МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ

Відділення інформаційних технологій

Звіт

з навчальної практики №1

Студента групи 1П-17

Глигало В.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник практики

Видриган-Лаврук А. М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Черкаси, 2019

ОП

**День 1**

1. Напишіть програму, яка обчислює індекс маси тіла (ІМТ) для особи. Спочатку додаток має зчитувати зріст та масу користувача. Для обчислення ІМТ використовується одна з 2 формул. Якщо зріст вимірюється в дюймах, а маса – у фунтах, ІМТ визначається так:

Для метрів та кілограмів ІМТ виражається відношеням

Виведіть відповідне значення ІМТ; вони будуть аналізуватись у наступних задачах.

import datetime

def printTimeStamp(name):

print('Автор програми: ' + name)

print('Час компіляції: ' + str(datetime.datetime.now()))

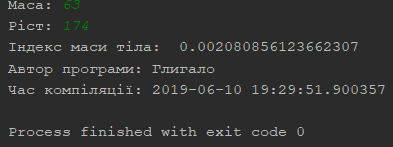
m = int(input("Маса: "))

l = int(input("Ріст: "))

imt = m/(l\*l)

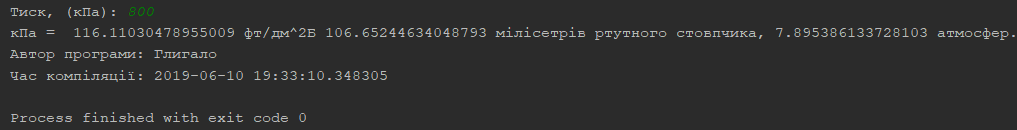
print("Індекс маси тіла: ", imt)

printTimeStamp("Глигало")



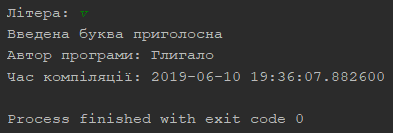
2. Створіть програму, яка зчитує тиск у кПа (кілопаскалях). Після зчитування даних, програма повинна повідомити про еквівалентний тиск у фт/дм2 (фунтах на квадратний дюйм), міліметрах ртутного стовпчика та атмосферах.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
P = int(input("Тиск, (кПа): "))  
ftdm = P/6.89  
atm = P/101.325  
mm = P/7.501  
print("кПа = ", ftdm, "фт/дм^2Б", mm, "мілісетрів ртутного стовпчика,", atm,"атмосфер.")  
printTimeStamp("Глигало")



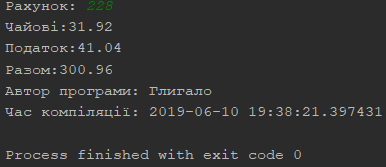
3. Створіть програму, яка зчитує букву англійського алфавіту від користувача. Якщо користувач ввів a, e, i, o або u, програма повинна вивести повідомлення, що введена буква – голосна. Якщо користувач вводить y, програма має повідомити, що інколи y – голосна, а інколи - приголосна. Інакше слід вивести повідомлення, що буква приголосна.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
Letter = input("Літера: ")  
if Letter == "a" or Letter =="e" or Letter =="i" or Letter =="o" or Letter =="u":  
 print("Введена буква голосна")  
elif Letter == "y":  
 print("Ця буква може бути голосною або приголосною")  
else:  
 print("Введена буква приголосна")  
printTimeStamp("Глигало")



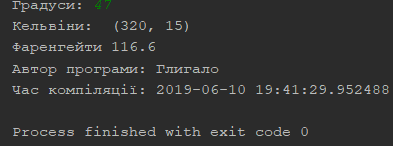
4. Програма в цій вправі повинна починати роботу зі зчитування вартості їжі, замовленої з ресторану (назва страви, кількість порцій, ціна за порцію). Потім програма обчислить податок та чайові за їжу. Розраховуйте на чайові в розмірі 14% від замовлення та податок у розмірі 18% від замовлення. Вивід вашої програми має включати розмір податку, чайових та загальну суму для оплати. Формат виводу передбачає представлення всіх значень з використанням 2 знаків після коми.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
a = int(input("Рахунок: "))  
x = a \* 0.14  
y = a \* 0.18  
z = x + y + a  
print("Чайові:{:.2f}".format(x))  
print("Податок:{:.2f}".format(y))  
print("Разом:{:.2f}".format(z))  
printTimeStamp("Глигало")



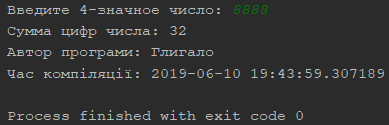
1. Напишіть програму, яка спочатку зчитує температуру в градусах Цельсія. Далі вона має вивести еквіваленту температуру в градусах Кельвіна та Фаренгейта

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
C = int(input("Градуси: "))  
K = C + 273,15  
F = C\*9/5 + 32  
print("Кельвіни: ", K)  
print("Фаренгейти", F)  
printTimeStamp("Глигало")



6. Напишіть програму, яка зчитуватиме 4-значне ціле число та буде виводити суму його цифр. Наприклад, якщо ввести 3141, програма має вивести 9 (=3+1+4+1).

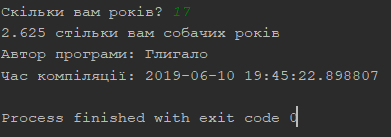
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
n = input("Введите 4-значное число: ")  
l = list(n)  
d1 = int(l[0])  
d2 = int(l[1])  
d3 = int(l[2])  
d4 = int(l[3])  
print("Сумма цифр числа:", d1 + d2 + d3+ d4 )  
printTimeStamp("Глигало")



7. Поширеною думкою є те, що 1 рік людського життя еквівалентний 7 рокам життя собами. Проте це просте правило перестає виконуватись, коли собака стає дорослою (приблизно в 2 роки). У результаті деякі люди вважають, що краще вважати перші 2 роки життя собаки еквівалентними 10.5 рокам життя людини, а наступні – 4м рокам людини.

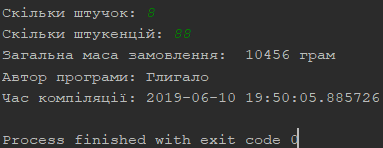
Напишіть програму, яка реалізує переведення з людських років у собачі за даними правилами. Ваш додаток має виводити попередження, коли вводиться від’ємне число.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
years = float(input("Скільки вам років? "))  
if years < 1:  
 print("Bruh")  
elif years < 10.5:  
 print(years/10.5, "стільки вам собачих років")  
else:  
 print(years//10.5 + years % 10.5 / 4, "стільки вам собачих років")  
printTimeStamp("Глигало")



8. Онлайн-ритейлер продає 2 товари: штучки та штукенції. Кожна штучка важить 75г. Кожна штукенція важить 112г. Напишіть програму, яка зчитує кількість штучок та кількість штукенцій в замовленні від користувача. Потім програма має обчислювати та виводити на екран загальну масу замовлення.

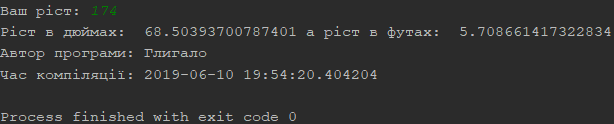
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
Sht = int(input("Скільки штучок: "))  
Shtyk = int(input("Скільки штукенцій: "))  
print("Загальна маса замовлення: ", Sht\*75 + Shtyk\*112, "грам")  
printTimeStamp("Глигало")



9. Багато людей думають про свій зріст у футах і дюймах, але у нас використовується метрична система. Напишіть програму, яка зчитує зріст у сантиметрах, а виводить у футах і дюймах.

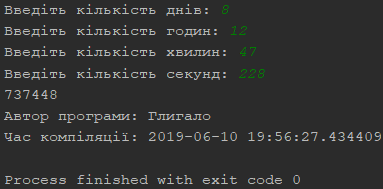
Підказка: 1 фут = 12 дюймів. 1 дюйм = 2.54 см.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
l = int(input("Ваш ріст: "))  
print("Ріст в дюймах: ", l/2.54, "а ріст в футах: ", l/2.54/12 )  
printTimeStamp("Глигало")



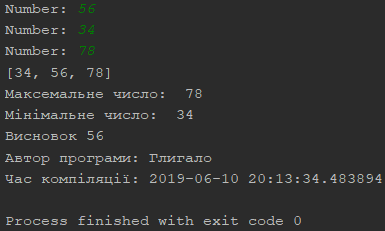
10. Напишіть програму, яка зчитує проміжок часу у вигляді кількості днів, годин, хвилин та секунд. Обчисліть та виведіть загальну кількість секунд у цьому проміжку часу.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
d = float(input("Введіть кількість днів: "))  
g = float(input("Введіть кількість годин: "))  
h = float(input("Введіть кількість хвилин: "))  
s = float(input("Введіть кількість секунд: "))  
d1 = 86400  
g1 = 3600  
h1 = 60  
print (int(d\*d1 + g\*g1 + h\*h1+ s))  
printTimeStamp("Глигало")



11. Напишіть програму, яка зчитує 3 цілих числа і сортує їх у порядку від меншого до більшого. Використовуйте функції min() та max() для знаходження найменшого та набільшого значень. Значення між ними можна обчислити, знайшовши суму всіх 3 чисел та віднявши мінімум та максимум.

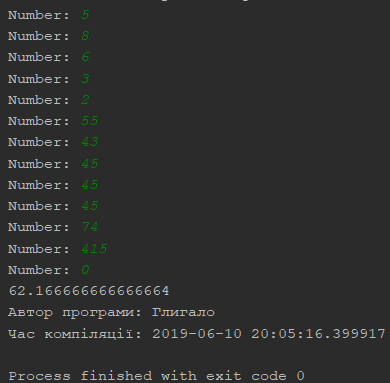
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
l = []  
for i in range(3):  
 l.append(int(input("Number: ")))  
l.sort()  
print(l)  
print("Максемальне число: ", max(l))  
print("Мінімальне число: ", min(l))  
print("Висновок", sum(l)-max(l)-min(l))  
printTimeStamp("Глигало")



12. Створіть програму, яка обчислює середнє для набору значень, введеного користувачем. Користувач буде вводити 0 як індикатор того, що більше значень вводитись не буде. Ваша програма має виводити відповідне повідомлення про помилку, якщо перше введене значення буде нулем. Ця вправа не передбачає використання непримітивних типів даних.

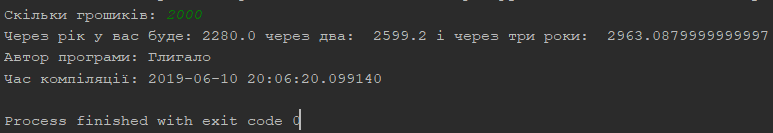
Підказка: оскільки 0 позначає кінець вводу, він має не включатись в обчислення середньго.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
l = []  
while True:  
 s = int(input("Number: "))  
 l.append(s)  
 if s == 0:  
 break  
l.pop()  
def midlle(numbers):  
 return float(sum(numbers)) / max(len(numbers), 1)  
print(midlle(l))  
printTimeStamp("Глигало")



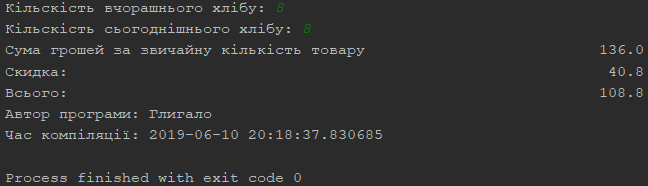
13. Уявімо, що Ви тільки-но відкрили новий накопичувальний рахунок зі ставкою 14% річних. Отриманий прибуток виплачується в кінці року та додається до балансу рахунку. Напишіть програму, що починає роботу зі зчитування кількості покладених на рахунок грошей. Далі вона має обчислити та вивести на екран суму на рахунку через 1, 2 і 3 роки. Показ сум обмежується 2 знаками після крапки.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
Money = int(input("Скільки грошиків: "))  
print("Через рік у вас буде:", Money\*0.14 + Money, "через два: ", (Money\*0.14 + Money)\*0.14 + (Money\*0.14 + Money),  
 "і через три роки: ",((Money\*0.14 + Money)\*0.14 + (Money\*0.14 + Money))\*0.14 + ((Money\*0.14 + Money)\*0.14 + (Money\*0.14 + Money)))  
printTimeStamp("Глигало")



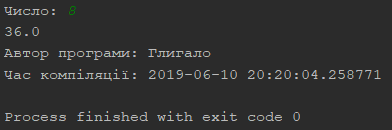
14. Пекарня продає буханки хлібу за 8.50 грн. На вчорашній хліб встановлено знижку на 60%. Напишіть програму, яка починає зі зчитування кількості буханок вчорашнього хлібу, які хоче придбати користувач. Потім програма має вивести на екран звичайну вартість товару, скидку за те, що він вчорашній, а також загальну суму покупки. Всі значення мають виводитись двома десятковими знаками після крапки та вирівнюватись по правому краю.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
Bread\_1 = int(input("Кільскість вчорашнього хлібу: "))  
Bread\_2 = int(input("Кільскість сьогоднішнього хлібу: "))  
print("Cума грошей за звичайну кількість товару {0:>30}" .format((Bread\_1 + Bread\_2)\*8.50 ))  
print("Скидка: {0:>63}" .format((Bread\_1)\*8.50\*0.6))  
print("Всього: {0:>63}" .format((Bread\_1)\*8.50\*0.6 + Bread\_2\*8.50))  
printTimeStamp("Глигало")



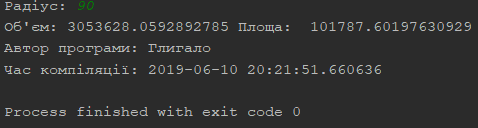
15. Напишіть програму, яка зчитує ціле число n від користувача та показує суму всіх цілих чисел від 1 до n. Формула для знаходження цієї суми:

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
n = int(input("Число: "))  
print(n\*(n+1)/2)  
printTimeStamp("Глигало")



16. Напишіть програму, яка починає зі зчитування радіусу r від користувача. Програма продовжуватиме роботу, обислюючи та виводячи на екран площу кругу з радіусом r та об’єм кулі з радіусом r. Використовуйте константу pi з модуля math.

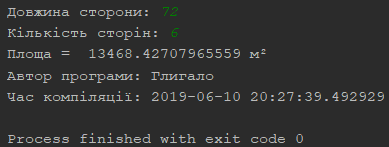
import datetime  
from math import pi  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
r = int(input("Радіус: "))  
print("Об'єм:", (4/3)\*pi\*r\*\*3, "Площа: ", 4\*pi\*r\*\*2)  
printTimeStamp("Глигало")



17. Полігон називають регулярним, якщо всі його сторони мають однакові довжини, а кути між усіма суміжними сторонами теж однакові. Площу регулярного полігону можна обчислити за формулою

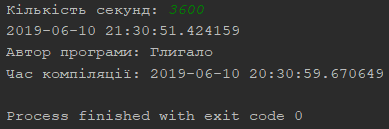
де s – довжина сторони, а n – кількість сторін. Напишіть програму, яка зчитує ці значення і виводить на екран площу сконструйованого за даними значеннями полігону.

import math  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
s = int(input("Довжина сторони: "))  
n = int(input("Кількість сторін: "))  
print("Площа = ",(n\*s\*\*2)/(4\*math.tan(math.pi/n)), "м²")  
printTimeStamp("Глигало")



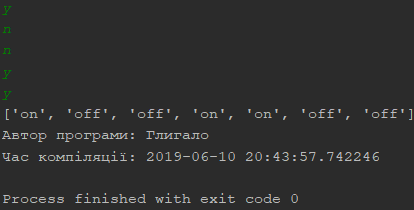
18. Напишіть програму, яка починає роботу зі зчитування кількості секунд від користувача. Далі вона має вивести еквівалентний проміжок часу у формі Д:ГГ:ХХ:СС (дні, години, хвилини, секунди). Потрібно, щоб форматований вивід передбачав по 2 знакових місця для годин, хвилин та секунд з дописуванням незначущого нуля за потреби.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
now = datetime.datetime.now()  
print(now + datetime.timedelta(seconds=int(input("Кількість секунд: "))))  
printTimeStamp("Глигало")



19. Припустимо, розумна програма може не вмикати будильник, щоб людина виспалась на вихідних чи у відпустці. Розробіть додаток, який буде приймати на вхід від користувача позначки про вихідний та відпустку, а виводитиме рішення, чи вмикати будильник. Вмикати його не слід, якщо день вихідний або людина у відпустці.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
s=[]  
n=0  
for i in range(5):  
 if input() == "n":  
 s.append("off")  
 else:  
 s.append("on")  
 n + 1  
for i in range(2): s.append("off")  
print(s)  
printTimeStamp("Глигало")

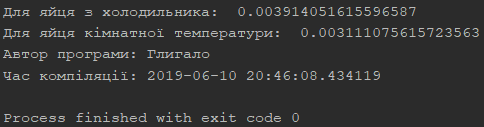


21. **Готуємо ідеальне яйце.** Під час варіння протеїни яйця спочатку денатурують (змінюють природну структуру), а потім коагулюють (згортаються). Коли температура досягає критичної точки, хімічні реакції розпочинаються і протікають швидше з ростом температури. Білок яйця коагулює при температурі близько 63оC, а жовток – за температури понад 70оC. Для варки рідкого яйця потрібно достатньо довго нагрівати його при температурі від 63 оC до 70оC. Для приготування крутого яйця дозволено досягати температури понад 70 оC.

Наступна формула виражає час (в секундах), коли центр жовтка досягне температури (у градусах Цельсія):

де M – маса яйця, – густина, – питома теплоємність, а K – теплопровідність яйця. Релевантними значеннями для малого яйця є маса M = 47г та M = 67г для великого яйця, 1.038 , , and . За умовами задачі – температура киплячої води (у градусах Цельсія), а – початкова температура (у градусах Цельсія) яйця перед його покладанням у воду. Реалізуйте формулу в програмі, задавши , та обчисліть для великого яйця, взятого з холодильника () та при кімнатній температурі ().

import datetime  
import math  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
print("Для яйця з холодильника: ", (67\*\*(2/3)\*3.7\*1.038\*\*(1/3))/((5.4\*10\*\*3)\*(4\*math.pi/3)\*\*(2/3))\*math.log(0.76\*((4-100)/(70-100))))  
print("Для яйця кімнатної температури: ", (67\*\*(2/3)\*3.7\*1.038\*\*(1/3))/((5.4\*10\*\*3)\*(4\*math.pi/3)\*\*(2/3))\*math.log(0.76\*((20-100)/(70-100))))  
printTimeStamp("Глигало")



**Рівень Б**

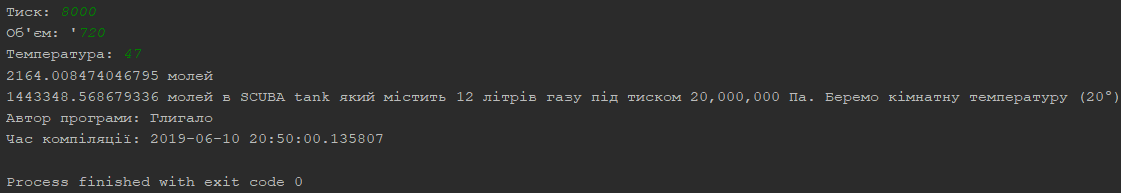
1. Закон ідеального газу – математичне наближення поведінки газів при зміні тиску, об’єму та температури. Зазвичай він описується рівнянням

де P – тиск у паскалях (Па), V – об’єм у літрах, n – молярна маса речовини, R = 8.314 mol K J – константа ідеального газу, T – температура в градусах Кельвіна.

Напишіть програму, яка обчислює молярну масу газу, коли користувач задає тиск, об’єм і температуру. Протестуйте свою програму на визначення кількості молей газу в SCUBA tank. Типовий SCUBA tank містить 12 літрів газу під тиском 20,000,000 Па. Беремо кімнатну температуру (20º).

Підказка: Температура конвертується з Цельсія в Кельвіни шляхом додавання 273.15 градусів.

import datetime  
#import math  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
p = int(input("Тиск: "))  
v = int(input("Об'єм: '"))  
t = int(input("Температура: "))  
t = t+273.15  
print((p\*v)/(8.314\*t), "молей")  
print((20000000\*12)/(8.314\*20), "молей в SCUBA tank який містить 12 літрів газу під тиском 20,000,000 Па. Беремо кімнатну температуру (20º)")  
printTimeStamp("Глигало")

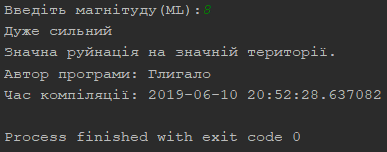


2. Нижче наведена таблиця містить діапазони магнітуд землетрусів за шкалою Ріхтера та опис відповідного землетрусу:

|  |  |
| --- | --- |
| Магнітуда | Опис |
| <2.0 | Мікро (micro) |
| 2.0-3.0 | Дуже слабкий (very minor) |
| 3.0-4.0 | Слабкий (minor) |
| 4.0-5.0 | Легкий (light) |
| 5.0-6.0 | Помірний (moderate) |
| 6.0-7.0 | Сильний (strong) |
| 7.0-8.0 | Дуже сильний (major) |
| 8.0-10.0 | Великий (great) |
| >=10.0 | Рідкісно великий (meteoric) |

Напишіть програму, яка зчитує від користувача магнітуду та виводить відповідний опис. Докладніше [тут](https://uk.wikipedia.org/wiki/%D0%A8%D0%BA%D0%B0%D0%BB%D0%B0_%D0%A0%D1%96%D1%85%D1%82%D0%B5%D1%80%D0%B0).

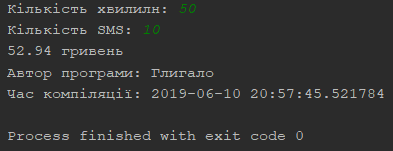
import datetime  
#import math  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
Mg = float(input('Введіть магнітуду(ML):'))  
if Mg <= 2.0 :  
 print('Мікро\nМікроземлетруси, не відчуваються.')  
elif Mg >= 2.0 and Mg <= 3.0 :  
 print('Дуже слабкий\nЯк правило не відчуваються, але реєструються.')  
elif Mg >= 3.0 and Mg <= 4.0 :  
 print('Слабкий\nЧасто відчуваються, дуже рідко завдають шкоди.')  
elif Mg >= 4.0 and Mg <= 5.0 :  
 print('Легкий\nВідчутне тремтіння речей всередині будинків, значна шкода малоймовірна.')  
elif Mg >= 5.0 and Mg <= 6.0 :  
 print('Помірний\nМоже завдати значної шкоди старим та погано сконструйованим будівлям на незначній території.')  
elif Mg >= 6.0 and Mg <= 7.0 :  
 print('Сильний\nМоже спричинити руйнацію на території до 150 км довжиною/шириною в населених регіонах.')  
elif Mg >= 7.0 and Mg <= 8.0 :  
 print('Дуже сильний\nЗначна руйнація на значній території.')  
elif Mg >= 8.0 and Mg <= 10.0 :  
 print('Великий\nСерйозна руйнація на територіях довжиною/шириною в сотні кілометрів.')  
else:  
 print('Рідкісно великий')  
printTimeStamp("Глигало")



3. Деякий телефонний тариф включає 200 хвилин в мережі та 50 текстових повідомлень за ₴35.00 на місяць. Кожна додаткова хвилина коштує ₴0.17, а додаткова SMS - ₴0.15. Всі рахунки включають ₴0.44 внеску в Пенсійний фонд та 5%-вий податок.

Напишіть програму, яка зчитує кількість хвилин та текстових повідомлень, використаних за місяць. Виведіть базову плату за користування (без внесків та податків), а також загальний рахунок для користувача. Забезпечте виведення грошових сум з 2 знаками після коми.

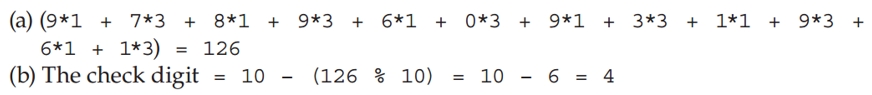
import datetime  
#import math  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
t = int(input("Кількість хвилилн: "))  
sms = int(input("Кількість SMS: "))  
if t > 200:  
 t1 = t - 200  
else:  
 t1 = 0  
if sms > 50:  
 sms1 = sms - 50  
else:  
 sms1 = 0  
suma = (35 + t1\*0.17 + sms1\*0.15)\*0.5 + (35 + t1\*0.17 + sms1\*0.15) + 0.44  
print("{:.2f} гривень" .format(suma))  
printTimeStamp("Глигало")



4. 13-цифровий міжнародний стандартний номер книги (International Standard Book Number) – це унікальний код, який ідентифікує книгу, що продається. Остання цифра називається перевірочною цифрою (check digit) і використовується для перевірки на помилки. Для обчислення цього значення

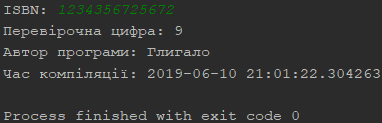
* 1. кожну з 12 перших цифр почережно множать на 1 або 3,
  2. додають добутки та ділять результат за модулем 10. Перевірочна цифра отримується відніманням від 10 отриманого результату.

Наприклад, для ISBN 9789609319614



Напишіть програму, що зчитуватиме 13-цифровий ISBN та перевірятиме останню цифру на валідність цього номеру.

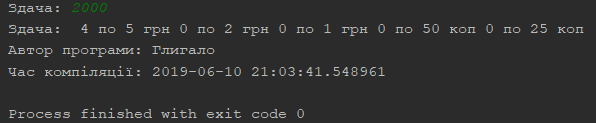
import datetime  
#import math  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
a = input("ISBN: ")  
l = list(a)  
sum = int(l[0]) + int(l[1])\*3 + int(l[2]) + int(l[3])\*3 + int(l[4]) + int(l[5])\*3 + int(l[6]) + int(l[7])\*3 + int(l[8]) + int(l[9])\*3 + int(l[10]) + int(l[11])\*3  
num = 10 -(sum%10)  
print("Перевірочна цифра:", num)  
printTimeStamp("Глигало")



5. Розгляньте програмне забезпечення, яке працює на апараті для приготування кави. Одна із задач, яку воно повинно виконувати – визначення того, скільки здачі повертати з готівкової оплати товару.

Напишіть програму, яка починає роботу зі зчитування кількості копійок (ціле число). Далі програма має обчислювати та виводити на екран номінали монет та їх кількість для того, щоб видати здачу покупцю. Здача має віддаватись якомога меншою кількістю монет. Припускаємо, що машина завантажена монетами номіналами 25, 50 копійок, 1, 2 та 5 грн.

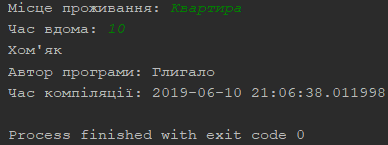
import datetime  
#import math  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
Money = int(input("Здача: "))  
m5 = Money // 500  
m2 = (Money - ((Money // 500)\*500))//200  
m1 = ((Money - ((Money // 500)\*500))-(((Money - ((Money // 500)\*500))//200)\*200))//100  
m50 = (((Money - ((Money // 500)\*500))-(((Money - ((Money // 500)\*500))//200)\*200)) - ((((Money - ((Money // 500)\*500))-(((Money - ((Money // 500)\*500))//200)\*200))//100)\*100))//50  
m25 = ((((Money - ((Money // 500)\*500))-(((Money - ((Money // 500)\*500))//200)\*200)) - ((((Money - ((Money // 500)\*500))-(((Money - ((Money // 500)\*500))//200)\*200))//100)\*100))-(((((  
Money - ((Money // 500)\*500))-(((Money - ((Money // 500)\*500))//200)\*200)) - ((((Money - ((Money // 500)\*500))-(((Money - ((Money // 500)\*500))//200)\*200))//100)\*100))//50)\*50))//25  
print("Здача: ",m5 , "по 5 грн",m2 , "по 2 грн",m1 , "по 1 грн",m50 , "по 50 коп",m25 , "по 25 коп")  
printTimeStamp("Глигало")



6. Напишіть програму, яка рекомендуватиме людині домашнього улюбленця відповідно до стилю життя його майбутнього власника.

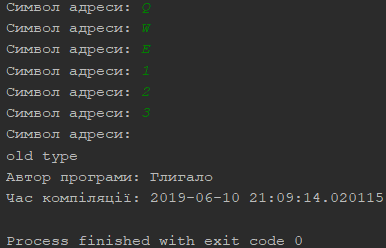
|  |  |  |
| --- | --- | --- |
| **Місце проживання** | **Час вдома** | **Рекомендація** |
| Будинок | Понад 18 годин | В’єтнамське порося |
| Будинок | Від 10 до 17 годин | Собака |
| Будинок | Менше 10 годин | Змія |
| Квартира | Більше 10 годин | Кішка |
| Квартира | Менше 10 годин | Хом’як |
| Гуртожиток | Більше 6 годин | Рибки |
| Гуртожиток | Менше 6 годин | Мурашник |

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
Place = input("Місце проживання: ")  
if Place == "Будинок":  
 Time = int(input("Час вдома: "))  
 if Time >= 18:  
 print("В'єтнамське порося")  
 elif Time < 10:  
 print("Змія")  
 else:  
 print("Собака")  
elif Place == "Квартира" :  
 Time = int(input("Час вдома: "))  
 if Time > 10:  
 print("Кішка")  
 else:  
 print("Хом'як")  
elif Place == "Гуртожиток":  
 Time = int(input("Час вдома: "))  
 if Time > 6:  
 print("Рибки")  
 else:  
 print("Мурашник")  
printTimeStamp("Глигало")



8. У деяких штатах США старіші номерні знаки складаються з 3 великих літер, за якими йдуть 3 цифри. Коли всі можливі номерні знаки були вичерпані, перейшли до формату «4 цифри + 3 великі букви». Напишіть програму, яка зчитує рядок символів, а потім визначає, чи підходить введений номер під якийсь із форматів. Якщо це не так, слід повідомити користувача про помилку.

import datetime  
#import math  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
l = []  
while True:  
 s = (input("Символ адреси: "))  
 l.append(s)  
 if s == "":  
 break  
l.pop()  
if len(l) < 6 or len(l) > 7:  
 print("Wrong address")  
else:  
 if (l[1]) and (l[2]) and (l[3]) and (l[4]) is int:  
 print("New type of address")  
 else:  
 print("old type")  
printTimeStamp("Глигало")



9. Напишіть програму, яка конвертує десяткове число в двійкове. Зчитуйте десяткове число від користувача як ціле число, а потім використовуйте алгоритм ділення для виконання перетворення.

**Ініціалізувати змінну result порожнім рядком**

**Оголосити змінну q – число для перетворення**

**repeat**

**Оголосити r та присвоїти йому значення остачі від ділення q на 2**

**Звести r до рядкового типу та додати її до початку**

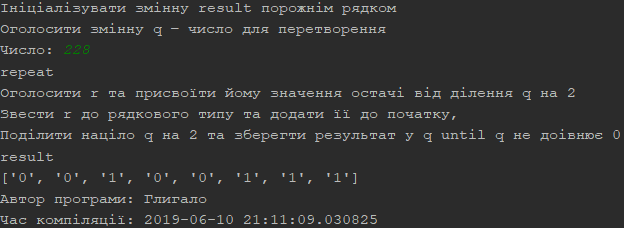
**result**

**Поділити націло q на 2 та зберегти результат у q**

**until q не доівнює 0**

Коли алгоритм завершує роботу, змінна result міститиме двійкове представлення числа. Відобразіть результат на екрані з відповідним пояснювальним текстом.

import datetime  
#import math  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
print("Ініціалізувати змінну result порожнім рядком")  
result = []  
print("Оголосити змінну q – число для перетворення")  
q = int(input("Число: "))  
print("repeat\nОголосити r та присвоїти йому значення остачі від ділення q на 2")  
while q > 0:  
 r = q % 2  
 result.append(str(r))  
 q = q//2  
print("Звести r до рядкового типу та додати її до початку, \nПоділити націло q на 2 та зберегти результат у q until q не доівнює 0")  
print("result")  
print(result)  
printTimeStamp("Глигало")



10. Поверхня Землі викривлена, і відстань між градусами довготи варіюється залежно від широти. У результаті, знаходження відстані між 2 точками на поверхні Землі дещо ускладнюється.

Нехай (t1, g1) та (t2, g2) – широти і довготи 2 точок на поверхні планети. Відстань між ними в кілометрах визначається за формулою:

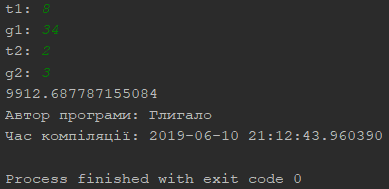
**Відстань = 6371.01 × arccos(sin(t1) × sin(t2) + cos(t1) × cos(t2) × cos(g1 − g2))**

Величина 6371.01 – середній радіус планети Земля в кілометрах.

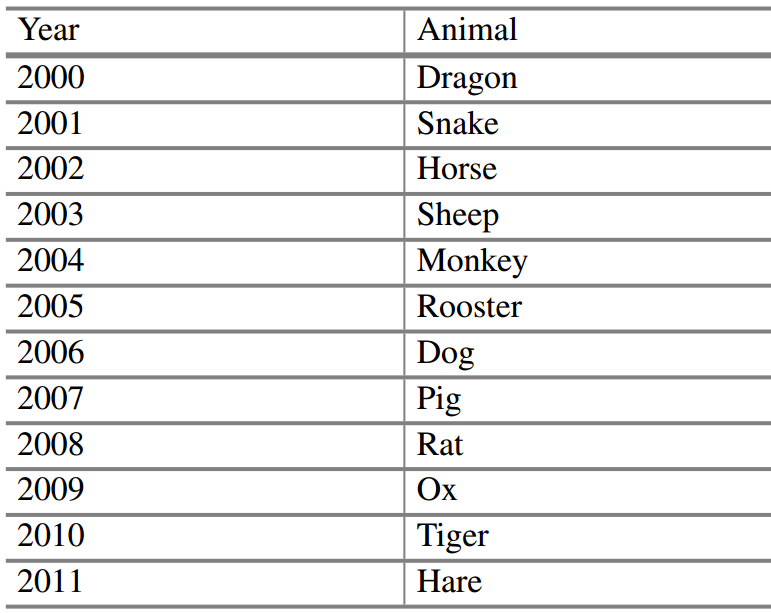
Напишіть додаток, що дозволятиме користувачу вводити широту і довготу 2 точок у градусах. Ваша програма має виводити відстань між точками в кілометрах.

Підказка: тригонометричні функції в Python працюють з радіанною мірою. Як результат, буде потрібно конвертувати користувацький ввід з градусів у радіани до обчислення відстані. Модуль math містить функцію radians(), яка переводить градуси в радіани.

import datetime  
import math  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
t1 = int(input("t1: "))  
g1 = int(input("g1: "))  
t2 = int(input("t2: "))  
g2 = int(input("g2: "))  
l = 6371.01\*math.acos(math.radians(math.sin(math.radians(t1))\*math.sin(math.radians(t2)) + math.cos(math.radians(t1))\*math.cos(math.radians(t2))\*math.cos(math.radians(g1 - g2))))  
print(l)  
printTimeStamp("Глигало")

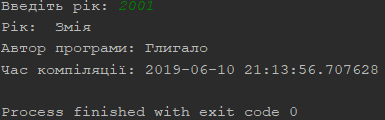


12. Китайський гороскоп ставить у відповідність 12-річному циклу назви тварин:



Напишіть програму, яка зчитує рік і виводить тварину, що йому відповідає. Програма має коректно працювати з роками нашої ери, а не лише з представленими в таблиці.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
x = ["Дракон","Змія","Кінь","Вівця","Мавпа","Птах","Пес","Свиня","Миша","Бик","Тигр","Заєць"]  
y = int(input("Введіть рік: "))  
if y >=2000 and y <= 2011:  
 print("Рік: ", x[y-2000])  
elif y >2011:  
 while y >= 2011:  
 y-=12  
 print("Рік: "+x[y-2000])  
elif y <2000:  
 while y <= 2000:  
 y+=12  
 print("Рік: "+x[y-2000])  
printTimeStamp("Глигало")



14. Коли дмухає вітер в холодну погоду, повітря відчувається ще холоднішим, ніж є насправді. Це відбувається тому, що рух повітря пришвидшує охолодження теплих об’єктів, таких як люди. Даний ефект називають прохолодністю вітру.

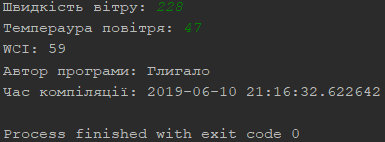
У 2001 році в Канаді, Великобританії та США було впроваджено формулу, яка обчислює індекс прохолодності вітру (wind chill index):

У даній формулі – температура повітря в градусах Цельсія, – швидкість вітру в км/год.

Напишіть програму, яка спочатку зчитує температуру повітря та швидкість вітру, а потім виводить індекс прохолодності вітру, округлений до найближчого цілого.

Індекс прохолодності вітру вважається коректним за умови, якщо температура не перевищує 10ºС, а швидкість вітру буде понад 4.8 км/год. Дані умови потрібно врахувати при виводі інформації.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
V = int(input("Швидкість вітру: "))  
T = int(input("Темпераура повітря: "))  
WCI = 13.12+0.6215\*T-11.37\*V\*\*0.16+0.3965\*T\*V\*\*0.16  
print("WCI:", int(WCI))  
if V > 4.8 and T<10:  
 print("WCI коректний")  
printTimeStamp("Глигало")



15. Високосні роки визначаються за наступними правилами:

• Будь-який рік, що націло ділиться на 400, високосний.

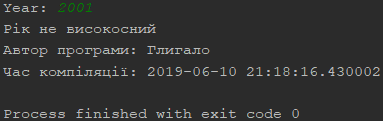
• З решти всі роки, що діляться на 100 – невисокосні.

• З решти всі роки, що діляться на 4, є високосними.

• Всі інші - невисокосні.

Напишіть програму, яка зчитує рік та виводить повідомлення про те, чи є цей рік високосним.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
Year = int(input("Year: "))  
if Year % 400 == 0:  
 print("Рік високосний")  
else:  
 if Year % 100 == 0 and Year % 4 == 0:  
 print("Рік високосний")  
 else:  
 print("Рік не високосний")  
printTimeStamp("Глигало")



17. Кількість енергії, необхідної для збільшення температури 1 граму матеріалу на 1ºС, називається теплоємністю C. Загальна енергія, потрібна для підняття температури m грамів матеріалу на ΔT градусів Цельсія можна обчислити за формулою:

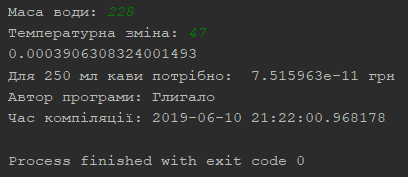
Напишіть програму, яка зчитує масу води та температурну зміну від користувача. Ваша програма має відображати загальну кількість енергії, яку потрібно додати чи забрати, щоб досягнути заданої зміни температури.

Підказка: питома теплоємність води складає 4.186 Дж/г ºС. Оскільки вода має густину 1.0 г/мл, можна використовувати як грами, так і мілілітри в даному випадку.

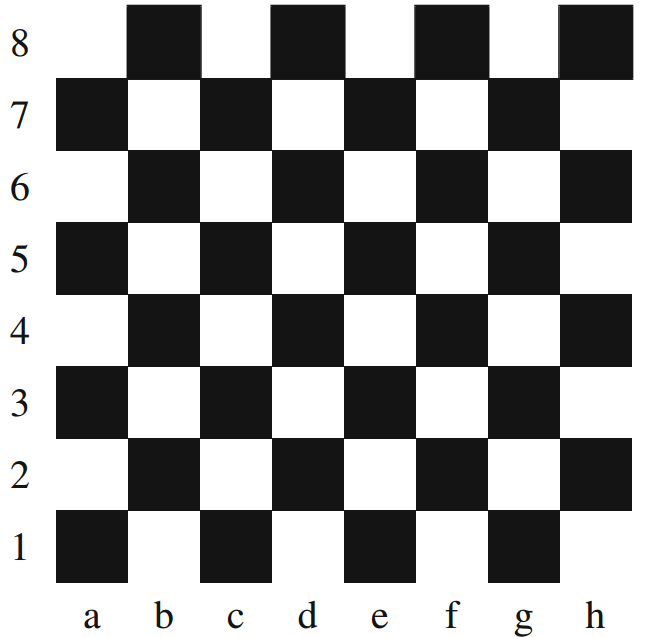
Розширьте свою програму, щоб вона також обчислювала вартість нагріву води. Оплата електроенергії зазвичай відбувається в кВт год, а не в джоулях. У даній вправі слід припустити, що електроенергія коштує 1.33 грн за кВт-год. Використовуйте програму, щоб обчислити вартість кип’ятіння 1 чашки води для кави.

Підказка: слід розглянути множник, що дозволяє конвертувати джоулі в кВт-год, щоб завершити виконання вправи.

import datetime  
#from math import sqrt  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
m = int(input("Маса води: "))  
dt = int(input("Температурна зміна: "))  
q = 4.186  
C = q/(m\*dt)  
print(C)  
print("Для 250 мл кави потрібно: ", (q/(250\*80))\*2.7\*10\*\*(-7)\*1.33,"грн")  
printTimeStamp("Глигало")

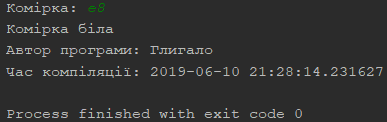


18, Позиції на шаховій дошці визначаються буквою (колонка) та цифрою (рядок), як показано на рисунку.



Напишіть програму, яка зчитує позицію від користувача. Використовуйте оператор if, щоб визначити, чи колонка починається з білої або чорної клітини. Далі використайте модулярну арифметику, щоб повідомити колір клітинки в цьому рядку. Наприклад, при вводі a1 програма має повідомити, що клітинка чорна. Якщо користувач вводить d5, програма виводить повідомлення про те, що комірка біла. Ваш додаток повинен перевіряти, чи введені дані коректні (не виходять за межі дошки), і якщо ні – виводити на екран повідомлення про помилку.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
l = ["a", "b", "c", "d", "e", "f", "g", "h"]  
pat = input("Комірка: ")  
if pat[0] not in l or int(pat[1])>8 or int(pat[1])<1 :  
 print("Такої нема")  
elif ((l.index(pat[0])+1) + int(pat[1]))%2 == 0:  
 print("Комірка чорна")  
else:  
 print("Комірка біла")  
printTimeStamp("Глигало")

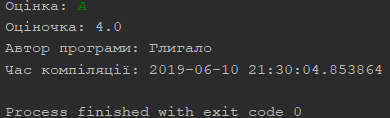


19. У деякому університеті використовується буквенна система оцінювання у відповідності до балів із таблиці:

|  |  |
| --- | --- |
| **Буквенна оцінка** | **Бальна оцінка** |
| A+ | >4.0 |
| A | 4.0 |
| A- | 3.7 |
| B+ | 3.3 |
| B | 3.0 |
| B- | 2.7 |
| C+ | 2.3 |
| C | 2.0 |
| C- | 1.7 |
| D+ | 1.3 |
| D | 1.0 |
| F | 0 |

Напишіть програму, яка зчитуватиме буквенну оцінку від користувача. Далі вона виводить на екран бальну оцінку, що відповідає введеній буквенній. Забезпечте коректну роботу для ситуації, коли введена буквенна оцінка не передбачена в таблиці.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
Marks = {"A+":">4.0","A":4.0,"A-":3.7,"B+":3.3,"B":3.0,"B-":2.7,"C+":2.3,"C":2.0,"C-":1.7,"D+":1.3,"D":1.0,"F":0}  
Mark = input("Оцінка: ")  
if Mark in Marks.keys():  
 print("Оціночка: "+str(Marks[Mark]))  
else:  
 print("Нема такої")  
printTimeStamp("Глигало")



21. Найбільший спілдьний дільник двох додатніх цілих чисел (n, m) є найбільшим числом d, на яке вони діляться без остачі. Один з алгоритмів для знаходження НСД:

**Ініціалізувати d значенням меншого з чисел m і n.**

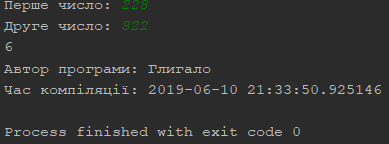
**While на d націло не діляться m і n do**

**Зменшити значення d на 1**

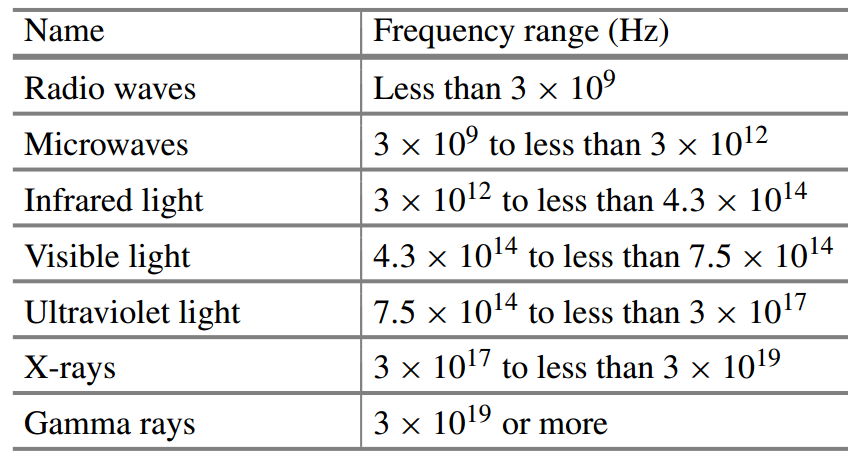
**Вивести d**

Напишіть програму, яка зчитує 2 цілих додатніх числа та знаходить найбільший спільний дільник за алгоритмом.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
n = int(input("Перше число: "))  
m = int(input("Друге число: "))  
while n != 0 and m != 0:  
 if n > m:  
 n %= m  
 else:  
 m %= n  
print(n+m)  
printTimeStamp("Глигало")

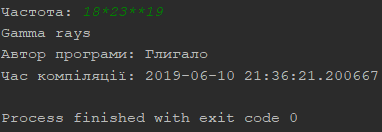


22. Електромагнітне випромінювання можна розділити на 7 категорій за частотою:



Напишіть програму, яка зчитує частоту радіації та виводить відповідну назву хвиль.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
Num = input("Частота: ")  
num = Num.split("\*", 3)  
Tic = int(num[0])\*int(num[1])\*\*int(num[3])  
if Tic < 3\*10\*\*9:  
 print("Radio waves")  
elif Tic >= 3\*10\*\*19:  
 print("Gamma rays")  
elif Tic> 3\*10\*\*9 and Tic < 3\*10\*\*12:  
 print("Microwaves")  
elif Tic > 3\*10\*\*12 and Tic < 4.3\*10\*14:  
 print("Infrared light")  
elif Tic > 4.3\*10\*14 and Tic < 7.5\*10\*\*14:  
 print("Visible light")  
elif Tic > 7.5\*10\*\*14 and Tic < 3\*10\*17:  
 print("Ultraviloet light")  
else:  
 print("X-ray")  
printTimeStamp("Глигало")

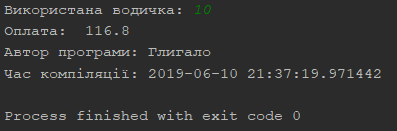


23. Компанія, що займається постачанням води, обчислює клієнтські рахунки за наступною формулою:

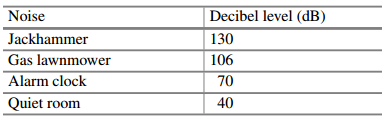
1. Фіксований внесок – 20 грн.
2. За перші 30 кубометрів плата складає 9.86 грн./м3
3. За наступні 20 кубометрів – 11.22 грн./м3
4. За наступні 10 кубометрів – 13.06 грн./м3
5. За додаткові кубометри – 17.89 грн./м3

Напишіть програму, яка буде зчитувати споживання води в кубометрах та виводити на екран рахунок.

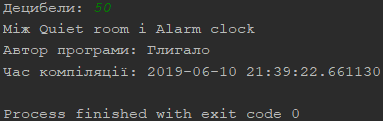
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
water = int(input("Використана водичка: "))  
if water <= 30:  
 print("Оплата: ", str(20+(water\*9.68)))  
elif 30 < water <= 50:  
 print("Оплата: ", str(20+(30\*9.68)+((water-30)\*11.22)))  
elif 50 < water <= 60:  
 print("Оплата: ", str(20+(30\*9.68)+(20\*11.22)+((water-50)\*13.06)))  
else:  
 print("Оплата: ", str(20+(30\*9.68)+(20\*11.22)+(10\*13.06)+(water-60)\*17.89))  
printTimeStamp("Глигало")



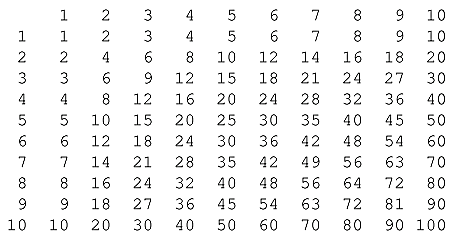
24. Наступна таблиця перелічує рівень гучності в децибелах для кількох поширених шумів. Напишіть програму, яка зчитує рівень шуму в децибелах від користувача. Якщо користувач вводить рівень шуму, що співпадає з табличним значенням, виведіть повідомлення, що містить лише тип шуму. Якщо користувач вводить кількість децибелів, яка знаходиться в проміжку між шумами, програма повідомлятиме, між якими шумами розташовується даний рівень шуму. Забезпечте коректну роботу програми при значеннях, нижчих за найтихіший шум, та значеннях, що перевищують найгучніший шум з таблиці.



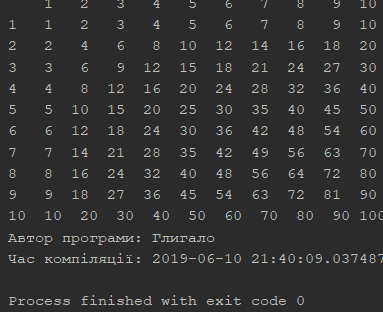
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
sound = int(input("Децибели: "))  
dc = {130:"Jackhammer",106:"Gas lawnmower",70:"Alarm clock",40:"Quiet room"}  
if sound in dc.keys():  
 print(dc[sound])  
elif sound <40:  
 print("Тихіше чим в Quiet room")  
elif 70>sound >40:  
 print("Між Quiet room і Alarm clock")  
elif 106>sound >70:  
 print("Між Gas lawnmower і Alarm clock")  
elif 130>sound >106:  
 print("Між Gas lawnmower і Jackhammer")  
else:  
 print("Більше Jackhammer")  
printTimeStamp("Глигало")



25. Напишіть програму, яка відобразить табличку множення 10х10, як показано на рисунку:

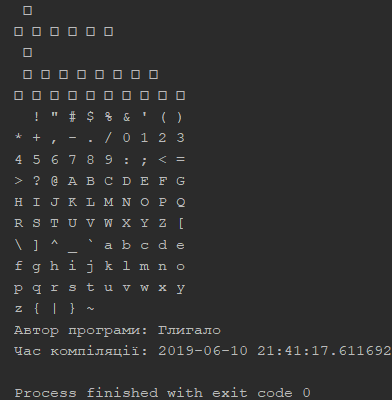


import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
print ("{:>0}{:>5}{:>4}{:>4}{:>4}{:>4}{:>4}{:>4}{:>4}{:>4}{:>4}".format("",1,2,3,4,5,6,7,8,9,10))  
for x in range(1,11):  
 print((x), end= '')  
 for y in range(1,11):  
 print("%4d" % (x\*y), end='')  
 print()  
printTimeStamp("Глигало")



27. Виведіть на екран символи таблиці ASCII від 0 до 127 разом з їх номером у десятковій, двійковій, вісімковій та шістнадцятковій системах числення.

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
for i in range(0, 128):  
 print(chr(i), end=' ')  
 if (i - 1) % 10 == 0:  
 print()  
  
print()  
printTimeStamp("Глигало")



**Рівень С**

1. Рулетка має 38 комірок: 18 чорних, 18 червоних і 2 зелених. Зелені комірки позначаються як 0 та 00. Червоними комірками є 1, 3, 5, 7, 9, 12, 14, 16, 18, 19, 21, 23, 25, 27, 30 32, 34 та 36. Решта цілих чисел з діапазону від 1 до 36 є чорними.

На рулетку можна зробити багато ставок. Для цієї вправи розглянемо наступні можливі ставки:

• Одне число (від 1 до 36, 0 або 00)

• Червоне vs Чорне

• Парне vs Непарне (0 та 00 не включаються)

• Від 1 до 18 vs від 19 до 36

Напишіть програму, яка симулює кручення колеса рулетки за допомогою генератора випадкових чисел. Виведіть вибране число та всі ставки, які потрібно оплатити. Наприклад, для 13:

На рулетці випало 13...

Виплатити 13

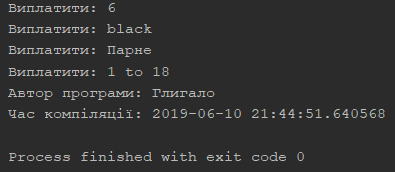
Виплатити Black

Виплатити Odd

Виплатити 1 to 18

Якщо результати симуляції випадають на 0 або 00, слід вивести Виплатити 0 або Виплатити 00 без іншої інформації.

import datetime  
import random  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
l = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,0,00]  
red = [1,3,5,7,9,12,14,16,18,19,21,23,25,27,30,32,34,36]  
  
s = random.choice(l)  
print("Виплатити:", s )  
if s in red:  
 print("Виплатити: red")  
elif s in [0, 00]:  
 print("Виплатити: Green")  
else:  
 print("Виплатити: black")  
if s%2 == 0:  
 print("Виплатити: Парне")  
elif s in [0, 00]:  
 pass  
else:  
 print("Виплатити: Не парне")  
if s < 18:  
 print("Виплатити: 1 to 18")  
else:  
 print("Виплатити: 19 to 36")  
printTimeStamp("Глигало")

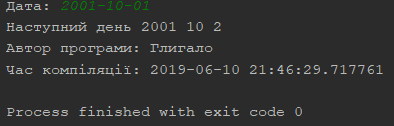


2. Напишіть програму, яка зчитує дату від користувача та обчислює дату наступного дня. Дата буде вводитись у числовій формі трьома окремими виразами (рік, місяць, день). Наприклад, якщо користувач вводить значення, яке представляє день 2017-05-10, Ваша програма має показати повідомлення, яке говоритиме, що наступним днем буде 2017-05-11.

* Якщо користувач вводить значення на зразок 2017-04-30, програма має вказати, що наступним днем буде 2017-05-01.
* Якщо користувач вводить значення, що представляє дату 2016-12-31, програма має вивести наступний день - 2017-01-01.
* Забезпечте коректну роботу для високосних років.

Не використовуйте вбудованих методів для обчислення наступного дня!

import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
date = input("Дата: ")  
Date = date.split("-", 3)  
year = int(Date[0])  
if (year % 400 == 0):  
 s = True  
elif ((year - ((year // 400)\*400))%100 == 0):  
 s = False  
elif ((year - (((year - (year//400\*400))//100)\*100))%4 == 0):  
 s = True  
else:  
 s = False  
month = int(Date[1])  
if month in (1, 3, 5, 7, 8, 10, 12):  
 month\_l = 31  
elif month == 2:  
 if s:  
 month\_l = 29  
 else:  
 month\_l = 28  
else:  
 month\_l = 30  
day = int(Date[2])  
if day < month\_l:  
 day += 1  
else:  
 day = 1  
 if month == 12:  
 month = 1  
 year += 1  
 else:  
 month += 1  
print("Наступний день", year,month, day)  
for year in (year,day,month):  
 if day > month\_l:  
 break  
printTimeStamp("Глигало")

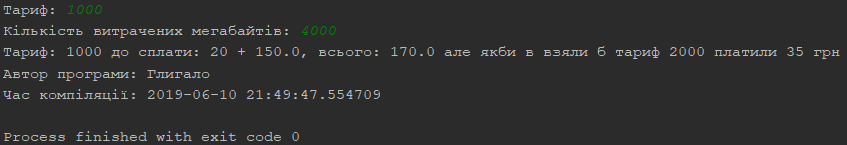


4. Інтернет-провайдер має три тарифи для населення:

|  |  |
| --- | --- |
| Тариф 1000 | 1000 Мб на місяць за 20 грн, позатарифно: 1Мб = 0.05 грн. |
| Тариф 2000 | 2000 Мб на місяць за 35 грн, позатарифно: 1Мб = 0.04 грн. |
| Тариф 5000 | 5000 Мб на місяць за 85 грн., позатарифно: 1 Мб = 0.02 грн. |

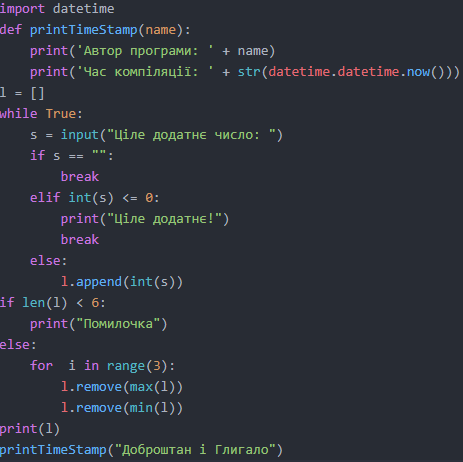
Напишіть програму, яка повинна отримувати цифру-тариф та кількість витрачених мегабайтів, а виводити підсумковий рахунок за місяць. Якщо використовується тариф 1000 або 2000, виведіть додатково інформацію про те, скільки б платили користувачі, якби перейшли на більш дорогі тарифи.

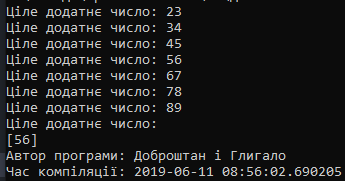
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
l = {1000:20, 2000:35, 5000:85}  
s = {1000:0.05, 2000:0.04, 5000:0.02}  
trf = int(input("Тариф: "))  
mb = int(input("Кількість витрачениx мегабайтів: "))  
if (trf==1000) and (mb > 1300):  
 if mb > 2300:  
 print("Тариф: {0} до сплати: {1} + {2}, всього: {3} але якби в взяли б тариф 5000 до сплати було б: {4}".format(trf, l[trf], s[trf]\*(mb - trf),s[trf]\*(mb - trf) + l[trf], l[5000] + (mb - 5000)\*s[5000]))  
 else:  
 print("Тариф: {0} до сплати: {1} + {2}, всього: {3} але якби в взяли б тариф 2000 до сплати було б: 35 грн".format(trf, l[trf], s[trf]\*(mb - trf),s[trf]\*(mb - trf) + l[trf]))  
elif (trf==2000) and (mb > 3250):  
 print("Тариф: {0} до сплати: {1} + {2}, всього: {3} але якби в взяли тариф 5000 до сплати було б: {4}".format(trf, l[trf], s[trf]\*(mb - trf), s[trf]\*(mb - trf) + l[trf], l[5000] + (mb - 5000)\*s[5000]))  
elif mb <= trf:  
 print("Тариф:", trf, "до сплати: ", l[trf], "грн")  
else:  
 print("Тариф: {0} до сплати: {1} + {2}, всього: {3}" .format(trf, l[trf], s[trf]\*(mb - trf),s[trf]\*(mb - trf) + l[trf]))  
printTimeStamp("Глигало")



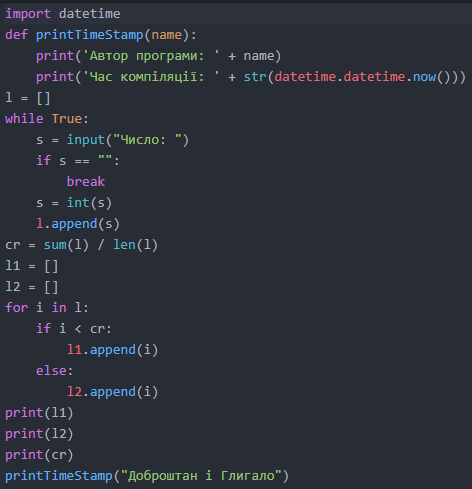
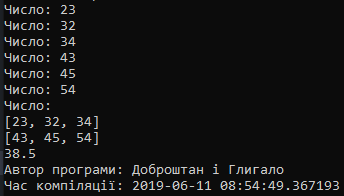
**День 2**

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 нього0

 .

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

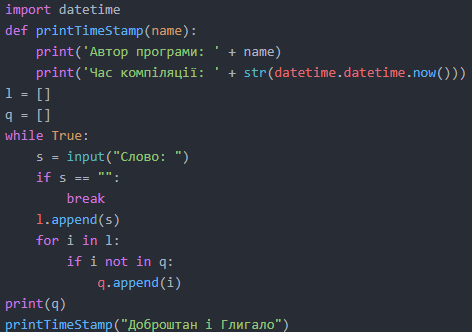
second

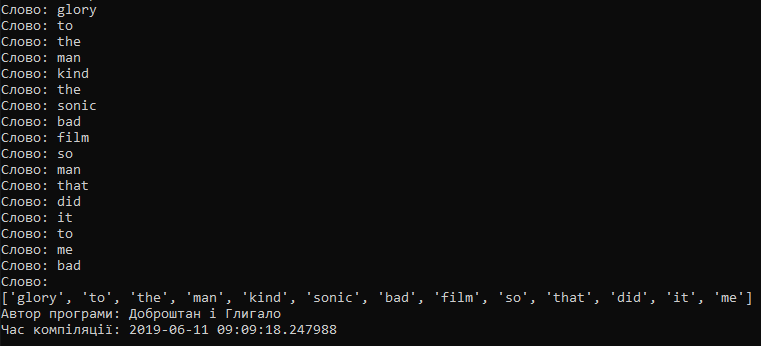
програма повинна буде вивести

first

second

third





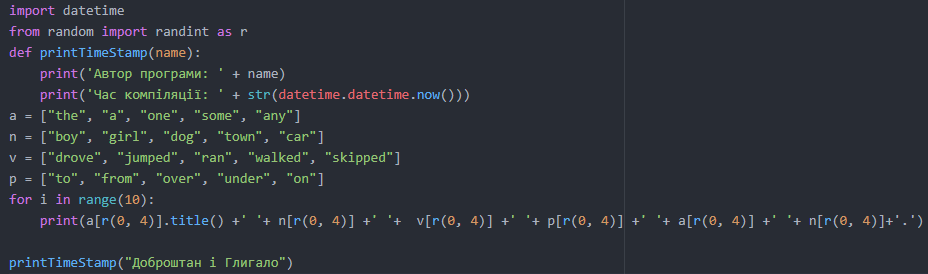
4. Напишіть додаток, який використовує генерування випадкових чисел для створення речень. Використовуйте 4 списки рядків: article, noun, verb та preposition. Створіть речення, обравши з кожного масиву слова в такому порядку: артикль, іменник, дієслово, прийменник, артикль та іменник. Після вибору кожного 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.

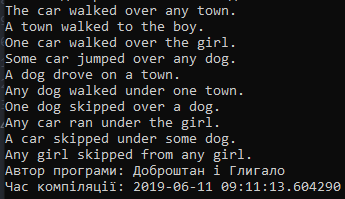
*Артиклі:* "the", "a", "one", "some" та "any";

*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:

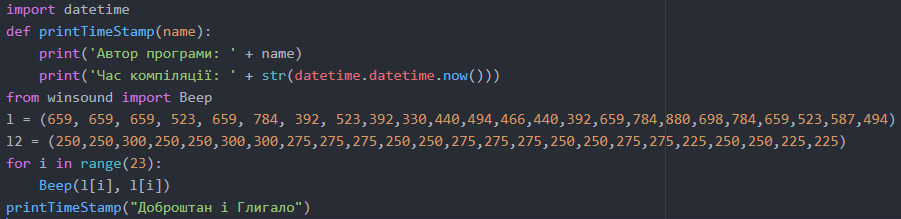
(659,250), (659,250), (659,300), (523,250), (659,250), (784,300), (392,300), (523,275), (392,275), (330,275), (440,250), (494,250), (466,275), (440,275), (392,275), (659,250), (784,250), (880,275), (698,275), (784,225), (659,250), (523,250), (587,225), (494,225).

Для підключення біперу потрібно імпортувати модуль

import winsound

і викликати функцію

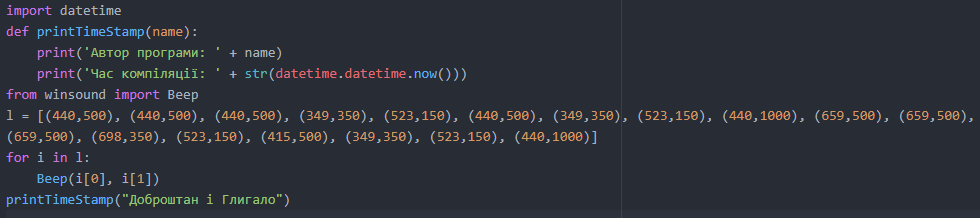
winsound.Beep(частота у Гц, тривалість у мс)



6

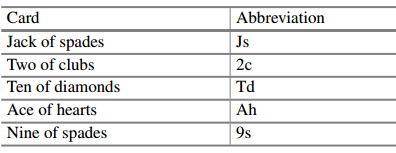
6. Якщо Вам сподобалась попередня задача, спробуйте тепер іншу мелодію, збережену в *двовимірному* масиві (списку списків):

(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)

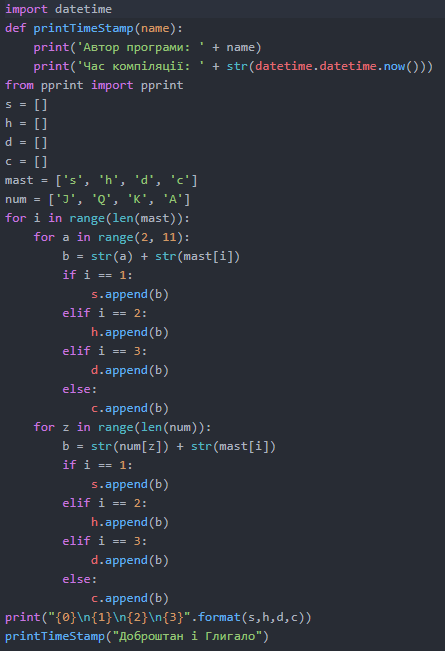


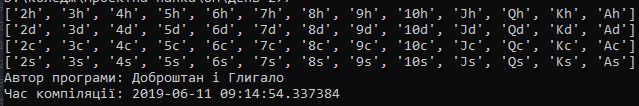


7. Звичайна колода карт має 52 карти, кожна з яких описується мастю і значенням. Мастями можуть бути піки (spades), черви (hearts), трефи (diamonds) та бубни (clubs), а значеннями – числа від 2 до 10, валет (Jack), дама (Queen), король (King) і туз (Ace). Приклади позначень карт описані в таблиці.



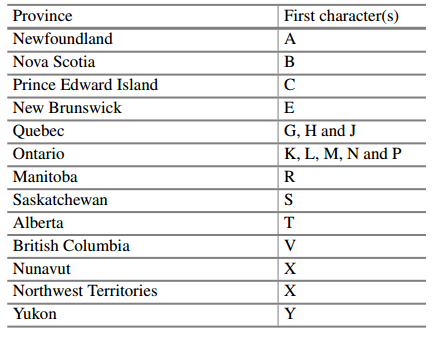
Напишіть програму, яка буде створювати колоду карт у вигляді списку.



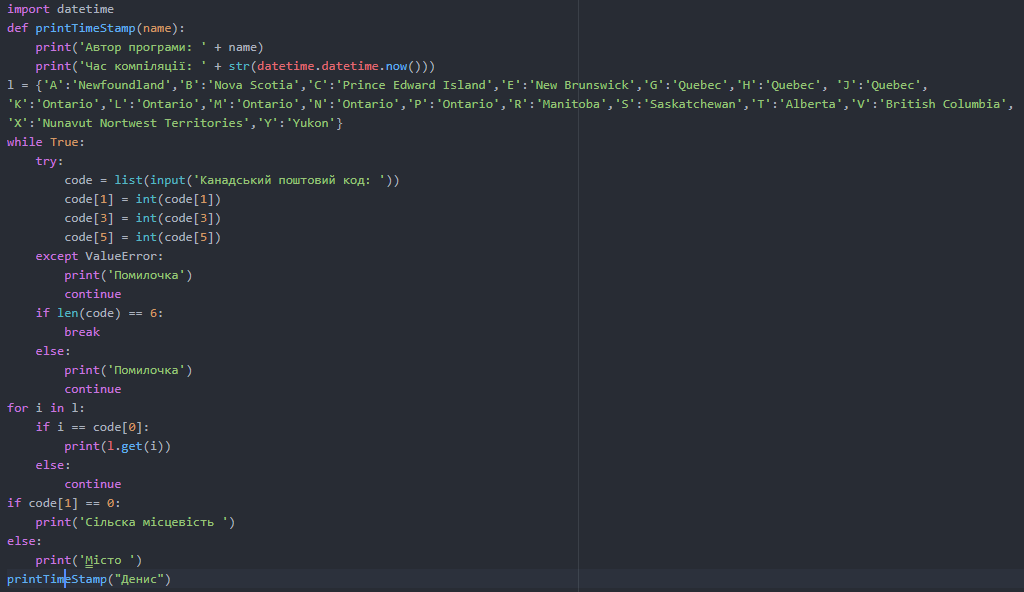


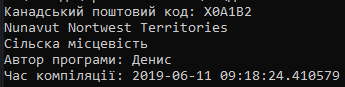
Рівень Б

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.



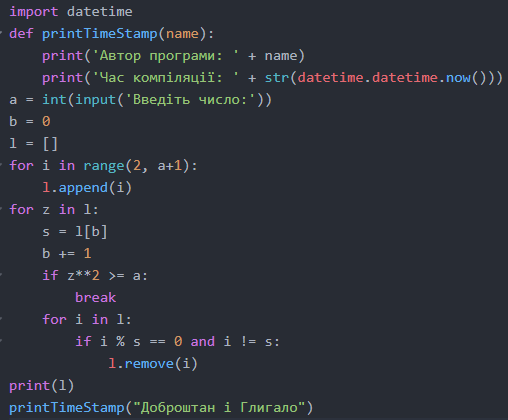
Другий символ позначає сільську місцевість (0) або місто (інакше).

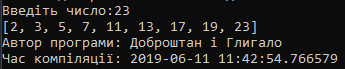
Напишіть програму, яка зчитує поштовий код та визначає провінцію та місцевість проживання. Наприклад, T2N1N4 вказує на проживання в місті провінції Альберта. Якщо введено X0A1B2, поштовий код вказує на село в Nunavut або Northwest Territories. Використовуйте словник для відображення перших символів коду в назву провінції. Забезпечте коректну роботу програми для некоректного коду. 



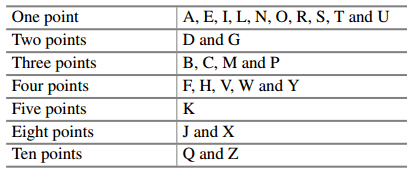
4.

|  |
| --- |
| 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 . |



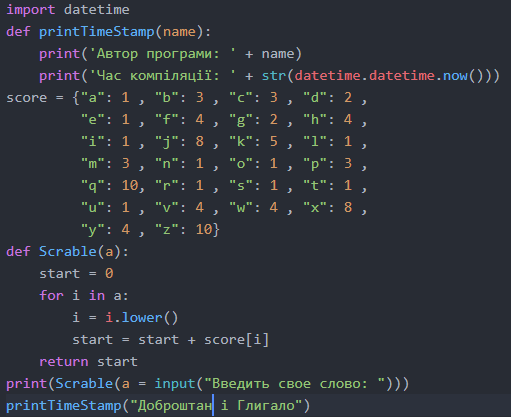


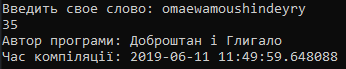
5. У грі Scrabble™ кожна буква оцінюється балами, як показано в таблиці. Загальна оцінка слова – це сумарна кількість очків за кожну букву:



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.

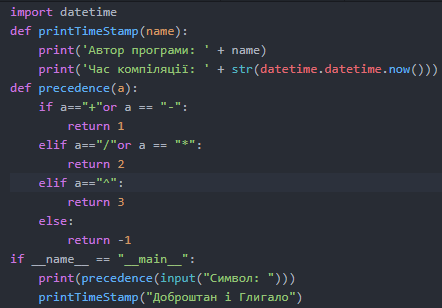
У грі Scrabble™ включено деякі squares, які домножають значення букви чи всього слова на певні значення. Це правило в задачі ігнорується.

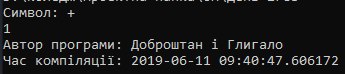




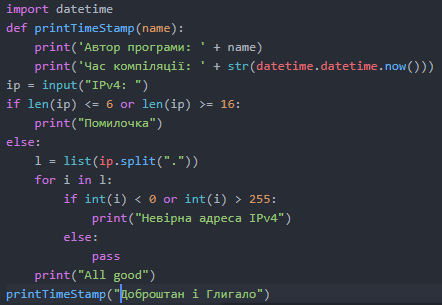
8. Напишіть функцію precedence(), яка повертає ціле число, що показує пріоритет математичного оператора. Рядок, який містить оператор, передається як єдиний параметр функції. Сама функція має повертати 1 для + та -, 2 для \* та /, 3 для ˆ. Якщо переданий рядок не має цих операторів, функція має повернути -1.

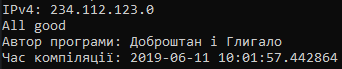
Основна програма буде зчитувати оператор від користувача та або виводить пріоритет оператора, або повідомлення, що не було введено оператор. Your main program should only run when the file containing your solution has not been imported into another program.





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].

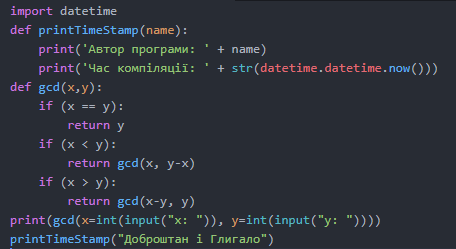


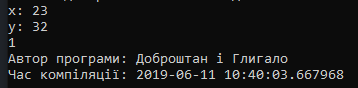


День 3

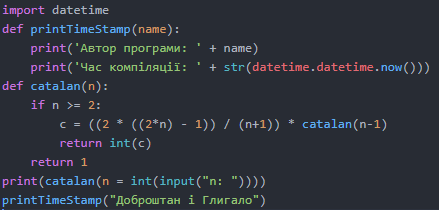
**Рівень А**

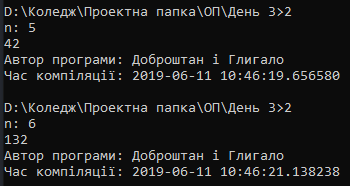
1. Опишіть рекурсивну функцію GCD, яка повертає найбільший спільний дільник чисел x та y. Обчислення НСД відбувається рекурсивним чином відповідно до формули





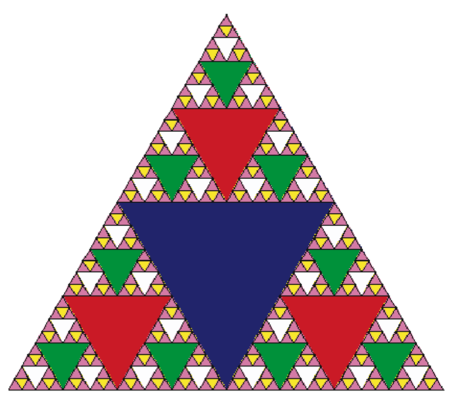
2. Вивести n-те число Каталана за допомогою рекурсивного алгоритму





**Рівень С**

**3.** Фрактали є самоподібними структурами, тому природною є рекурсивна структура алгоритму їх побудови. Напишіть програму, яка буде виводити на екран килим Серпінського.



Ідея побудови килиму наступна. Починаємо з великого залитого одним кольором трикутника. Ділимо його на 4 частини, з’єднавши середини кожної зі сторін трикутника (темносиній трикутник на рисунку). Продовжуємо робити так і для отриманих трикутників (червоні трикутники, потім зелені, білі та жовті). Таким чином, Вам будуть потрібні функції, що

* знаходитимуть середини сторін заданого трикутника (визначається покоординатно)
* відрисовуватимуть сам трикутник, наприклад, за допомогою turtle

формуватимуть нові трикутники за вище наведеними правилами



