi + 1) + 1
```

Złożoność obliczeniowa: $O(n)$

Informacja do zadań 2.2. – 2.4.

W pliku `liczby.txt` znajduje się 1000 wierszy. W każdym wierszu znajdują się dwie liczby oddzielone pojedynczym znakiem tabulatora. Liczby w danej parze są zapisane w takim samym systemie liczbowym (od dwójkowego do szesnastkowego), oraz druga liczba jest pierwszą liczbą zapisaną w odwrotnej kolejności w systemie dziesiętnym (czyli na przykład 135 – 531)

Napisz program(-y), który(-e) da(-dzą) odpowiedzi do poniższych zadań. Odpowiedzi zapisz w pliku wyniki2.txt, a każdą z nich poprzedź numerem odpowiedniego zadania.

Plik `liczby_przyklad.txt` zawiera 50 wierszy przykładowych danych spełniających warunki zadania. Odpowiedzi dla danych z pliku `liczby_przyklad.txt` są podane pod treściami zadań.

Zadanie 2.2. (0-2)

Na podstawie powyższej zależności określ, w jakich systemach liczbowych są zapisane określone pary. Jako rozwiązanie podaj liczbę par zapisanych w każdym z systemów od dwójkowego do szesnastkowego.

***Uwaga**: Dla każdej pary liczb istnieje tylko jeden system liczbowy w którym te liczby spełniają podaną zależność*

Dla danych z pliku `liczby_przyklad.txt` poprawna odpowiedź to:

2: 1	10: 5
3: 3	11: 5
4: 3	12: 2
5: 2	13: 2
6: 5	14: 5
7: 2	15: 5
8: 3	16: 3
9: 4	

Zadanie 2.3. (0-2)

Znając systemy w których zostały zapisane pary liczb znajdź w gronie wszystkich liczb liczbę największą i najmniejszą. Wynik podaj w jednej linii, najpierw liczba najmniejsza a potem największa w systemach w jakich zostały pierwotnie zapisane.

Dla danych z pliku `liczby_przyklad.txt` poprawna odpowiedź to:

42 36470

Scanned with CamScanner

Zadanie 2.4. (0-3)

Przeprowadź analizę częstości wystąpień znaków (od 0 do F) we wszystkich
Wynik podaj w procentach z dokładnością do 2 miejsc po przecinku.

Dla danych z pliku `liczby_przyklad.txt` poprawna odpowiedź to:

0: 14.77%	8: 4.14%
1: 19.46%	9: 2.52%
2: 13.51%	A: 3.6%
3: 12.79%	B: 1.44%
4: 7.21%	C: 1.08%
5: 8.29%	D: 0.54%
6: 5.05%	E: 0.72%
7: 4.32%	F: 0.54%

Do oceny oddajesz:

- Plik tekstowy `wyniki2.txt` zawierający odpowiedzi do zadań 2.2.-2.4.
  (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- Plik(i) zawierający(e) kody źródłowe o nazwie(-ach)
  (Uwaga – jeśli nie oddasz plików, zadanie zostanie ocenione na 0 punktów)

Zadanie 2.2. ...liczby_solve.py...

Zadanie 2.3. ...liczby_solve.py...

Zadanie 2.4. ...liczby_solve.py

Zadanie 3. Kebab

Procesem zwijania danej liczby nazywamy zastępowanie tej liczby poprzez sumę liczb uzyskanych z rozkładu na czynniki pierwsze. Proces powtarzamy do momentu, gdy uzyskamy dwa razy taką samą liczbę.

Przykład:

*Proces zwijania dla liczby **60** to: 60 -> 12 -> 7 -> 7, bo:*

*60 = 2 * 2 * 3 * 5 -> 2 + 2 + 3 + 5 = 12*
*12 = 2 * 2 * 3 -> 2 + 2 + 3 = 7*
7 = 7 (7 w rozkładzie na czynniki pierwsze to po prostu 7) -> 7 = 7
Uzyskaliśmy dwa razy liczbę 7 więc proces zwijania się kończy

*Proces zwijania liczby **60** ma **długość 4***

Liczbą kebabową nazywamy liczbę uzyskaną poprzez zsumowanie wszystkich liczb uzyskanych w procesie zwijania.

Przykład:

*Liczbą kebabową liczby **60** jest liczba **86** (60 + 12 + 7 + 7)*

W pliku `kebab.txt` znajduje się 750 wierszy. W każdym wierszu znajduje się tylko jedna liczba. **Napisz program**(-y), który(-e) da(-dzą) odpowiedzi do poniższych zadań. Odpowiedzi zapisz w pliku wyniki3.txt, a każdą z nich poprzedź numerem odpowiedniego zadania.

Plik `kebab_przyklad.txt` zawiera 50 wierszy przykładowych danych spełniających warunki zadania. Odpowiedzi dla danych z pliku `kebab_przyklad.txt` są podane pod treściami zadań.

Zadanie 3.1. (0-3)

Dla każdej liczby z pliku przeprowadź proces zwijania. Podaj liczbę o najdłuższym procesie zwijania oraz długość tego procesu. Najpierw wypisz długość procesu, a w nowej linii liczbę. Jeśli jest kilka takich liczb, wypisz je wszystkie.

Dla danych z pliku `kebab_przyklad.txt` poprawna odpowiedź to:

10
4405
2566

Zadanie 3.2. (0-2)

Ile jest liczb kebabowych które są palindromami, a ile liczb kebabowych które są liczbami pierwszymi? Jako wynik podaj najpierw ilość kebabowych palindromów, a następnie ilość kebabowych liczb pierwszych.

Dla danych z pliku `kebab_przyklad.txt` poprawna odpowiedź to:

3 3

Zadanie 3.3. (0-2)

Liczbą kebabową typu „mieciany mieciany" nazywamy liczbę kebabową, która podczas procesu zwijania miała tyle samo liczb parzystych jak i nieparzystych. Ile jest liczb kebabowych typu „mieciany mieciany" w pliku?

Dla danych z pliku `kebab_przyklad.txt` poprawna odpowiedź to:

6

Zadanie 3.4. (0-2)

Liczbą kebabową typu „falafel" nazywamy liczbę kebabową, której suma dzielników właściwych (czyli mniejszych od tej liczby) jest równa tej liczbie. Ile jest liczb kebabowych typu „falafel" w pliku?

Dla danych z pliku `kebab_przyklad.txt` poprawna odpowiedź to:

Oceny oddajesz:

Plik tekstowy `wyniki3.txt` zawierający odpowiedzi do zadań 3.1.-3.4. (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
Plik(i) zawierający(e) kody źródłowe o nazwie(-ach)
(Uwaga – jeśli nie oddasz plików, zadanie zostanie ocenione na 0 punktów)

Zadanie 3.1. kebab-solve.py

Zadanie 3.2. kebab-solve.py

Zadanie 3.3. kebab-solve.py

Zadanie 3.4. kebab-solve.py

niby spoko, trochę kojarzy się z ..., f a l a f e l [...]" ~

Zadanie 4. (0-2)

Oceń poprawność poniższych zdań. Zaznacz **P** jeśli zdanie jest poprawne, a **F** jeśli zdanie jest fałszywe.

1.	Model warstwowy TCP/IP składa się z większej ilości warstw niż model ISO/OSI	P	F
2.	Protokół sieciowy IPv6 pozwala określać adresy o długości 128 bitów, a IPv4 – 32 bitów	P	F
3.	DNS służy do zamiany adresów IP na odpowiadające im adresy domenowe Domain Name	P	F
4.	W przypadku topologii gwiazdy wszystkie urządzenia są podłączone do jednego punktu centralnego np. switcha	P	F

Zadanie 5. (0-1)

Dwie osoby: A i B komunikują się między sobą z wykorzystaniem szyfrowania asymetrycznego. Jeśli osoba A wyśle do osoby B dokument zaszyfrowany swoim kluczem prywatnym, to czy osoba B może mieć 100% pewności, że otrzymała wiadomość od osoby A? Zakładamy, że osoba B posiada klucz publiczny osoby A, oraz odrzucamy wszelkie możliwości wykradzenia osobie A jej klucza prywatnego przez osobę poufną. Odpowiedz jednym zdaniem, najpierw stwierdzając poprawność pytania, a następnie uzasadniając swój wybór.

Tak, bo jedynie osoba A mogła swoim prywatnym kluczem zaszyfrować komunikat, bo tylko ona go zna zatem osoba B jest na 100% pewna.

Zadanie 6. Korepetycje

Michał od razu po rozpoczęciu studiowania, aby stać się chociaż w małym stopniu niezależnym finansowo postanowił, że zacznie udzielać korepetycji z przedmiotów które lubi. Aby się nie pogubić, Michał sumiennie zapisywał wszelkie formalności w pliku `kursanci.txt`, gdzie znajduje się 235 wierszy. Znajdują się tam informacje dotyczące udzielanych przez niego korepetycji od października 2025 roku do końca lutego 2026 roku. W każdym wierszu podane są dane opisujące jedną lekcję z kursantem, oddzielone pojedynczymi znakami tabulatora:

- Imię kursanta
- Przedmiot z którego udzielane są korepetycje
- Data korepetycji w formacie dd-mm-rrrr
- Godzina rozpoczęcia zajęć
- Godzina zakończenia zajęć
- Stawka za godzinę (różna w zależności od przedmiotu)

Fragment pliku `kursanci.txt`:

```
Jan          Fizyka       06-10-2025  09:00 11:00 40
Wiktor       Matematyka   06-10-2025  11:30 12:30 50
Agnieszka    Matematyka   07-10-2025  09:00 10:15 50
Katarzyna    Informatyka  07-10-2025  11:00 12:45 60
Zbigniew     Fizyka       07-10-2025  13:30 14:45 40
```

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

Zadanie 6.1. (0-1)

Który z kursantów zapłacił najwięcej za jedną lekcję? Podaj jego imię, przedmiot z którego miał korepetycje oraz ich datę.

Zadanie 6.2. (0-2)

Wykonaj zestawienie w którym pokażesz, ile każdy kursant zapłacił w sumie za korepetycje u Michała. Zestawienie posortuj malejąco względem sumy.

Zadanie 6.3. (0-1)

Michał chce przeprowadzić analizę zadowolenia. Twierdzi, że jeśli kursant rezygnował z korepetycji u niego po jednych ćwiczeniach, to korepetycje mu się nie dobały. Ilu było takich kursantów?

Zadanie 6.4. (0-2)

Michał chce stworzyć u siebie na komputerze foldery dla każdej osoby, w których będzie umieszczał dla nich materiały i zadania domowe. Pomyślał jednak „A co jeśli trafi się parę osób o takich samych imionach?". Myślał, myślał aż w końcu wymyślił! Każdej osobie da specjalny nick, i pod takimi nazwami utworzy foldery. Wpadł na pomysł, aby miały one taką samą formę: 3 pierwsze litery imienia zapisane dużymi literami, 3 pierwsze litery przedmiotu zapisane dużymi literami i liczba **wszystkich** korepetycji, których udzielił na ten moment danej osobie.

Utwórz nick każdej osobie (Uwaga – dana osoba może chodzić na korepetycje z paru przedmiotów naraz, więc będzie miała parę nicków) i utwórz zestawienie z wszystkimi nickami posortowane alfabetycznie, aby Michał w łatwy sposób mógł stworzyć foldery.

[III] [PPP] [K]

Przykład:
Dla Bartka który był na wszystkich korepetycjach dokładnie 13 razy i tworzymy folder dla informatyki nick będzie miał postać BARINF13

Wskazówka:
Do rozwiązania zadania mogą przydać się informacje z zadania 6.3.

Informacja do zadań 6.5. i 6.6.

Michał na początku października miał w portfelu 21,37zł. Ponieważ w weekendy nie ma zajęć na studiach i nie udziela korepetycji, to zjeżdża do swojego rodzinnego domu. Z akademika wyjeżdża w sobotę rano, a wraca w niedzielę wieczorem, gdzie bilet na pociąg kosztuje 10zł zarówno z akademika do domu jak i z domu do akademika. We wtorki chodzi do sklepu zrobić zakupy na resztę tygodnia i wydaje tam 250zł. W każdy czwartek wychodzi ze znajomymi na Miasteczko Studenckie, ale ponieważ jest bardzo rozrzutny wydaje bardzo różne kwoty pieniędzy:

- Jeśli w portfelu ma co najwyżej 500zł to wydaje zawsze 1/5 kwoty w portfelu zaokrągloną w dół do groszy, ale minimalnie 50zł
- Jeśli w portfelu ma więcej niż 500zł ale co najwyżej 600zł to wydaje zawsze 1/2 kwoty w portfelu zaokrągloną w dół do groszy, ale minimalnie 100zł
- Jeśli ma więcej niż 600zł w portfelu to Michał baluje do samego rana, stawia wszystkim „przekąski" i wydaje zawsze 400zł niezależnie od tego ile ma pieniędzy

Ponadto, 15 dnia każdego miesiąca musi zapłacić za pokój w akademiku 600zł. 20 grudnia **zjechał do domu** i aż do 3 stycznia nie zarabiał ani nie wydawał pieniędzy, bo jest to czas przerwy świątecznej.

Scanned with CamScanner

Zadanie 6.5. (0-3)

Ile Michał będzie miał pieniędzy **ostatniego dnia** lutego?

Wskazówka:

*Do rozwiązania zadania **może** być potrzebne **ręczne** przygotowanie danych*

Zadanie 6.6. (0-1)

Na podstawie powyższej symulacji utwórz wykres liniowy ze znacznikami ilustrujący ilość pieniędzy w portfelu Michała na przekroju całego miesiąca. Pamiętaj o czytelnym opisie wykresu – dodaj opisy osi i tytuł wykresu.

Do oceny oddajesz:

- Plik tekstowy `wyniki6.txt` zawierający odpowiedzi do zadań 6.1.-6.5. (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- Plik zawierający wykres do zadania 6.6. o nazwie wykres_portfel.jpg
- Plik(i) zawierający(e) komputerową realizację zadania 6. o nazwie(-ach) (Uwaga – jeśli nie oddasz plików, zadanie zostanie ocenione na 0 punktów)

torki_salve.xlsx

Zadanie 7. Gra

Bartek i Piotrek od zawsze chcieli wydać jakąś grę. Po paru miesiącach udało im się stworzyć prostą grę rytmiczną „Klik!", w której gracze klikają przyciski na klawiaturze w rytm muzyki. Ponieważ jest to gra sieciowa to postawili serwer w swoim pokoju w akademiku. Ostatnio zrobili kopię zapasową bazy danych, a jej zawartość znajduje się w plikach `gracze.txt`, `mapy.txt` i `rozgrywki.txt`. Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w wierszach rozdzielono znakiem średnika.

Plik o nazwie `gracze.txt` zawiera informacje o 20 zarejestrowanych graczach. W każdym wierszu znajduje się:

- ID – numer identyfikacyjny gracza
- Nick gracza
- Data_dołączenia w formacie mm-dd-rrrr
- Kraj gracza

Fragment pliku `gracze.txt`:

```
1;Tapcio;01-12-2020;Polska
2;Rytmiś;03-23-2021;Polska
3;Klikacz69;05-17-2019;Polska
4;Osuwacz;02-08-2022;Polska
```

Plik o nazwie `mapy.txt` zawiera informacje o 30 mapach dostępnych w grze. W każdym wierszu znajduje się:

- ID – numer identyfikacyjny mapy
- Autor piosenki
- Tytuł piosenki
- Trudność mapy w skali od 1 do 10
- BPM – liczba uderzeń na minutę określająca szybkość piosenki
- Max_Punkty – maksymalna liczba punktów doświadczenia jaką może zdobyć gracz podczas rozgrywki na danej mapie

Fragment pliku `mapy.txt`:

```
1;Cysmix i Emilka;Deszcz łez;7;128;1500
2;Solejek;Renatka;7;182;1500
3;Kuba Omsiak;Moja Miłość;5;128;1000
4;Pościgacze;Zgubiłem się;6;170;1200
```

Plik o nazwie `rozgrywki.txt` zawiera informacje o 4000 zarejestrowanych rozgrywkach graczy. W każdym wierszu znajduje się:

- ID_rozgr – numer identyfikacyjny rozgrywki
- ID_gracza – numer identyfikacyjny gracza
- ID_mapy – numer identyfikacyjny mapy
- Data_rozgrywki w formacie mm-dd-rrrr
- Wynik – dokładność gracza w danej rozgrywce wyrażona w procentach

Scanned with CamScanner

Fragment pliku `rozgrywki.txt`:

```
1;13;1;08-08-2018;19.26
2;13;1;08-08-2018;17.66
3;13;1;08-08-2018;26.54
4;13;1;08-11-2018;33.63
```

Z wykorzystaniem powyższych danych oraz dostępnych narzędzi informatycznych wykonaj podane zadania. Wyniki zapisz w pliku tekstowym `wyniki7.txt`. Odpowiedź do każdego zadania poprzedź numerem tego zadania.

Zadanie 7.1. (0-1)

Który z graczy jako pierwszy uzyskał 100% na dowolnej mapie? Jako odpowiedź podaj datę, nick gracza, nazwę mapy i jej trudność.

Zadanie 7.2. (0-2)

Wykonaj zestawienie, w którym przedstawisz ile rozgrywek odbyło się w każdym miesiącu 2020 roku. Zestawienie posortuj malejąco względem liczby rozgrywek.

Zadanie 7.3. (0-3)

Podczas grania mapy gracz po jej ukończeniu może dostać punkty doświadczenia. Jeśli gracz uzyskał mniej niż 30% precyzji, nie dostaje żadnych punktów za rozgrywkę. W przeciwnym wypadku dostaje punkty doświadczenia w zależności od jego precyzji.

Przykład:
Jeśli gracz 1 ukończył mapę A z precyzją 50%, która ma maksymalnie 1000 punktów do zdobycia, to podczas tej rozgrywki zdobędzie ***500*** *punktów.*
Jeśli jednak gracz 1 zagrał później jeszcze raz tą samą mapę i uzyskał precyzję 75%, to dostanie wtedy ***250*** *punktów (bo uzyskał już 500 punktów za wcześniejszą rozgrywkę)*

Ponadto, wyniki są zawsze zaokrąglane **do pełnego punktu**.

Który z graczy ma najwięcej punktów doświadczenia? Jako rozwiązanie podaj jego nick, kraj pochodzenia, datę dołączenia i liczbę punktów doświadczenia.

Zadanie 7.4. (0-2)

Ilu graczy zagrało co najmniej raz mapę „MEGALOMAŁA" autorstwa Tofik Lis, ale nigdy ***nie zagrało*** mapy „Ulica Rockefellera (Turbo Mix)" autorstwa Dziewczyna Getterka?

excel
przekrój całego okresu.

Zadanie 7.5. (0-2)

Ze względu na spory odzew graczy Bartkowi i Piotrkowi udało się zorganizować turniej gry „Klik!". Chcą dodać oficjalne wyniki na swoją stronę internetową, więc zmodyfikowali bazę danych gry i dodali nową tabelę **Turniej**, która zawiera kolumny:

- Miejsce – pozycja gracza w tabeli końcowej
- ID_gracza – numer identyfikacyjny gracza
- PD – liczba punktów doświadczenia zdobyta przez gracza na turnieju
- Nagroda – wysokość nagrody

Zakładamy, że część graczy nie mogła wziąć udziału w turnieju. Zapisz w języku SQL zapytanie, w wyniku którego otrzymasz informację, ilu graczy nie wzięło udziału w turnieju.

Miejsce na zapis zapytania

```
SELECT COUNT(*)
FROM Gracze
WHERE Gracze.ID NOT IN
(SELECT ID_gracza
 FROM Turniej);
```

Do oceny oddajesz:

- Plik tekstowy `wyniki7.txt` zawierający odpowiedzi do zadań 7.1.-7.4. (odpowiedź do każdego zadania powinna być poprzedzona jego numerem)
- Plik(i) zawierający(e) komputerową realizację zadania 7. o nazwie(-ach) (Uwaga – jeśli nie oddasz plików, zadanie zostanie ocenione na 0 punktów)

gra.accdb

MATURA
DIAMENT

MATURA
DIAMENT

MATURA
DIAMENT

Scanned with CamScanner

KARTA ODPOWIEDZI

Numer ID

H	5	3	8

Wypełnia sprawdzający

Nr Zad.	Punkty 0	1	2	3
1.1.	0	1	2	3
1.2.	0	1	2	3
1.3.	0	1	2	
2.1.	0	1	2	3
2.2.	0	1	2	
2.3.	0	1	2	
2.4.	0	1	2	3
3.1.	0	1	2	3
3.2.	0	1	2	
3.3.	0	1	2	
3.4.	0	1	2	
4.	0	1	2	
5.	0	1		

Nr Zad.	Punkty 0	1	2	3
6.1.	0	1		
6.2.	0	1	2	
6.3.	0	1		
6.4.	0	1	2	
6.5.	0	1	2	3
6.6.	0	1		
7.1.	0	1		
7.2.	0	1	2	
7.3.	0	1	2	3
7.4.	0	1	2	
7.5.	0	1	2	

Kod sprawdzającego

Scanned with CamScanner