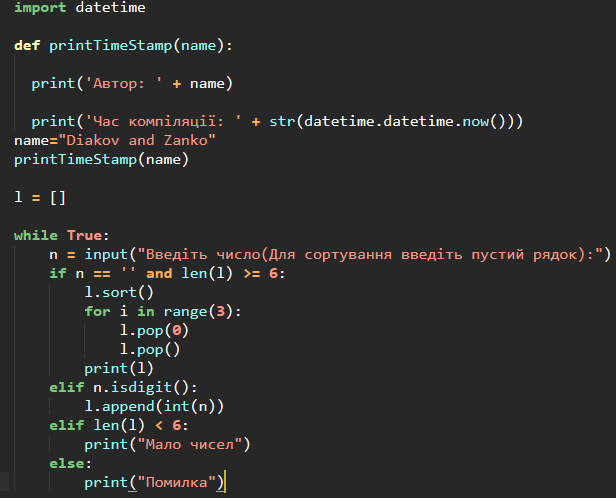
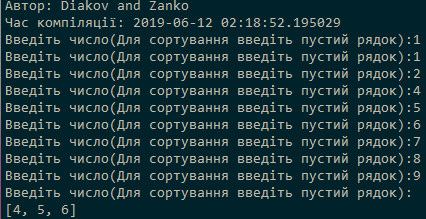
**LEVEL A**

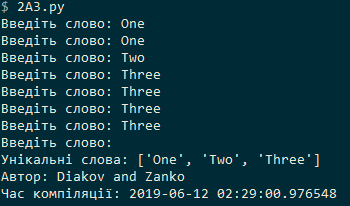
|  |  |
| --- | --- |
| 1. | При аналізі даних, зібраних у результаті експерименту, часто може виникати потреба у видаленні найбільш екстремальних значень до виконння обчислень. Ваша програма має видалити зі списку n (ціле додатнє число, введене користувачем), по троє найбільших та найменших значень і вивести решту значень (порядок не важливий). Також забезпечте коректну обробку ситуації, коли у списку не вистачатиме елементів для видалення. |





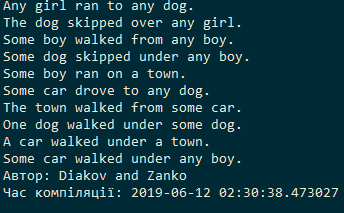
|  |  |
| --- | --- |
| 2. | Напишіть програму, яка зчитує числа від користувача, поки не буде введено порожній рядок. Далі вона повинна знайти середнє значення із введених чисел та вивести 2 списки: введених чисел, що не перевищують середнє значення та, відповідно, більше або рівні за нього. |
| 3. | Напишіть програму, яка зчитує від користувача слова, кожне з нового рядка. Ввід закінчується порожнім рядком, після чого виводиться список всіх унікальних слів, які були введені. Наприклад, для вводу  first  second  first  third  second  програма повинна буде вивести  first  second  third |



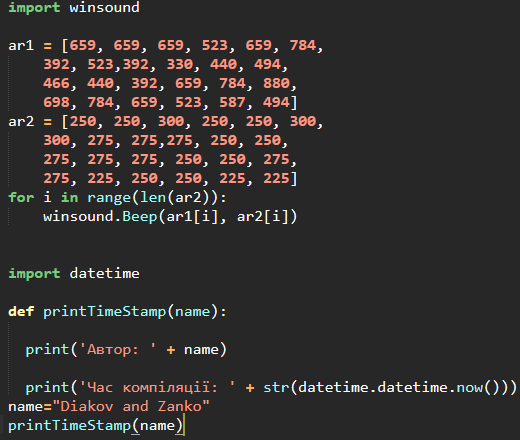


|  |  |
| --- | --- |
| 4. | Напишіть додаток, який використовує генерування випадкових чисел для створення речень. Використовуйте 4 списки рядків: article, noun, verb та preposition. Створіть речення, обравши з кожного масиву слова в такому порядку: артикль, іменник, дієслово, прийменник, артикль та іменник. Після вибору кожного слова виконайте його конкатенацію з реченням. Слова мають відокремлюватись пробілами. Загальне речення має починатись з великої літери та закінчуватись крапкою. Додаток має згенерувати та відобразити 10 речень.  *Артиклі:* "the", "a", "one", "some" та "any";  *Іменники:* "boy", "girl", "dog", "town" та "car";  *Дієслова*: "drove", "jumped", "ran", "walked" та "skipped";  *Прийменники*: "to", "from", "over", "under" та "on". |



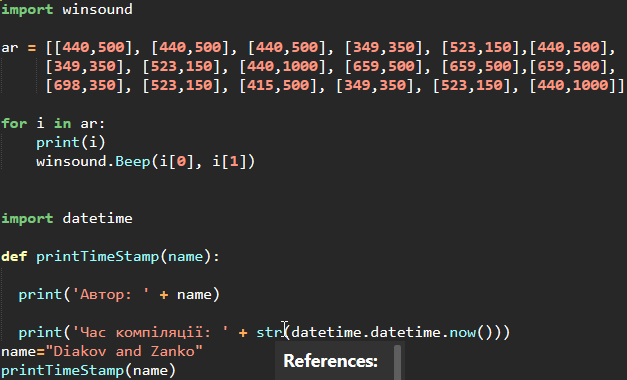


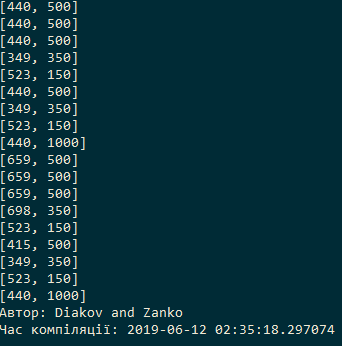
|  |  |
| --- | --- |
| 5. | Нехай грає музика! Запрограмуйте комп’ютер на відтворення мелодії, заданої одновимірним масивом частот та одновимірним масивом тривалостей відтворення відповідної частоти. Відповідні пари значень наведені нижче:  (659,250), (659,250), (659,300), (523,250), (659,250), (784,300), (392,300), (523,275), (392,275), (330,275), (440,250), (494,250), (466,275), (440,275), (392,275), (659,250), (784,250), (880,275), (698,275), (784,225), (659,250), (523,250), (587,225), (494,225).  Для підключення біперу потрібно імпортувати модуль  import winsound  і викликати функцію  winsound.Beep(частота у Гц, тривалість у мс) |



**Программа виводится у звуковому форматі**

|  |  |
| --- | --- |
| 6. | Якщо Вам сподобалась попередня задача, спробуйте тепер іншу мелодію, збережену в *двовимірному* масиві (списку списків):  (440,500), (440,500), (440,500), (349,350), (523,150), (440,500), (349,350), (523,150), (440,1000), (659,500), (659,500), (659,500), (698,350), (523,150), (415,500), (349,350), (523,150), (440,1000) |

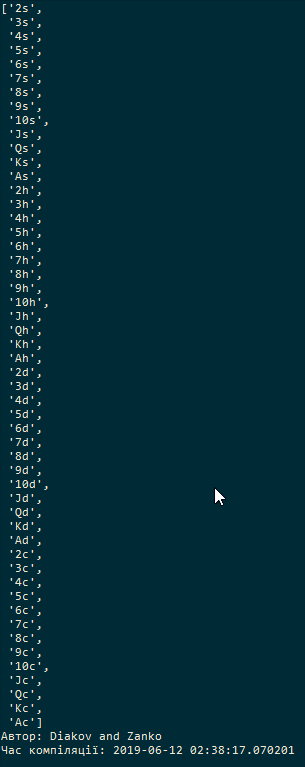




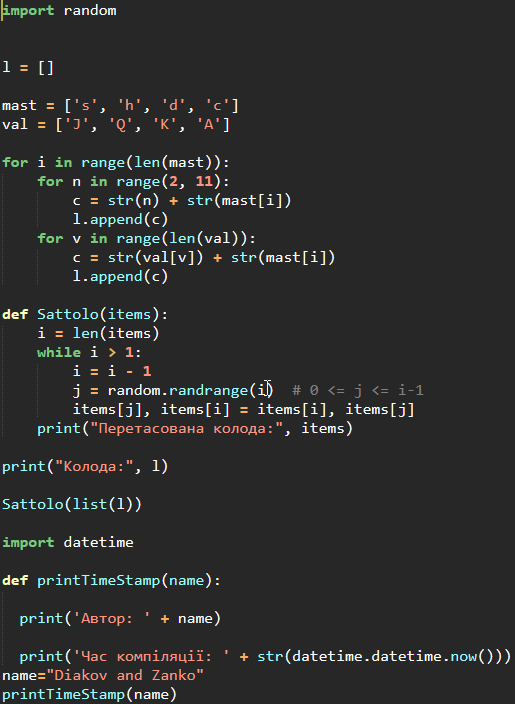
**Програма виводить частоту та грає мелодію**

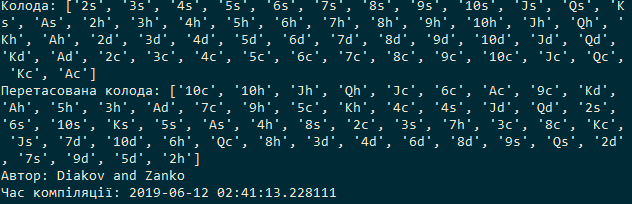
|  |  |
| --- | --- |
| 7. | Звичайна колода карт має 52 карти, кожна з яких описується мастю і значенням. Мастями можуть бути піки (spades), черви (hearts), трефи (diamonds) та бубни (clubs), а значеннями – числа від 2 до 10, валет (Jack), дама (Queen), король (King) і туз (Ace). Приклади позначень карт описані в таблиці.    Напишіть програму, яка буде створювати колоду карт у вигляді списку. |



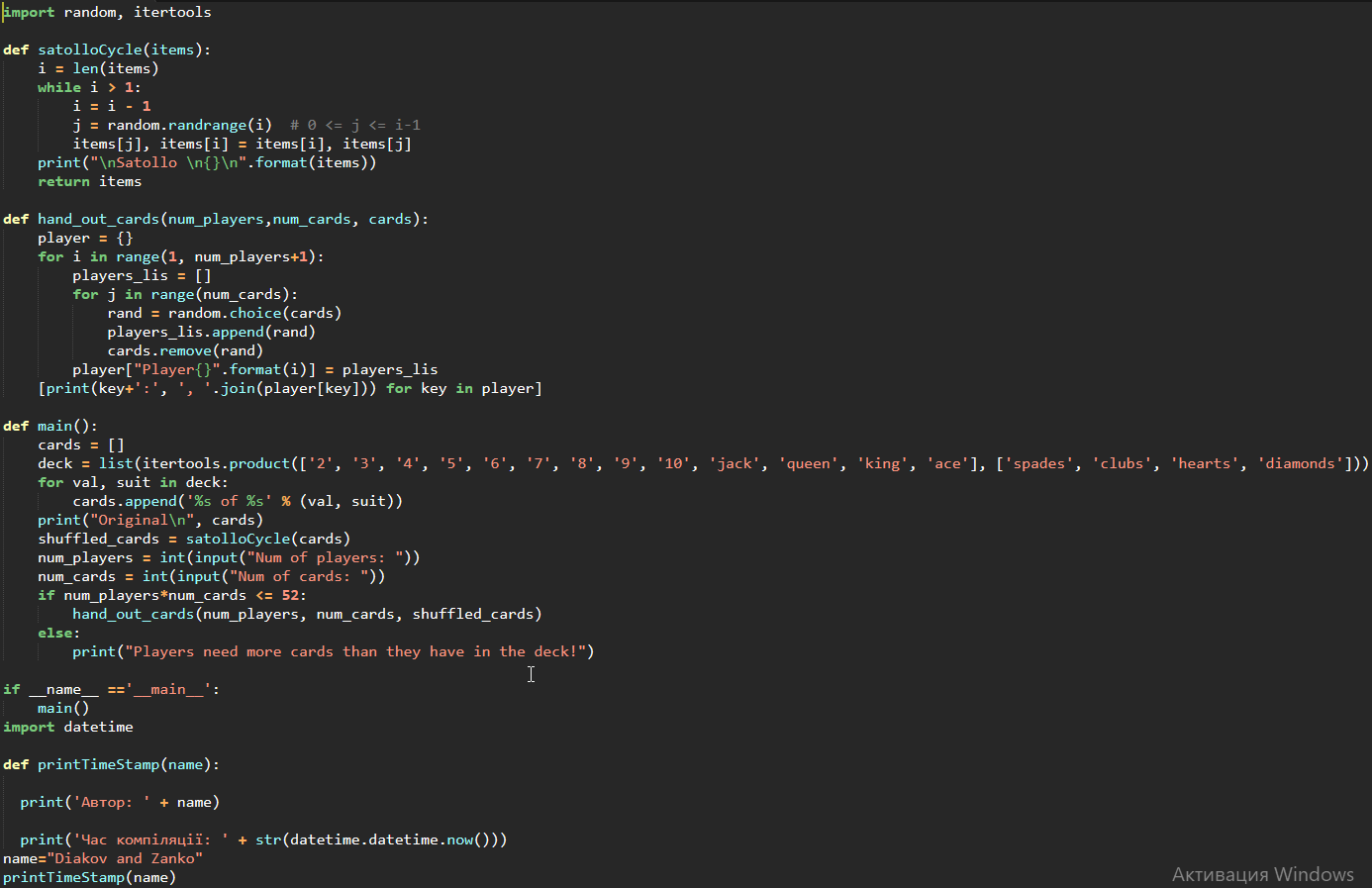


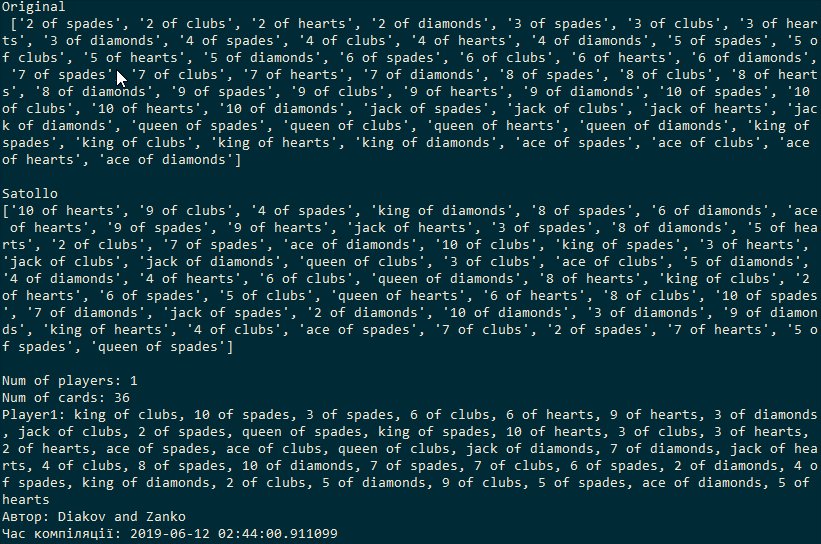
|  |  |
| --- | --- |
| 8. | Алгоритми тасування колоди карт, побудованих на базі алгоритму Фішера-Йетса описані [тут](https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D1%81%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%A4%D0%B8%D1%88%D0%B5%D1%80%D0%B0_%E2%80%94_%D0%99%D0%B5%D1%82%D1%81%D0%B0). Створіть колоду карт, яку можна буде потасувати за допомогою алгоритму Сатолло. Виведіть початкову колоду та перетасовану на екран. |



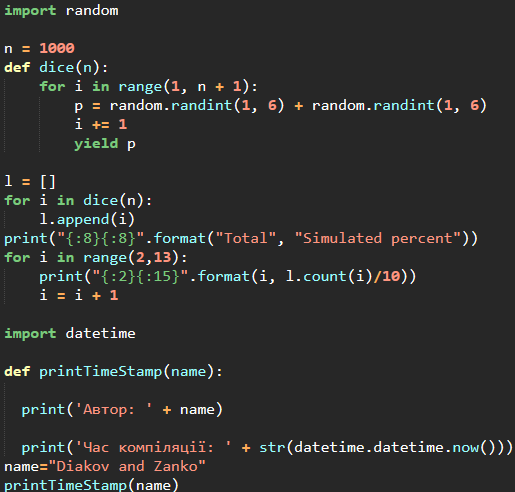


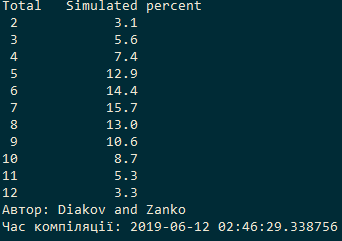
|  |  |
| --- | --- |
| 10. | Створивши та перетасувавши колоду в попередніх задачах, прийшов час здати їх гравцям. Кількість гравців та карт для кожного з них залежить від гри, тому користувач повинен після перетасовки ввести кількість карт на одного гравця та кількість гравців. За цими даними потрібно вивести інформацію зі списку про гравців та карти, що їм були здані. Забезпечте коректну роботу програми, коли карт для гравців потрібно більше, ніж є в колоді. |





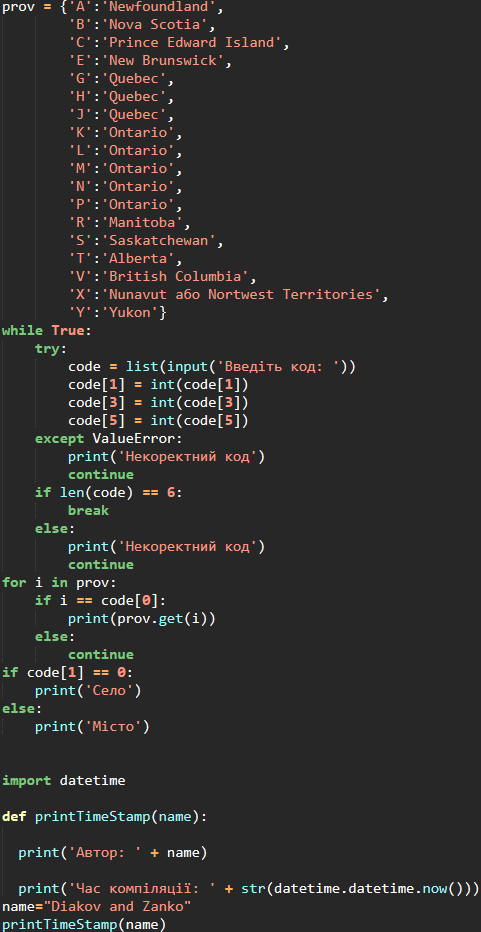
|  |  |
| --- | --- |
| 11. | Напишіть програму, яка симулюватиме 1000 кидків пари костей та виводитиме статистику сум, які випадають у відсотках. Оформити результат можна так:    Очікуваний відсоток – стандартне значення, отримане з розподілу Гауса. |

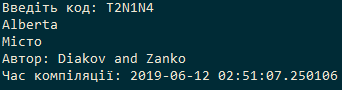




**LEVEL B**

|  |  |
| --- | --- |
| 1. | Канадський поштовий код складається з 6 символів: перший, третій і п’ятий з них букви, а решта – цифри. Провінцію можна визначити за першим символом коду (таблиця). Букви D, F, I, O, Q, U, W та Z не можуть бути першими.    Другий символ позначає сільську місцевість (0) або місто (інакше).  Напишіть програму, яка зчитує поштовий код та визначає провінцію та місцевість проживання. Наприклад, T2N1N4 вказує на проживання в місті провінції Альберта. Якщо введено X0A1B2, поштовий код вказує на село в Nunavut або Northwest Territories. Використовуйте словник для відображення перших символів коду в назву провінції. Забезпечте коректну роботу програми для некоректного коду. |





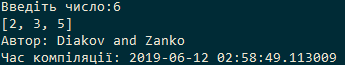
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2. | На більшості кнопочних телефонів для набору текстового повідомлення використовують цифрову клавіатуру. Оскільки кожна кнопка прив’язана до кількох букв, її потрібно натискати декілька разів для вибору потрібної відповідно до таблиці. Натиснення цифри 2, 3, 4 або 5 разів генерує другу, третю, четверту або п’яту букву для цієї кнопки.   |  |  | | --- | --- | | **Кнопка** | **Символи** | | 1 | . , ? ! : | | 2 | A B C | | 3 | D E F | | 4 | G H I | | 5 | J K L | | 6 | M N O | | 7 | P Q R S | | 8 | T U V | | 9 | W X Y Z | | 0 | пробіл |   Напишіть програму, яка показує, скільки та яких натиснень кнопок потрібно, щоб набрати текст користувача. Сконструюйте словник, який відображає кожну букву чи символ на кнопку, яку потрібно натиснути. Наприклад, для тексту Hello, World! програма має вивести 4433555555666110966677755531111. Забезпечте коректну роботу як з малими, так і великими літерами. Ігноруйте символи, не представлені в таблиці. |





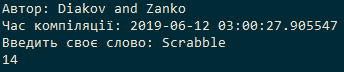
|  |  |
| --- | --- |
| 4. | Решето Ератосфена – алгоритм, який дозволяє знайти прості числа, що не перевищують деякого n. Він був придуманий понад 2000 років тому і описується так:  Запишіть всі числа від 0 до n  Викресліть 0 і 1, тому що вони не є простими  Задайте p = 2  While p < n do  Викреслити всі числа, кратні p (проте не сам p)  Перейти (змінна p) до наступного числа в списку, яке  не буде закресленим  Вивести всі незакреслені числа.  Сформуйте список усіх чисел від 2 до введеного значення n, а потім видаляйте з нього всі числа, які викреслює алгоритм. У результаті повинен отриматись список із простих чисел для введеного діапазону.  Підказка. Пришвидшити роботу початкового алгоритму можна за рахунок неповного перебору чисел. Насправді, достатньо перевірити числа не до n, а лише до . |



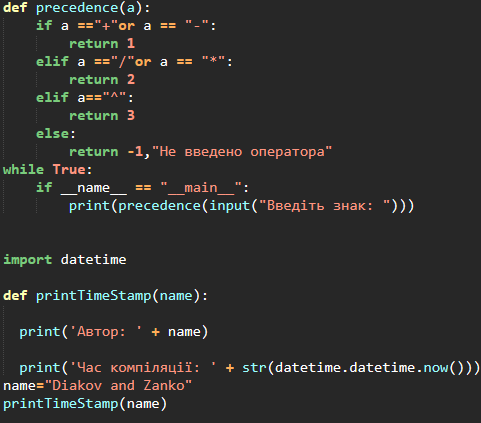


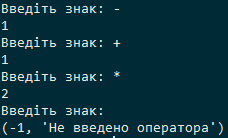
|  |  |
| --- | --- |
| 5. | У грі Scrabble™ кожна буква оцінюється балами, як показано в таблиці. Загальна оцінка слова – це сумарна кількість очків за кожну букву:    Напишіть програму, яка обчислює та виводить Scrabble™-оцінку для слова. Створіть словник, який відображає букви на бальні значення, а потім з його допомогою аналізуйте введені слова, виводячи їх оцінку в балах.  У грі Scrabble™ включено деякі squares, які домножають значення букви чи всього слова на певні значення. Це правило в задачі ігнорується. |





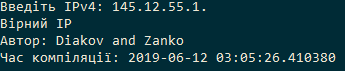
|  |  |
| --- | --- |
| 8. | Напишіть функцію precedence(), яка повертає ціле число, що показує пріоритет математичного оператора. Рядок, який містить оператор, передається як єдиний параметр функції. Сама функція має повертати 1 для + та -, 2 для \* та /, 3 для ˆ. Якщо переданий рядок не має цих операторів, функція має повернути -1.  Основна програма буде зчитувати оператор від користувача та або виводить пріоритет оператора, або повідомлення, що не було введено оператор. Your main program should only run when the file containing your solution has not been imported into another program. |



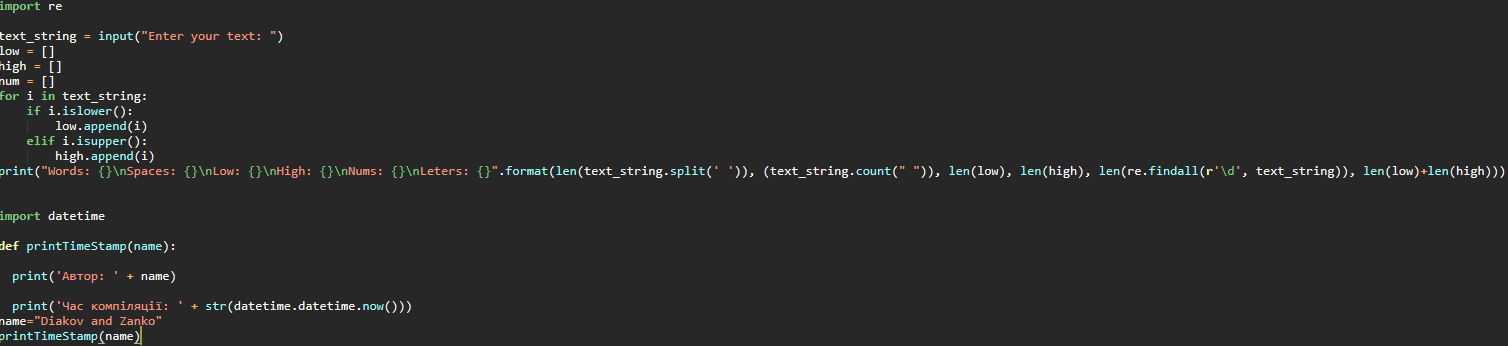


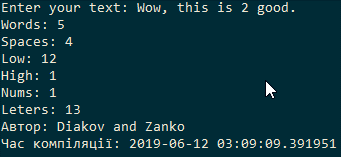
|  |  |
| --- | --- |
| 10. | Напишіть програму, яка зчитує IPv4 адресу та перевіряє її валідність. Форма запису IPv4 адреси – x.x.x.x, де кожен x має бути цілим числом з діапазону [0, 255]. |





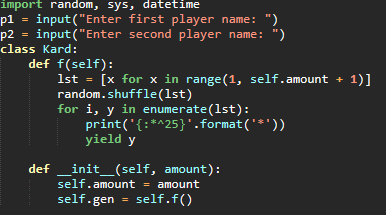
|  |  |
| --- | --- |
| 11. | Введіть з клавіатури деякий текст та підрахуйте в ньому кількість слів, букв, цифр,малих літер, великих літер, пробілів. |

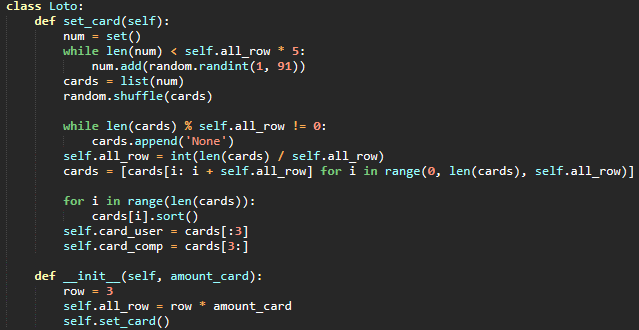


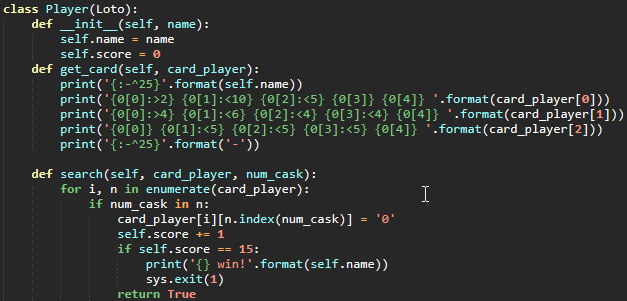


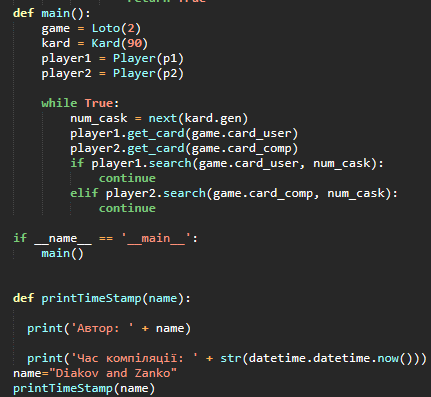
**LEVEL C**

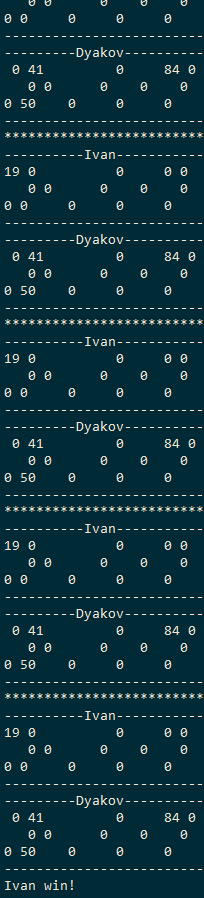
|  |  |
| --- | --- |
| 4. | Виграшна картка лото містить лінію з 5 чисел, які до цього були витягнуті з мішка. Гравці позначають числа, які були витягнуті, закриваючи їх. У нашому додатку будемо заміняти число на 0 у лото-словнику, якщо воно було оголошено.  Напишіть програму, яка працює зі словниками, що представляють картки лото. Якщо картка містить рядок з 5 нулів, вона вважається виграшною. Програма має формувати 2 картки лото (9 стовпців чисел, кожен з яких відповідає десятку числа, 3 рядка, по 5 чисел у кожному), генерувати випадкове число та заново виводити картку після цього. Як тільки якась із карток виграє, програма завершує роботу з повідомленням про виграш відповідного гравця.  http://s001.radikal.ru/i193/1107/87/916146c47dd1.jpg  Підказка. Оскільки всі числа додатні, знаходження 5 нулів на картці відповідає задачі рівності суми 5 входжень нулю. Можливо, Вам так буде простіше моделювати гру. |











|  |  |
| --- | --- |
| 5. | Напишіть гру «Однорукий бандит» для консолі. У процесі будуть відображатись три випадкових символи з переліку:  Cherry - u"\U0001F352",  Bell - u"\U0001F514",  Lemon - u"\U0001F34B",  Orange - u"\U0001F34A",  Star - u"\u2606",  Skull - u"\U0001F480"  Гравцю видається кредит у розмірі 100 грн., а кожний запуск коштує 5 грн. Якщо «однорукий бандит» показує два однакових символи, користувач виграє 10 грн, а якщо три – 25 грн. У випадку випадіння трьох дзвіночків (bells) сума виграшу складає 100 грн. Якщо випадає два черепи – гравець додатково втрачає 5 грн, а при отриманні трьох черепів вся сума згорає. Гравець може зняти вигране до початку нової спроби або продовжувати грати, поки на рахунку будуть гроші. |

